

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет програмування та комп'ютерних і телекомунікаційних систем

Кафедра телекомунікацій, медійних та інтелектуальних технологій

КВАЛІФІКАЦІЙНИЙ ПРОЄКТ

Бакалавр

Освітній рівень

Спеціальність 172 Телекомунікації та радіотехніка

Шифр і назва спеціальності

на тему МОДУЛЬ КЕРУВАННЯ ПАРАМЕТАМИ СИСТЕМИ SMART HOUSE НА ОСНОВІ ПОВЕДІНКИ МЕШКАНЦІВ

КПТР.2017019.01.07 ПЗ

Виконав:

студент 4 курсу, група ТР-17-1

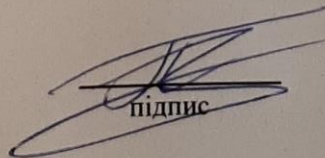


підпис

О.І. Свачій

Ініціали, прізвище

Керівник: д.т.н. проф.



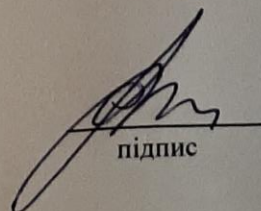
підпис

О.І. Полікаровських

Ініціали, прізвище

До захисту допускаю:

Зав. кафедри ТМІТ
д-р техн. наук, доц.



підпис

С.К. Підченко

Ініціали, прізвище

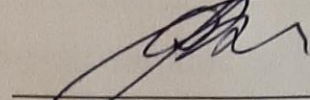
«10» 06 2021р.

Хмельницький 2021

Факультет програмування та комп'ютерних і телекомунікаційних систем
Кафедра телекомунікацій, медійних та інтелектуальних технологій
Спеціальність 172 – Телекомунікації та радіотехніка

ЗАТВЕРДЖУЮ

Завідувач кафедри ТМІТ



« 16 » 02 2021 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНИЙ ПРОЄКТ

дипломнику Свачій Олегу Ігоровичу

(Прізвище, ім'я, по батькові студента)

1 Тема проєкту Модуль керування параметрами системи Smart House на основі поведінки мешканців

керівник проєкту Полікаровських Олексій Ілліч д.т.н., професор

(Прізвище, ім'я, по батькові, науковий ступінь, учене звання)

Затверджено

наказом ректора університету від 5 02 2021 р. № 11

2 Строк подання студентом проєкту на кафедру « 1 » 06 2021 р.

3 Вихідні дані до проєкту

Мета роботи: розробка апаратного та програмного забезпечення для системи Smart House на основі аналізу поведінки мешканців.

Об'єкт дослідження: апаратна та програмна реалізація системи на основі прогнозування поведінки мешканців.

Предмет дослідження: апаратна та програмна реалізація методу машинного навчання у системах Smart House.

4 Термін подання роботи до захисту

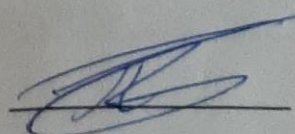
5 Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Вступ. Загальні відомості про системи Smart House, актуальність, існуючі рішення. Розробка структурної схеми системи Smart House із самонавчанням.

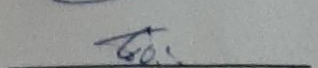
Опис власної реалізації системи Smart House з елементами самонавчання.

Висновок

Науковий керівник

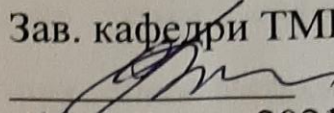


Завдання отримав



ЗАТВЕРДЖУЮ

Зав. кафедри ТМІТ


«10» лютого 2021 р.

ПЛАН ІНДИВІДУАЛЬНОЇ РОБОТИ

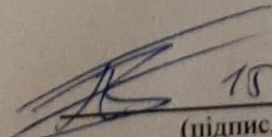
№ п/п	Найменування виду роботи	Форма звітності, термін виконання	Відмітка наукового керівника
1.	Вибір тематики	15.02.2021р.	Виконано
2.	Аналіз початкових даних	16.02.2021р.	Виконано
3.	Написання вступу та 1 розділу	11.03.2021р.	Виконано
4.	Написання 2 розділу	27.03.2021р.	Виконано
5.	Написання 3 розділу та висновку	14.04.2021р.	Виконано
6.	Корекція зауважень керівника проекту	30.04.2021р.	Виконано
7.	Оформлення кваліфікаційного проекту	10.05.2021р.	Виконано
8.	Рецензування кваліфікаційного проекту	15.05.2021р.	Виконано
9.	Підготовка доповіді та презентаційних матеріалів	19.05.2021р.	Виконано

Примітки:

1. _____


2. _____

Науковий керівник


15.02.21
(підпис, дата)

Олексій ПОЛКАРОВСЬКИХ

Студент


(підпис, дата)

Олег СВАЧІЙ

АНОТАЦІЯ

Кваліфікаційний проект бакалавра на тему «Модуль керування параметрами системи Smart House на основі поведінки мешканців» виконаний студентом групи 4-го курсу ТР-17-1 Свачій Олегом Ігоровичом на кафедрі «Телекомунікаційні, медійні та інтелектуальні технології» Хмельницького національного університету у 2021 р. Керівник роботи проф.каф. Полікаровських Олексій Ілліч.

Проект складається із вступу, 3-х розділів, основних висновків по роботі, списку використаних джерел (9 бібліографічних посилань, 1 сторінка) та 4 додатків (10 сторінок). Загальний обсяг роботи, в якому викладено основний зміст, складає 69 сторінок. Повний обсяг роботи 79 сторінок.

Кваліфікаційний проект присвячена розгляду питань створення системи розумного будинку з елементами машинного навчання на основі поведінки мешканців. В роботі висвітлено наступні питання: проведено аналіз існуючих підходів створення систем розумного будинку; проведено аналіз і класифікація методів машинного навчання. Розроблено алгоритм прогнозування поведінки мешканця будинку з використанням методу лінійної регресії. Розроблено апаратний модуль та програмне забезпечення.

Ключові слова: метод лінійної регресії, машинне навчання, система Smart House.

ABSTRACT

Bachelor's qualification project on "Smart House parameters management module based on residents' behavior" was performed by 4th year student of TR-17-1 group Svachii Oleh Ihorovych at the Department of Telecommunications, Media and Intellectual Technologies of Khmelnytsky National University in 2021. Head works of prof. Polikarovskiykh Olexiy Ilyich.

The project consists of an introduction, 3 sections, main conclusions on the work, a list of sources used (9 bibliographic references, 1 pages) and 4 appendices (10 pages). The total volume of the work, in which the main content is stated, is 68 pages. Full volume of work 79 pages.

The qualification project is devoted to the issues of creating a Smart House system with elements of machine learning based on the behavior of residents. The paper covers the following issues: analysis of existing approaches to creating smart home systems; the analysis and classification of methods of machine learning is carried out. An algorithm for predicting the behavior of the occupant of the house using the method of linear regression has been developed. Hardware module and software developed.

Keywords: linear regression method, machine learning, Smart House system.

ЗМІСТ

ВСТУП 3

РОЗДІЛ 1

ЗАГАЛЬНІ ВІДОМОСТІ ПРО СИСТЕМИ SMART HOUSE,
АКТУАЛЬНІСТЬ, ІСНУЮЧІ РІШЕННЯ. 5

1.1 Загальні відомості про системи розумного будинку 5

1.2 Приклади існуючих технологій системи Smart House 9

1.3 Програмні засоби реалізації концепції розумного будинку 16

1.4 Апаратні засоби реалізації концепції розумного будинку 20

Висновки до першого розділу 23

РОЗДІЛ 2

РОЗРОБКА СТРУКТУРНОЇ СХЕМИ СИСТЕМИ SMART HOUSE ІЗ
САМОНАВЧАННЯМ 24

2.1 Огляд задач та сфер застосування машинного навчання 24

2.2 Розробка структурної схеми системи із підтримкою машинного
навчання 26

2.3 Аналіз вхідних даних системи машинного навчання. Лінійна регресія 34

2.4 Застосування нелінійної регресії у системі машинного навчання..... 39

2.5 Бібліотека TensorFlow і її використання у пропонованій системі 41

Висновки до другого розділу 42

<i>КПТР.2017019.01.01 ПЗ</i>								
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>	Модуль керування параметрами системи Smart House на основі поведінки мешканців	<i>Лім.</i>	<i>Лист</i>	<i>Листів</i>
<i>Розроб.</i>		<i>Свачій О.І.</i>	<i>В.І. Свачій</i>	<i>10.06.19</i>			1	67
<i>Перев.</i>		<i>Поликардовських О.І.</i>	<i>О.І. Поликардовських</i>	<i>10.06.19</i>				
<i>Н. Контр.</i>		<i>Вісничук</i>	<i>В.І. Вісничук</i>	<i>10.06</i>		<i>ХНУ ТР-17-1</i>		
<i>Затв.</i>		<i>Вісничук</i>	<i>В.І. Вісничук</i>	<i>10.06</i>				

РОЗДІЛ 3

РЕАЛІЗАЦІЯ СИСТЕМИ SMART HOUSE З ЕЛЕМЕНТАМИ

САМОНАВЧАННЯ	43
3.1 Апаратна реалізація системи розумного будинку	43
3.2 Розробка принципової схеми	50
3.3 Програмна реалізація машинного навчання, опис алгоритму та програмного забезпечення	58
Висновки до другого розділу	66
ВИСНОВКИ.....	67
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	68
ДОДАТКИ.....	69

					<i>КПТР.2017019.01.01 ПЗ</i>	<i>Лист</i>
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		2

ВСТУП

Метою роботи є розробка апаратного та програмного забезпечення для системи Smart House на основі аналізу поведінки мешканців.

В сучасному світі розвиток новітніх технологій проникає у всі сфери життєдіяльності людини, і це стосується не тільки промисловості чи сільського господарства. Важливе місце займає сфера автоматизації та спрощення побутових завдань, тому що це дозволяє людям значно підняти рівень життя.

Ми давно використовуємо автоматизовані системи в побуті, наприклад пральні машини, кавові апарати, чи кондиціонер. По суті наш будинок складається з певної комбінації автоматичних підсистем, складність і кількість яких можна міняти в залежності від наших потреб і бажань. Чим більше автоматизованих приладів тим нам комфортніше і якість життя ліпша.

На новий рівень автоматизацію вивів розвиток телекомунікаційних технологій, які дозволяють нам віддалено і безконтактно керувати приладами. Ці технології не тільки дозволяють економити час, а що набагато важливіше дозволяють нам не залежати від свого місцеположення. Такі можливості дозволили спостерігати за діями модуля і керувати його роботою віддалено використовуючи інтернет, це в свою чергу привело до виникнення концепції «Розумного будинку».

В сучасному світі «Розумний Будинок» має бути невід'ємною частиною житлового приміщення, на рівні з водопроводом, каналізацією, опаленням чи іншими здобутками прогресу. Житловий будинок представляє собою безліч варіацій інженерних рішень: опалення, пожежна сигналізація, вентиляція, освітлення, охоронна сигналізація, силова електрика, водопровід, кондиціонер, домашній кінотеатр та інше. Зрозуміло що обслуговування такої системи потребує не тільки постійного моніторингу всіх систем, а й комплексних рішень на основі даних отриманих від різних датчиків.

Людина фізично не здатна цілодобово відслідковувати всю інженерію в своєму будинку, але з цим прекрасно справиться система «Розумний будинок»

					<i>КПТР.2017019.01.01 ПЗ</i>	<i>Лист</i>
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		3

і гарантуватиме нам комфортне і безпечне проживання на протязі всього строку експлуатації будівлі.

Визначення «Розумний будинок» передбачає не просто систему побутової автоматизації, яка виконує відносно прості сценарії керування приладами і інженерними системами, а й має оптимізувати і забезпечити скоординовану їх роботу

Робота «Розумного дому» базується на основі сценаріїв, рівень складності яких може коливатись від перемикання найпростіших автоматів до використання компонентів штучного інтелекту. Всі без виключення системи «Розумного дому» використовують покази датчиків в алгоритмах цих сценаріїв. Також мають бути реалізовані компенсаторні варіанти роботи системи, коли при виході з ладу окремих приладів, система продовжує працювати в альтернативних режимах роботи.

Концепція «Розумного будинку» крім підвищення комфорту і оптимізації витрат, робить наше життя значно безпечнішим, через цілодобовий моніторинг різноманітних датчиків. Що дає можливість убезпечити себе, наприклад від пожежі чи витoku газу. В будинках де проживають люди з обмеженими можливостями, по здоров'ю легко реалізувати спеціалізовані системи адаптовані під різні категорії обмежень: порушення слуху, зору, опорно-рухові порушення та інші.

Очевидно, що «Розумний дім» повинен функціонувати в залежності від «стилю життя» власника будинку і достатньо гнучко реагувати на його зміни. Тобто враховувати різницю між робочими і вихідними днями, зміною пори року і так далі. Також має бути можливість примусової зміни сценарію, наприклад при приїзді гостей, чи коли дім на тривалий час залишається без жильців (відпустка) .

					<i>КПТР.2017019.01.01 ПЗ</i>	<i>Лист</i>
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		4

РОЗДІЛ 1

ЗАГАЛЬНІ ВІДОМОСТІ ПРО СИСТЕМИ SMART HOUSE, АКТУАЛЬНІСТЬ, ІСНУЮЧІ РІШЕННЯ.

1.1 Загальні відомості про системи розумного будинку

Технології систем Smart House мають великий потенціал та вже на даний час приносить багато зручностей користувачам(рис 1.1). Енергоефективність, розваги, здоров'я, безпека, зручності, усі ці аспекти життя, на сьогоднішній день можуть полегшувати існуючі на ринку системи розумних будинків.

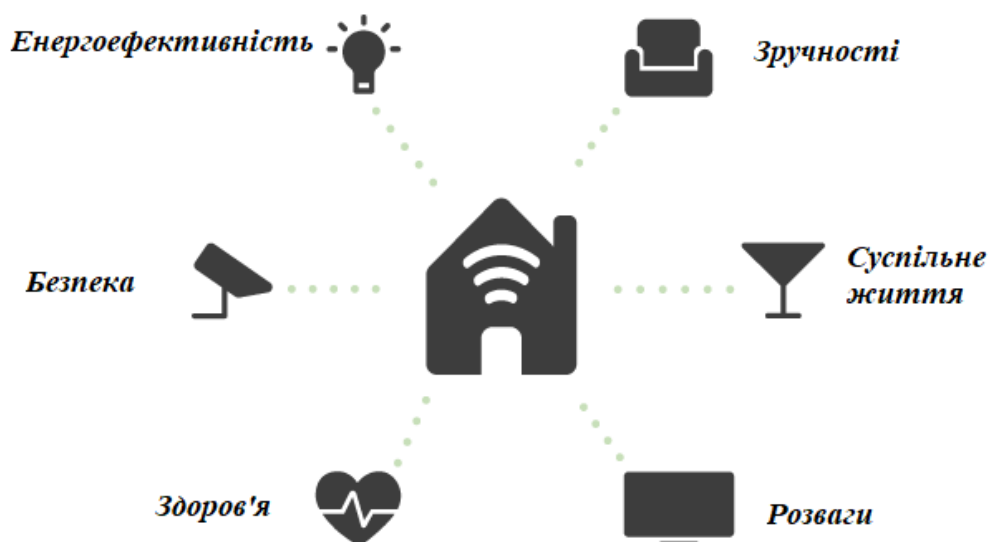


Рисунок 1.1 – Переваги сучасних розумних будинків

На початку розвитку цієї сфери зустрілось багато труднощів, особливо в плані сприйняття масою безпеки та надійності таких систем. Існує досить багато факторів життя у розумних будинках, не зрозумілих для маси людей. Такими факторами є конфіденційність, безпека, надійність та невинновдано велика ціна. Також, окрім цього досить часто буває, що очікування від таких систем завищенні, у порівнянні з справжнім функціоналом.

Напротязі останнього десятиліття цікавість людей до технологій розумних будинків дуже стрімко зросла, тому що розвиток технологій швидко збільшується.

Як видно, технологічний розвиток впливає не тільки на якість роботи підприємств або покращення можливостей воєнної техніки, а й також близько торкається життя людей.

Очевидно, що саме розвиток таких індустрій, як індустрія мобільних телефонів та інтернет реклами, збільшила рівень потреб суспільства, що й дало поштовх розвитку інноваційних ідей, до яких входять системи Smart House, змінюючи наші вимоги до традиційного будинку.

Очікування людей, щодо домашньої автоматизації дедалі зростають. Прилади які виконують прості завдання більше не цікавлять споживачів, наразі вимагається щоб дані прилади були частиною розподіленої системи.

Ринок систем розумного будинку не є ідеальним на сьогоднішній день, тому, ще потрібно пройти довгий шлях, щоб зрозуміти і адаптувати цей тип автоматизації життя до того рівня, що очікують споживачі. Розумні будинки відіграватимуть значну роль у майбутньому. В даний час вони все ще перебувають на початковій фазі, намагаючись охопити масового споживача.

Наразі попит невеликий, оскільки існуючі функціональні можливості пристроїв здебільшого обмежені простими діями, які додають складності, а не зручності для користувача [1].

Виробники продовжують будувати власні технологічні екосистеми, виключаючи загальні правила побудови програмного забезпечення. Одним з прикладів проблем таких екосистем , це невідповідність між протоколами різних пристроїв. Дана проблема значно стримує потенціал користувачів та скоує розширення можливостей розумних систем будинку. Отже, основною проблемою є те, що пристрої обмежені набором функцій що заклав виробник.

Smart House можна охарактеризувати як житловий простір, що оснащений сучасними технологічними датчиками, приладами та пристроями,

якими можна дистанційно керувати, отримувати до них доступ, та здійснювати моніторинг з метою надання послуг мешканцям будинку.

Даний простір є будь-яке побутове середовище, незалежно від його розміру та розташування. Датчики здатні виявляти переміщення та розташування предметів та людей, збираючи та відображаючи необхідні дані. «Розумність» системи досягається завдяки підключенню пристроїв через те, що зазвичай називають інтернет-речей (Internet of Things – IoT).

Елементи розумного будинку можна розбити на три групи;

- внутрішньо-мережеві;
- інтелектуальна керування;
- домашня автоматизація.

Внутрішня мережа може бути дротовою або бездротовою. Це є тією частиною системи, до якої пристрої підключаються один одного.

Інтелектуальне керування - надсилає та отримує інформацію та діє як пристрій сполучення між користувачем та пристроями.

Домашня автоматизація, стосується реальних пристроїв, що виконують поставлені завдання та пов'язують всю систему як всередині так і поза будинком.

Для швидшого впровадження технології Smart House, потрібно вирішувати безліч проблем. Одна з них полягає в тому, що загальний вигляд периферії розумного будинку повинен вписуватися в дизайнерську та технологічну архітектуру існуючого будинку [1].

Також існує проблема, яка полягає у тому, що від користувача не потрібно вимагати набувати досвіду для управління всією системою. Навпаки, система повинна адаптуватися до розпорядку дня користувача. Крім того, нові пристрої, що користувач захоче інтегрувати в існуючу мережу, повинні без труднощів здійснювати взаємозв'язок з іншими пристроями. Але оскільки виробники постійно збільшують кількість протоколів зв'язку для своєї продукції, що не є сумісними з продукцією інших виробників. Отже, варто

					<i>КПТР.2017019.01.01 ПЗ</i>	<i>Лист</i>
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		7

наголосити на тому, що є необхідність використання універсальних стандартів протоколів зв'язку між пристроями.

Інша відома проблема стосується надійності програмного забезпечення та їх алгоритмів. Тут головним недоліком є не завжди точна обробка зібраних даних. Що у подальшому може призвести до некоректного прогнозування поведінки людини. Особливо це стосується домашньої системи безпеки, де датчики не завжди можуть коректно виявляти різні фактори. Також постійний збір даних та їх аналіз, спричиняє зростання занепокоєння користувачів щодо безпечного збереження конфіденційної інформації.

На нинішньому етапі автоматизація будинків зустрінеться ще з численними проблемами, поки нарешті не буде вбудованим в наше повсякденне життя. Крім того, враховуючи складність сучасних систем Smart House, користувачі відчують загрозу від можливих атак, що можуть зашкодити як будівлі так і власнику.

Досить висока ціна на пристрої також є впливовим фактором. Але не зважаючи на це люди все рівно знаходили переваги в дистанційному керуванні та моніторингу споживання енергії, хоча часто функції енергозбереження відсутні у більшості виробників. Очікувана вигода від використання пристроїв розумного будинку повинна бути чіткою. Тобто, технологія розумного будинку має перейти від пристроїв, які «допомагають» користувачеві, до пристроїв, які необхідні користувачам у їхньому житті. Отже галузь систем Smart house почне стрімко розвиватись, якщо вона буде фінансово доступною для користувачів та якщо вони довіряться своїм «розумним будинкам». Також системи Smart House повинні вирішити проблеми безпеки даних та забезпечити повну конфіденційність [1].

					<i>КПТР.2017019.01.01 ПЗ</i>	<i>Лист</i>
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		<i>8</i>

1.2 Приклади існуючих технологій системи Smart House

Сучасні системи розумного будинку можна розділити на дві групи. Комерційні, це компанії що продають пристрої розумного будинку, програмне забезпечення яких сумісне з елементами тільки даної компанії. В іншу групу можна віднести платформи домашньої автоматизації будинку, що є безкоштовними та у вільному доступі.

Перелік провідних в Україні компаній систем «розумний будинок»

1. Вітчизняна система Ajax(рис 1.2)

Виробником системи є Україна. Дана система гарантує безпеку житла контролюючи пожежну, газову, електричну та інші можливі загрози для будинку.



Рисунок 1.2 – Зображення обладнання компанії Ajax System

Обладнання «Розумний будинок» Ajax використовує надійно зашифровану і захищену двосторонню радіозв'язку «Jeweller», яку розробили

					<i>КПТР.2017019.01.01 ПЗ</i>	Лист
Зм.	Лист	№ докум.	Підп.	Дата		9

власноруч. Особливістю є повна автономність від електромережі завдяки резервного джерела живлення. Усі прилади мають особливий та стильний дизайн.

Перевагами є:

- простий монтаж;
- бездротовий канал зв'язку між системними елементами;
- Wi-Fi і GSM-зв'язок;
- наявність захисту від зняття будь-якого з датчиків;
- можливий доступ інших користувачів;
- автономна робота хаба від акумулятора (до 16 годин);
- велика кількість способів інформування користувача;
- велика зона дії сигналу (до 2000 м);
- підключення до 100 пристроїв;
- розетка відображає витрати електроенергії (з урахуванням підключених приладів), є функція автоматичного відключення при перепадах напруги;
- невисока вартість комплекту (від 200\$).

Недоліки:

- функціонування тільки з роботою центрального контролера, відсутність автономності датчиків;
- відсутність власної камери відеоспостереження (є можливість підключення обладнання).

На сьогодні обладнання Ajax - це найкраща система в нашому вітчизняному рейтингу. Дана система багатфункціональна, надійна, зручна, компактна. Має чудовий дизайн, якісний захист від зломів і зрозумілий інтерфейс. Налаштування та встановлення комплектів та наборів системи є максимально спрощеним для того щоб, бути зрозумілим звичайному користувачу. Обладнання системи Ajax є відносно недорогим, що є ще одною перевагою [2].

					<i>КПТР.2017019.01.01 ПЗ</i>	<i>Лист</i>
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		<i>10</i>

2. BroadLink(рис 1.3)

Виробник – Китай. За замовчуванням немає українського інтерфейсу, але є версії прошивок з підтримкою мови.



Рисунок 1.3 – Зображення обладнання компанії BroadLink

Пристрої BroadLink це комплект цифрових пристроїв, створених для раціонального керування побутовою технікою, а також енергетичної, освітлювальної, охоронної та іншими системами будинку. Кожен елемент такого комплексу може працювати як самостійно, так і взаємодіяти один з одним.

Перевагами є:

- має широкий асортимент датчиків;
- автономна робота датчиків (функціонує без центрального хаба);
- є фірмова камера відеоспостереження;
- вартість обладнання (від 200\$);
- швидко налаштовується;
- контролюється по Wi-Fi через Інтернет з будь-якої точки планети.

Недоліки:

- відсутність резервного живлення хаба;
- невелика відстань дії сигналу (до 50 м).

					КПТР.2017019.01.01 ПЗ	Лист
Зм.	Лист	№ докум.	Підп.	Дата		11

Така система не вимагає наявності центрального контролера, адже всі його пристрої хоч і взаємопов'язані, але можуть працювати незалежно одне від одного. Побутова техніка в будинку налаштовується згідно зі сценаріями роботи в додатку на смартфоні [2].

3. Fibaro (рис 1.4)

Виробник – Польща. В більшості орієнтовано на Європу та Америку. Тому важко знайти русифіковану версію інтерфейсу.



Рисунок 1.4 – Зображення обладнання компанії Fibaro

В порівнянні з іншими виробниками, обладнання Fibaro має найкращу наповненість датчиками для контролю стану приміщень та керування домашньою технікою. Однак встановити і розібратися з такою системою може тільки професіонал.

Переваги:

- розетка відображає рівень енергоспоживання підключених пристроїв, а також є функція вимкнення при скачках напруги;
- велика кількість сценаріїв для користувача;
- чудове наповнення системи великою кількістю датчиків і пристроїв;
- розсилка повідомлень відразу на кілька телефонів;
- датчик протікання містить сирену;

- голосове керування через сервіс Google(англійською мовою);
- за рахунок можливості кожного її елемента бути ретранслятором сигналу, невелика дальність сигналу системи збільшується;
- наявність фірмової камери відеоспостереження.

Недоліки:

- монтаж і налаштування можуть здійснюватися тільки професійним персоналом;
- дальність сигналу (до 50 м без перешкод);
- висока вартість (від 600\$);
- обов'язкове підключення Fibaro Home Center до інтернету;
- необхідність в обов'язковій установці програмного забезпечення на ПК, а також урізане мобільний додаток;
- відсутність резервного живлення;
- неможливість функціонування без центрального хаба.

Система Fibaro відноситься до професійного обладнання по забезпеченню автоматизації та безпеки будинку з найширшим функціоналом [2].

4. Orvibo (рис 1.5)

Виробник – Китай. Відсутній український та російський інтерфейс, що ускладнює встановлення та налаштування.

Orvibo – це недорогий комплект простого в експлуатації обладнання, що забезпечує безпеки будинку.

Orvibo Security Kit – це обладнання, що знаходиться між класичною системою охорони приміщення та системою Smart House. Має просте і зрозуміле керування, можливості розширення бази пристроїв сторонніх виробників за рахунок стандартного протоколу ZigBee. Однак з метою зробити цю систему більш дешевою, тут використовуються досить примітивні датчики без захисту від злому, а камера розрахована тільки на роботу всередині приміщення.



Рисунок 1.5 – Зображення обладнання компанії Orvibo

Переваги [2]:

- Wi-Fi зв'язок з телефоном, до 10 номерів;
- автоматичне знаходження і підключення сенсорів до центрального контролеру;
- доступна вартість (від 150\$);
- простота в монтажі і підключенні, віддалений контроль через додаток на смартфоні;
- бездротовий протокол взаємодії між контролером і датчиками (ZigBee);
- наявність фірмової відеокамери;
- великий вибір пристроїв та змога масштабування системи (близько 100 датчиків), причому інших виробників.

Недоліки:

- дротове підключення до Інтернету (для надійності роботи системи);
- відсутність резервного живлення хаба на випадок відключення електроенергії;
- малий набір пристроїв в базовій комплектації;
- невелика зона дії сигналу (до 30 м).

5. Xiaomi (рис 1.6)

Виробник – Китай. Відсутній український та російський інтерфейс, що ускладнює встановлення та налаштування.



Рисунок 1.6 – Зображення обладнання компанії Xiaomi

Набір Xiaomi є доскональною стартовою платформою для підключення інших датчиків та пристроїв, в тому числі сторонніх виробників. Елементи системи працюють як самостійно, так і як єдине ціле. Можна створити досить функціональну систему контролю житлового простору, враховуючи безпеку. Дана система, беручи до уваги її вартість, підходить для ознайомлення з системами Smart House [2].

Переваги:

- наявність власної камери відеоспостереження;
- бездротовий протокол ZigBee;
- зручне керування за допомогою смартфона через Wi-Fi;
- низька вартість комплекту (всього 90\$);
- наявність різноманітних сценаріїв;

					<i>КПТР.2017019.01.01 ПЗ</i>	<i>Лист</i>
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		15

- повна автономність пристроїв;
- компактність і стильний дизайн.

Недоліки:

- відсутність резервного живлення центрального контролера;
- маленька зона дії сигналу (до 10 м);
- малий набір сенсорів і виконавчих пристроїв в базовому наборі.

1.3 Програмні засоби реалізації концепції розумного будинку

Існує безліч платформ домашньої автоматизації будинку, програмний код яких є безкоштовними та у вільному доступі. Переваги забезпечення з відкритим доступом полягає в тому що воно підтримується масою людей, і кожен раз оновлюється.

Нижче буде перелік платформ з відкритим кодом, які є найвідомішими:

1. *openHAB*(рис. 1.7)

Маючи майже півмільйона дописів на своїх форумах та 33 000 учасників, openHAB постійно вдосконалює свої початкові пропозиції. Платформа може інтегруватися з понад 1500 пристроїв, таких як Sony, Pioneer, LG, Samsung та багато інших.



Рисунок 1.7 – Зображення варіанту інтерфейсу платформи openHAB

openHAB - це безкоштовне програмне забезпечення, сервер для якого можна завантажити локально у будь-яку операційну систему. Для систем створених на даній платформі немає потреби у хмарних серверах [3].

2. Home Assistant(рис. 1.8)

Home Assistant - ще одна сильно розвинена спільнота домашньої автоматизації з відкритим кодом, що пропонує платформу, яка забезпечує контроль приладів та саме важливе - конфіденційність.

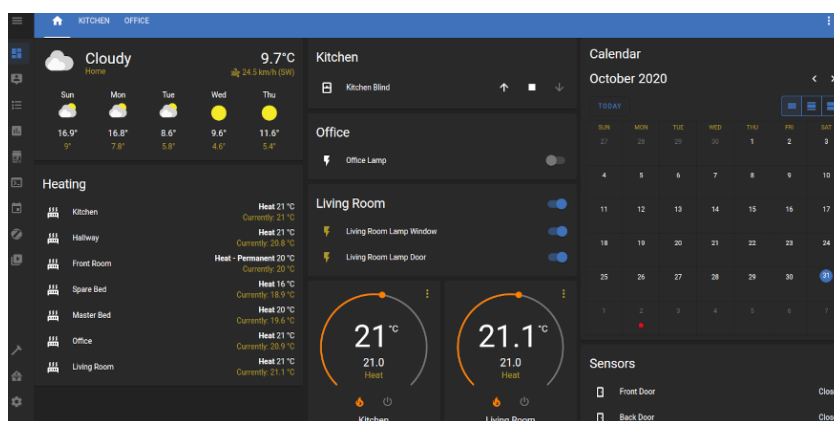


Рисунок 1.8 – Зображення варіанту інтерфейсу платформи Home Assistant

Багато в чому Home Assistant можна порівняти з openHAB в тому сенсі, що в них схожий інтерфейс, що дає можливість інтегрувати якомога більше пристроїв у зручному вигляді. Якщо дивитись з боку розробки, Home Assistant, мабуть, є найбільш гнучким з усіх платформ [3].

3. Jeedom (рис. 1.9)

Jeedom - чудовий варіант домашньої автоматизації, розвиток якого, на жаль, стримується мовою. Спільнота Jeedom є переважно на французькій мові, і навіть при тому що веб-сайт перекладено англійською мовою, їх принципи керування та форуми написані французькою мовою. Jeedom пропонує багато мов інтерфейсу, включаючи англійську, іспанську та німецьку. Однак, використання даного відкритого програмного забезпечення дещо складніше ніж openHAB та Home Assistant [3].

для початківців, яка допомагає тим, хто не знайомий з автоматизацією будинку, звикнути до різних концепцій. Наразі PiDome переписується, а поточна версія трохи застаріла [3].



Рисунок 1.11 – Зображення варіанту інтерфейсу платформи PiDome

1.4 Апаратні засоби реалізації концепції розумного будинку

Апаратні реалізації системи можна поділити такі, що розробляються фірмами виробниками та виконані самостійно. Використовуючи придбані прилади ми можемо користуватись як програмами виробника, так і відкритими програмними забезпеченням. Як правило у таких ПЗ є можливість, на сайті платформи завантажити доповнення на сервер розумного будинку, які відповідатимуть за підключення придбаного пристрою до системи.

Плюсами такого методу є:

- заощадження часу на апаратну та програмну реалізацію пристрою;
- легкість завантаження доповнень;
- легкість підключення придбаного пристрою.

Недоліки:

- не на всі пристрої є написані додаткові ПЗ;
- велика ціна.

Саморобні пристрої на сьогодні досить легко розробляти, завдяки зменшенню конструкторської роботи використовуючи безліч плат розробників, таких як: серія Arduino, STM плати, ESP8266, модулі датчиків, модулі драйверів виконавчих пристроїв та інші. Саме завдяки модульності апаратної частини саморобних пристроїв та великої кількості бібліотек та літератури в інтернеті, розробка елементів системи на сьогоднішній день займає набагато менше часу.

Невід'ємною частиною створення пристроїв, особливо у середовищах openHAB та Home Assistant є перелічені нижче модулі та плати розробника.

1) Мікрокомп'ютер Raspberry Pi.(рис. 1.12)

Даний мікрокомп'ютер досить часто виступає в ролі сервера системи Smart House. Окрім цього він може також паралельно виконувати інші операції, наприклад обробка баз даних датчиків та на мові Python, здійснювати навчання системи передбачувати певні сценарії.

					<i>КПТР.2017019.01.01 ПЗ</i>	<i>Лист</i>
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		20



Рисунок 1.12 – Зображення мікрокомп'ютера Raspberry Pi

2) Плата розробника Arduino.

Як правило Arduino досить часто використовується у системах openHAB та Home Assistant, адже є відповідні додаткові програмні забезпечення які відповідають за спілкування з сервером.

В даному випадку спілкування не може здійснюватися без Ethernet Shield (рис. 1.13), що здійснюватиме зв'язок з центральним контролером через локальну мережу.



Рисунок 1.13 – Зображення плати розробника Arduino Uno у поєднанні з Ethernet Shield

3) Плата розробника ESP8266 (рис. 1.14).

Так само як і Arduino, досить часто використовується у системах openHAB та Home Assistant. Однією з переваг ESP8266 над Arduino, є наявність можливості бездротової передачі даних по мережі WiFi.

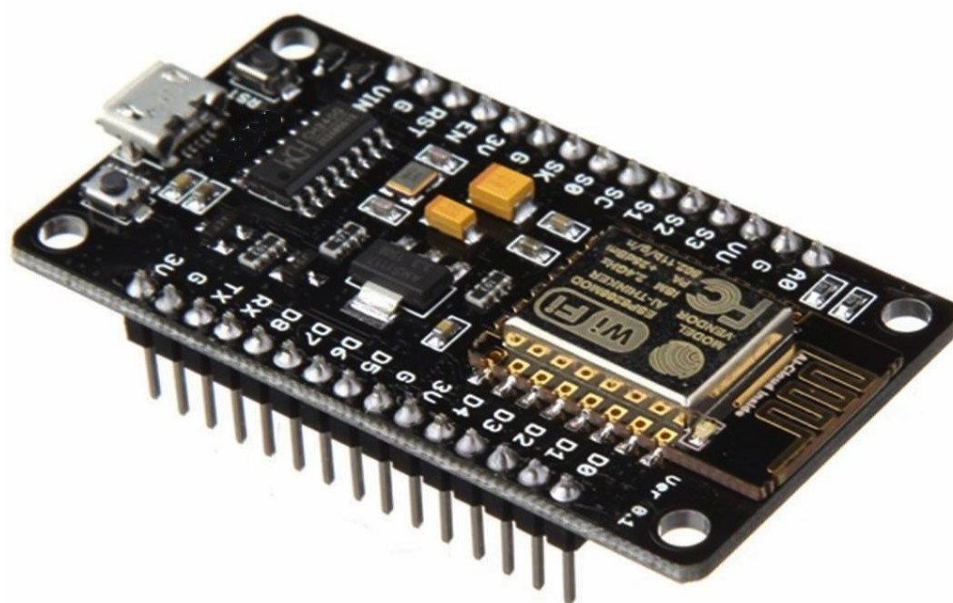


Рисунок 1.14 – Зображення плати розробника ESP8266

Серед перелічених вище апаратних засобів через низьку вартість та наявність усіх необхідних можливостей, для реалізації допоміжних блоків керування було вибрано плату розробника Arduino Uno у поєднанні з Ethernet Shield.

Висновки до першого розділу

У першому розділі описані загальні відомості про системи розумного будинку, їх становище на сьогоднішній день.

Вказано переваги та недоліки систем сучасного рівня. Було проведено аналіз та наведення прикладів існуючих систем Smart House. Вказано аналоги вітчизняних систем та закордонних.

Оглянуто засоби програмної та апаратної реалізації сучасних концепцій розумних будинків. Описана проблема надійності систем розумного будинку.

Обґрунтовано вибір варіантів програмної та апаратної реалізації системи, що є завданням кваліфікаційної роботи.

					<i>КПТР.2017019.01.01 ПЗ</i>	<i>Лист</i>
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		23

РОЗДІЛ 2

РОЗРОБКА СТРУКТУРНОЇ СХЕМИ СИСТЕМИ SMART HOUSE ІЗ САМОНАВЧАННЯМ

2.1 Огляд задач та сфер застосування машинного навчання

Машинне навчання - це програма штучного інтелекту, що надає системам можливість автоматичного навчання та вдосконалення своїх дій. Машинне навчання фокусується на розробці комп'ютерних програм, які можуть отримувати доступ до даних та використовувати їх для самостійного навчання.

Процес навчання починається із спостережень, прикладів та інструкцій виконання певних задач, щоб шукати закономірності в отриманих даних та приймати кращі рішення в майбутньому на основі вже проробленої роботи. Основна мета машинного навчання - дозволити системам навчатися самостійно, без втручання та допомоги людини.

Але, використовуючи класичні алгоритми машинного навчання, програмний код розглядається як послідовність ключових слів, це може імітувати людську здатність розуміти значення тексту.

Алгоритми машинного навчання часто класифікуються як контрольовані або неконтрольовані.

Керовані алгоритми машинного навчання можуть застосовувати вивчене в минулому до нових даних для прогнозування майбутніх подій. Аналізуючи вже відомий набір даних, що у даному випадку називаються даними навчання, алгоритм виробляє функцію для прогнозування вихідних значень. Алгоритм навчання також може порівнювати свої результати з правильним, передбаченим результатом та знаходити помилки, щоб відповідно модифікувати модель навчання [5].

					<i>КПТР.2017019.01.01 ПЗ</i>	<i>Лист</i>
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		24

На відміну від цього, некеровані алгоритми машинного навчання використовуються, коли інформація, яка використовується для навчання, не класифікується. Навчання без нагляду вивчає, як системи можуть використати функцію. Система не визначає правильний результат, але вона досліджує набори даних і може робити з них висновки.

Підсилення алгоритмів машинного навчання - це метод навчання, який взаємодіє із середовищем з яким взаємодіє система, виконуючи дії, виявляючи вірні рішення та навіть помилки. Пошук методом виконання практичних завдань та обробки помилок - найважливіші у посиленні здатності навчання. Цей метод дозволяє машинам та програмам автоматично визначати ідеальну дію в певному контексті, щоб максимізувати ефективність.

Машинне навчання дозволяє аналізувати велику кількість баз даних. Як правило, це забезпечує точніші та швидші результати, що спричиняє самостійному виявленню як ризиків так і покращень системи. Варто зазначити, що для очікуваних результатів, на навчання системи може знадобитися додатковий час та затрати [2].

Створення алгоритмів в даній сфері - непросте завдання. Перед тим як "Навчити" систему, розробник повинен отримати та оцінити великий обсяг даних, та вручну класифікувати кожний набір. Створення нових елементів програм навчання - це дуже трудомісткий процес, а не разова операція. Це є необхідністю, тому що програмна реалізація алгоритму створює більше параметрів, необхідних для отримання ключової інформації та її розподілення відповідно до ваших потреб.

Тільки після обробки та фільтрування безлічі інформації і аналізу, система розпізнає основні умови її роботи. Та навіть у такому випадку немає гарантії, що ви досягнете бажаних результатів виконання завдання системи [3].

2.2 Розробка структурної схеми системи із підтримкою машинного навчання

Серед всіх перелічених вище варіантів програмної реалізації системи Smart House. Було обрано платформу, що має найбільшу базу спільнот та обговорень на форумах. Цим критеріям відповідає безплатне програмне забезпечення openHAB(рис. 2.1)



Рисунок 2.1 – Емблема платформи openHAB

У даної платформи є підтримка ПЗ на основних операційних системах:

- Raspberry Pi;
- Linux;
- Windows;
- macOS.

Існує декілька варіантів встановлення програмного пакету на Raspberry Pi:

1) Варіант 1 – встановлення операційної системи openHABian

Встановлення розробленої платформою операційної системи openHABian, що розроблена виключно для роботи мікрокомп'ютера як сервера. Де програмне забезпечення може автоматично додавати пристрої у структуру розумного будинку.

2) Варіант 2 – автоматичне встановлення openHABian поверх існуючої операційної системи

					КПТР.2017019.01.01 ПЗ	Лист
Зм.	Лист	№ докум.	Підп.	Дата		26

У даному випадку користувач встановлює у систему Debian програмний пакет openHABian. Перевагою даного способу є те, що мікрокомп'ютер може паралельно виконувати запуск інших програм.

Завдяки перевазі другого варіанту було обрано саме такий спосіб встановлення openHAB, адже у кваліфікаційній роботі буде проводитись розрахунок для якого потрібне значення потужність мікрокомп'ютера. А написання програми буде відбуватись на мові програмування Python, яка у системі повинна буде працювати паралельно від роботи openHAB

Серед описаних вище методів апаратної реалізації очевидно, що дана кваліфікаційна робота буде виконуватись з допомогою саморобних пристроїв на базі плат розробника, перехідних плат та різних модулів. Даний метод має ряд плюсів:

- гнучкість у роботі пристроїв;
- можливість безмежної варіації видів пристроїв;
- економічний виграш;
- велика база допоміжної літератури.

Реалізація системи Smart House відбувається за допомогою структурних елементів, що розроблялись та підбирались власноруч.

Основними елементами розумного будинку є маршрутизатор та контролер. Більшість систем не працюють без даних складових.

У цій кваліфікаційній роботі функцію контролера виконують декілька структурних одиниць описаних нижче.

Raspberry Pi виконує (рис. 2.2) роль сервера та основної обчислювальної потужності системи. Raspberry може зв'язуватись з маршрутизатором, як через Ethernet кабель так і по каналу WiFi.

Мікрокомп'ютер отримуватиме інформацію від блоків керування 1-3. Отримання даних відбуватиметься по протоколу MQTT (Message Queue Telemetry Transport) протоколу. Після чого програма написана на мові Python буде оцінювати дані та реалізовувати навчання системи.

передачу даних між блоками та головним контролером за допомогою MQTT протоколу передачі даних.



Рисунок 2.4 – Зображення плати Ethernet Shield

Для кращого розуміння структури системи Smart House, що виконана у кваліфікаційній роботі на рис. 2.5 зображено структурну схему.

На структурній схемі можна побачити усі одиниці розумного будинку. Нижче перелічені всі структурні елементи системи:

- Контролер Smart House;
- Маршрутизатор;
- Блоки керування 1-3;
- Блок комутації;
- Блок керування температурою котла;
- Реле (13 шт.);
- Кінцевий вимикач (3шт.);
- Датчик вологи та температури (2шт.);
- Датчик присутності(ІЧ) (2шт.);
- Інфрачервоний датчик руху;
- Датчик витоку газу (2шт.);
- Датчик вологи (3шт.);

- Датчик пожежі;
- Датчик диму;
- Пристрій керування відкриття вікна (2шт.);
- Пристрій керування заслінкою вентиляції.

Більшість елементів, що використовуються для реалізації машинного навчання. До ключових елементів можна віднести такі структурні одиниці:

- Кінцевий вимикач відкриття дверей;
- Інфрачервоний датчик руху;
- Датчик присутності.

Також у даний список можна включити програмний моніторинг наявності в мережі локальної IP – адреси смартфона власника квартири. Ці дані головний контролер програмно враховуватиме, як основний чинник появи людини у квартирі.

					<i>КПТР.2017019.01.01 ПЗ</i>	<i>Лист</i>
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		<i>30</i>

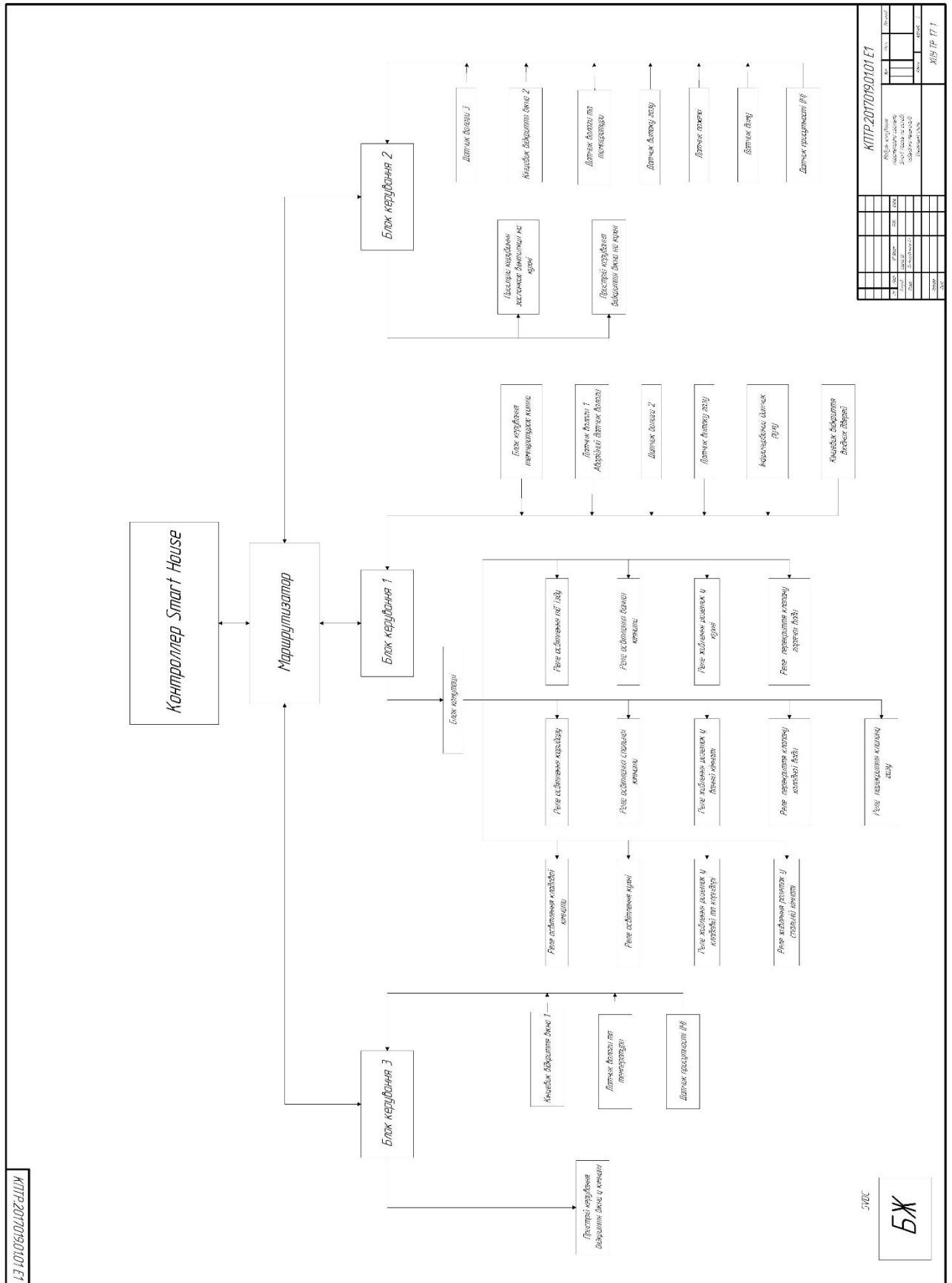


Рисунок 2.5 – Зображення структурної схеми системи Smart House кваліфікаційної роботи

Зм.	Лист	№ докум.	Підп.	Дата

- 2 – Кухня**
 - 2.1 – Обстановка
 - 2.2 – Обстановка над ступ.
 - 2.3 – Виробничі мебелі/об'єкти
- 2 – Ванна**
 - 2.1 – Обстановка
 - 2.2 – Живильні розетки
 - 2.3 – Живильні розетки
 - 2.4 – Корпусна конструкція
 - 2.5 – Корпусна конструкція/панель
 - 2.6 – Димачі, розетки/панель
 - 2.7 – Димачі, вентиляція ступ.
 - 2.8 – Виробничі мебелі/об'єкти

- 4 – Кухня**
 - 4.1 – Живильні розетки
 - 4.2 – Обстановка
 - 4.3 – Корпусна конструкція
 - 4.4 – Димачі, розетки/панель
 - 4.5 – Корпусна конструкція
 - 4.6 – Живильні розетки
 - 4.7 – Димачі, розетки/панель
 - 4.8 – Димачі, вентиляція ступ.
 - 4.9 – Виробничі мебелі/об'єкти
 - 4.10 – Димачі, розетки/панель
 - 4.11 – Димачі, розетки/панель
- 5 – Спальня**
 - 5.1 – Обстановка
 - 5.2 – Живильні розетки
 - 5.3 – Димачі, розетки/панель
 - 5.4 – Димачі, вентиляція ступ.
 - 5.5 – Виробничі мебелі/об'єкти
 - 5.6 – Димачі, розетки/панель

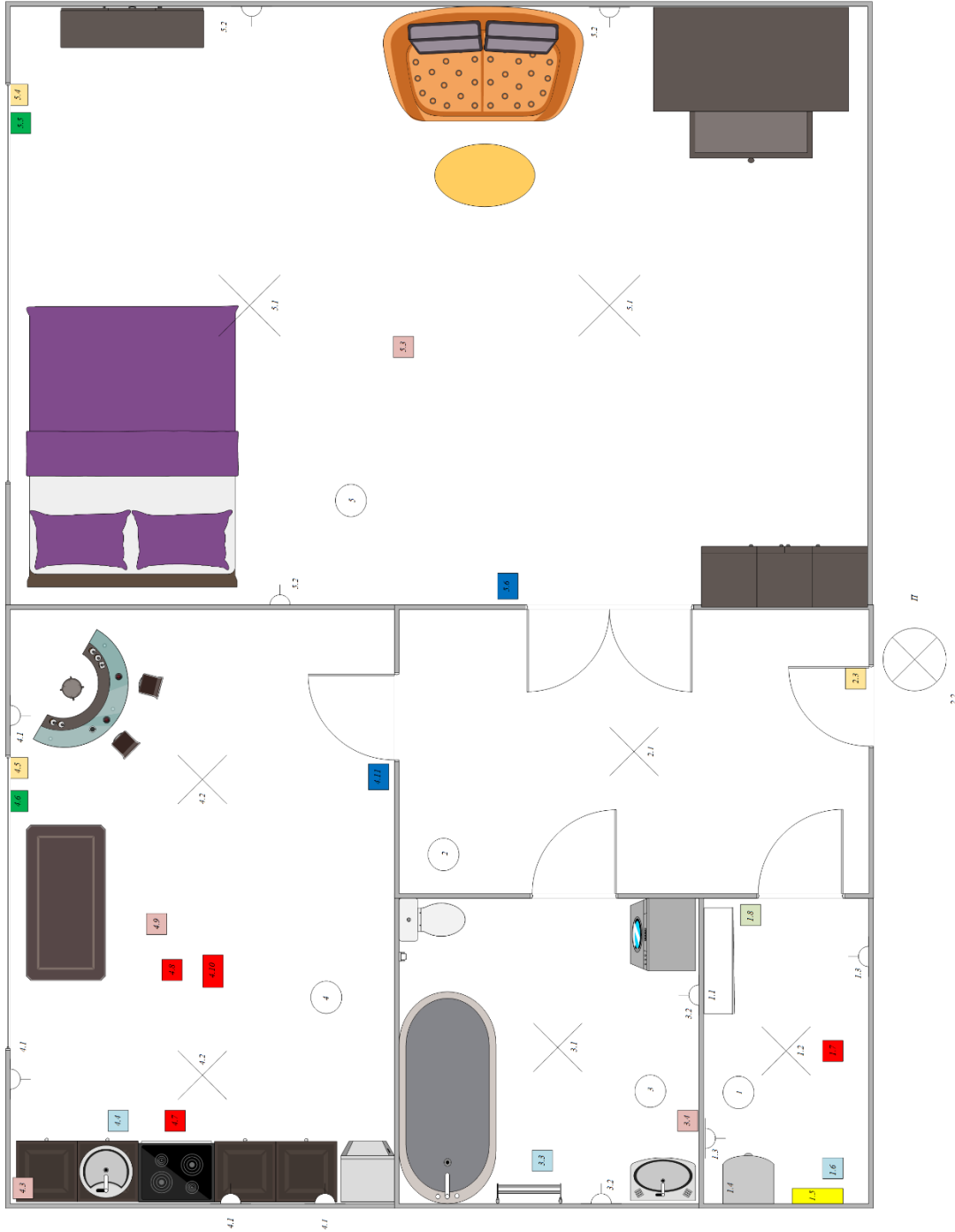
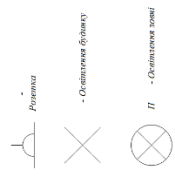


Рисунок 2.6 – Зображення планування квартири

Зм.	Лист	№ докум.	Підп.	Дата

КПТР.2017019.01.01 ПЗ

Для розуміння плану житлового приміщення у якому встановлена система Smart House було виконано планування квартири, що зображено на рис. 2.6. Дане планування обрано за основу у написанні алгоритмів машинного навчання та програмного коду керування структурними одиницями.

Завданням кваліфікаційної роботи є встановлення розумного будинку у однокімнатну квартиру. Варто зазначити, що машинне навчання системи виконується за умови проживання у квартирі однієї людини. При невиконанні умов контролер системи, на даному етапі розробки, не може здійснювати навчання коректно. Для виконання завдання кваліфікаційної роботи для обробки вхідних значень обрано метод лінійної регресії, що описаний у наступному підрозділі. Також варто зазначити, що так як за основу машинного навчання обрано метод лінійної регресії, потрібно дотримуватись регулярності своїх дій.

					<i>КПТР.2017019.01.01 ПЗ</i>	<i>Лист</i>
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		<i>33</i>

2.3 Аналіз вхідних даних системи машинного навчання.

Лінійна регресія

Регресія вивчає, як для усереднення даних найкраще апроксимувати криву. Тут використовується добре вивчені типи алгоритмів навчання. Застосовуючи регресію, ми намагаємося оцінити точки та криву даних, яка могла б їх створити. Крива що найкраще наближена дає нам модель, яка пояснює те, як міг бути отриманий цей набір даних.

Візьмемо для прикладу лінійне рівняння $y = 2x$, що описує залежність однієї змінної від іншої, тоді можна побачити певну пропорційність відношення цих змінних. В такому випадку коли лінія графіку залежності добре апроксимує точки отриманих даних, можна стверджувати - лінійна модель, що використовується є ефективною.

Не обов'язково обмежуватись однією лінією нахилу, можна використовувати багато різних ліній графіку для корегування загального коефіцієнту, замість того щоб використовувати один коефіцієнт нахилу. Рівняння, що містить параметр, називають моделлю. Висловлюючись у термінах машинного навчання, рівняння кривої найкращого наближення з'являється у результаті визначення параметрів моделі в процесі машинного навчання.

Інший приклад, рівняння $y = 5x$ також описує лінію, але з більш крутим нахилом. Можливо замінити цей коефіцієнт будь-яким дійсним числом, назвемо його ω , це рівняння знову буде описувати пряму лінію: $y = \omega x$. На рис. 2.7 зображено, як зміна параметра ω впливає на модель. Рівняння, що узагальнені таким чином можливо зобразити як $M = \{y = \omega x \mid \omega \in \mathbb{R}\}$. Тобто узагальнення вище слід розуміти як всі рівняння $y = \omega x$, у яких ω є дійсним числом [9].

M є множиною усіх можливих моделей. Формування значення ω генерує модель системи $M(\omega): y = \omega x$.

					<i>КПТР.2017019.01.01 ПЗ</i>	<i>Лист</i>
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		34

Алгоритм регресії створює функцію f , що перетворює вхідні дані у вихідні. Варто зазначити що вхідні дані можуть бути неперервними або дискретними, але вихідні дані повинні бути тальки неперервні(рис. 2.8).

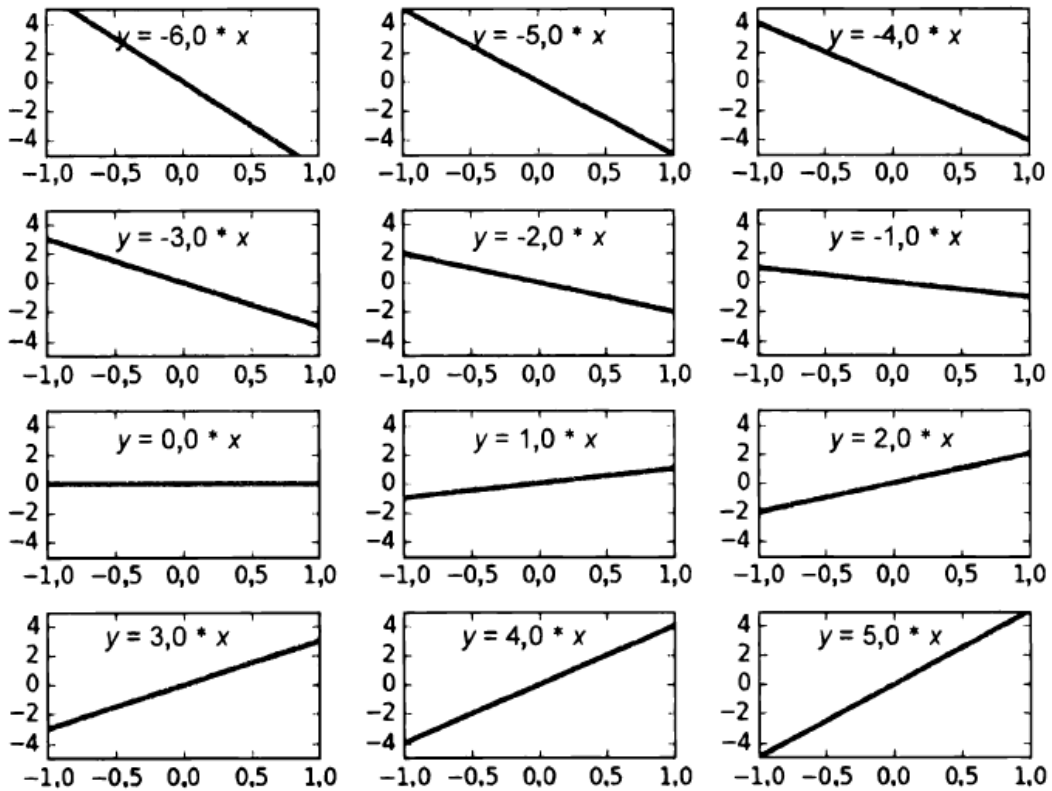


Рисунок 2.7 – Зміна параметра ω призводить до різних лінійних рівнянь. Множина всіх лінійних рівнянь створює лінійну модель M

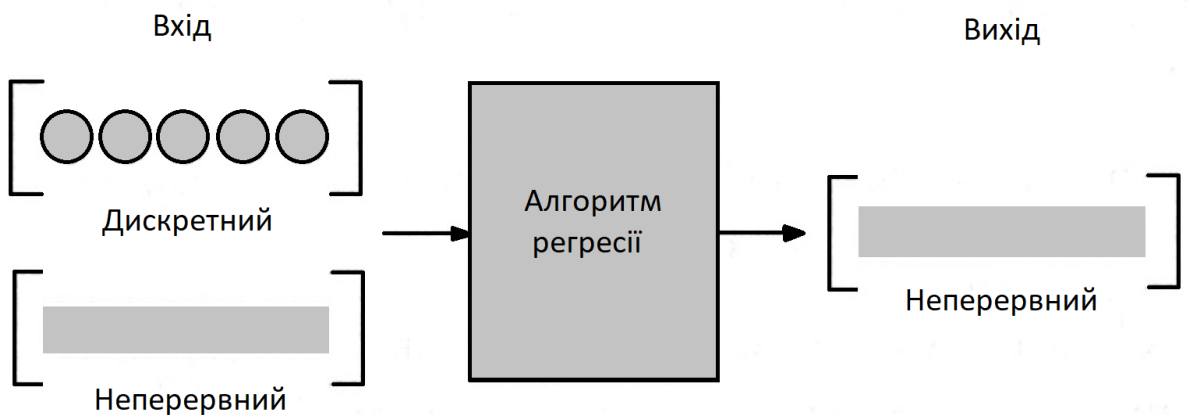


Рисунок 2.8 – Вхідні дані алгоритму регресії можуть бути неперервними або дискретними, але вихідні дані повинні бути тальки неперервні

Для розуміння лінійної регресії варто створити код імітування даних.

Для цього створимо файл Python, та надрукуємо код зображений нижче.

```
import numpy as np #Імпорт модуля NumPy
import matplotlib.pyplot as plt #Імпорт модуля matplotlib для
візуалізації даних
x_train = np.linspace(-1,1,100) #вхідні значення - 100 чисел
#для x від-1 до 1
y_train = 2 * x_train + np.random.randn(*x_train.shape) * 0.33
#вихідні дані з доданим шумом
plt.scatter(x_train, y_train)
plt.show()
# візуалізація даних з використанням бібліотеки
```

У ході виконання даного програмного коду відображається вікно(рис. 2.9).

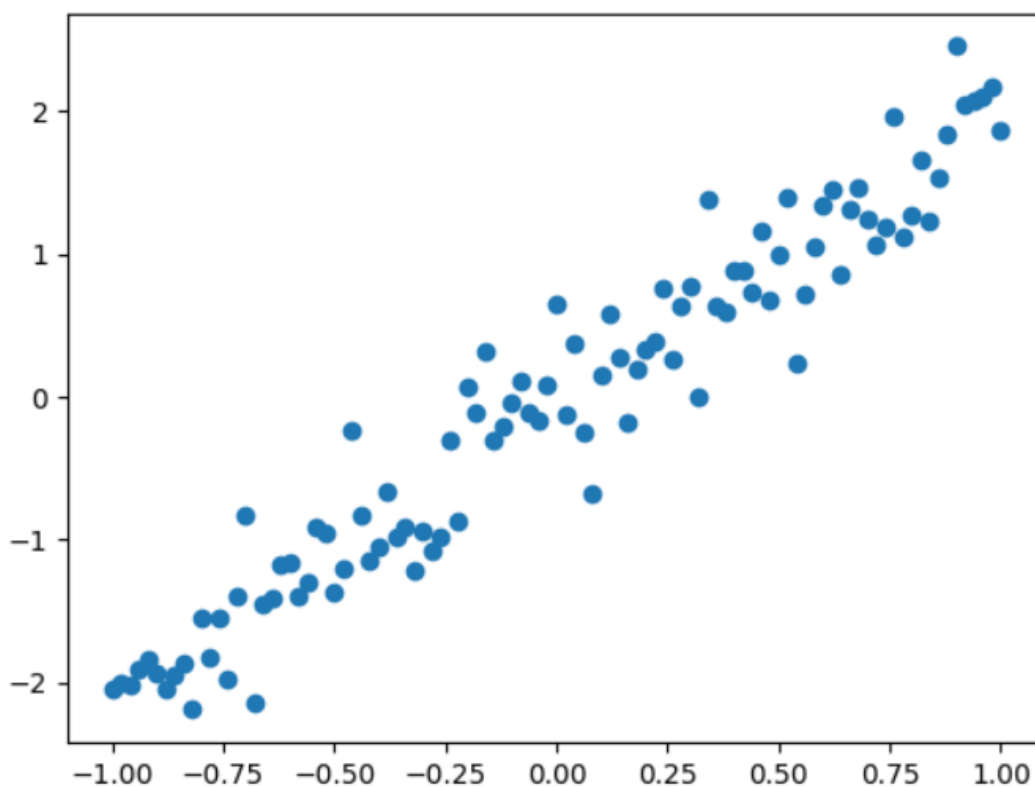


Рисунок 2.9 – Діаграма розсіювання для $y = 2x +$ (завади)

Тепер коли є можливість штучного створення бази даних, що зображено у програмному коді вище, потрібно апроксимувати їх лінією. Бібліотека TensorFlow багато разів оновлює та оцінює значення параметра для того знайти значення що найбільше підходить для всіх точок. Кожен крок

циклічного перебору даних для оновлення параметру, у даній бібліотеці має назву “epoch”.

Нижче зображено програмний код написаний на мові Python з використанням бібліотеки TensorFlow для використання елементів машинного навчання.

```
import tensorflow.compat.v1 as tf #Імпорт модуля tensorflow
tf.disable_v2_behavior()         #Вибір старої версії модуля
import numpy as np               #Імпорт модуля NumPy
import matplotlib.pyplot as plt  #Імпорт модуля matplotlib

learning_rate = 0.01            #константа алгоритму навчання
training_epochs = 100          #константа алгоритму навчання

x_train = np.linspace(-1,1,100)
y_train = 1 * x_train + np.random.randn(*x_train.shape) * 0.33
# нереальні дані, що потрібні для пошуку найбільш правильної
лінії

X = tf.placeholder(tf.float32)
Y = tf.placeholder(tf.float32)
# змінні необхідні для побудови лінії

def model(X, w):                #визначення моделі  $y=\omega X$ 
    return tf.multiply(X, w)

w = tf.Variable(0.0,name="weights")#визначення значення
перемінної

y_model = model(X, w)
c = tf.square(Y-y_model)
#визначення основної функції

train_op =
tf.train.GradientDescentOptimizer(learning_rate).minimize(c)
#операція, що викликається бібліотекою за алгоритмом навчання

sess = tf.Session()            #запуск функції бібліотеки
init = tf.global_variables_initializer()
sess.run(init)
#ініціалізація даних

for epoch in range(training_epochs):
    for (x, y) in zip(x_train, y_train):#цикл перебору бази
даних
        sess.run(train_op, feed_dict={X: x, Y: y})

w_val = sess.run(w)            # кінцеве значення коефіцієнту

sess.close()
#завершення функції бібліотеки
```

					<i>КПТР.2017019.01.01 ПЗ</i>	Лист
Зм.	Лист	№ докум.	Підп.	Дата		37

```
plt.scatter(x_train, y_train)
y_learned = x_train*w_val
print("w = ")
print(w_val)
plt.plot(x_train, y_learned, 'r')
plt.show()
#візуалізація кінцевого результату
```

Даний програмний код перебирає всі значення бази даних та в подальшому обраховує найбільш наближений коефіцієнт нахилу лінії. У випадку на рис. 2.10 програмний код видає значення коефіцієнту $w = 0.90601057$, це є досить близько до $w = 1$, що було встановлено у частині програмного коду яка відповідає за формування бази даних. На рис. 2.10 можна оцінити ефективність лінійної регресії.

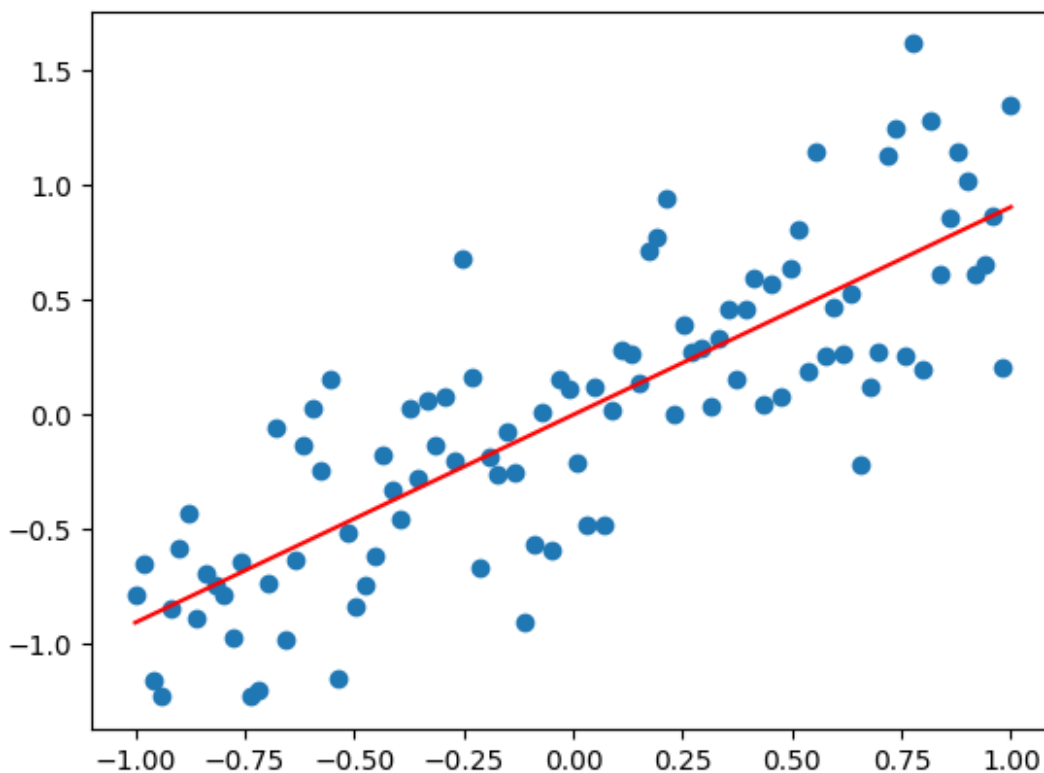


Рисунок 2.10 – Оцінка лінійної регресії

2.4 Застосування нелінійної регресії у системі машинного навчання

Зазвичай кореляція реального світу не носить лінійний характер. Наприклад, траєкторія ракети в космічному просторі викривляється по відношенню до спостерігача на землі. Інтенсивність сигналу Wi-Fi падає по закону зворотних квадратів. Зміна висоти квітки протягом його життя змінюється не лінійно.

Якщо точки даних формують згладжену криву, а не пряму лінію, модель регресії необхідно змінити з прямої лінії на інший метод. Один з таких методів використовує поліноміальну модель. Поліном є узагальненням лінійної функції. N-я ступінь полінома виглядає наступним чином (1):

$$f(x) = \omega_n x^n + \dots + \omega_1 x + \omega_0. \quad (1)$$

Якщо $n = 1$, поліном перетворюється в лінійну функцію. На рис 2.11 зображений приклад графіку з точками даних поліноміальної функції.

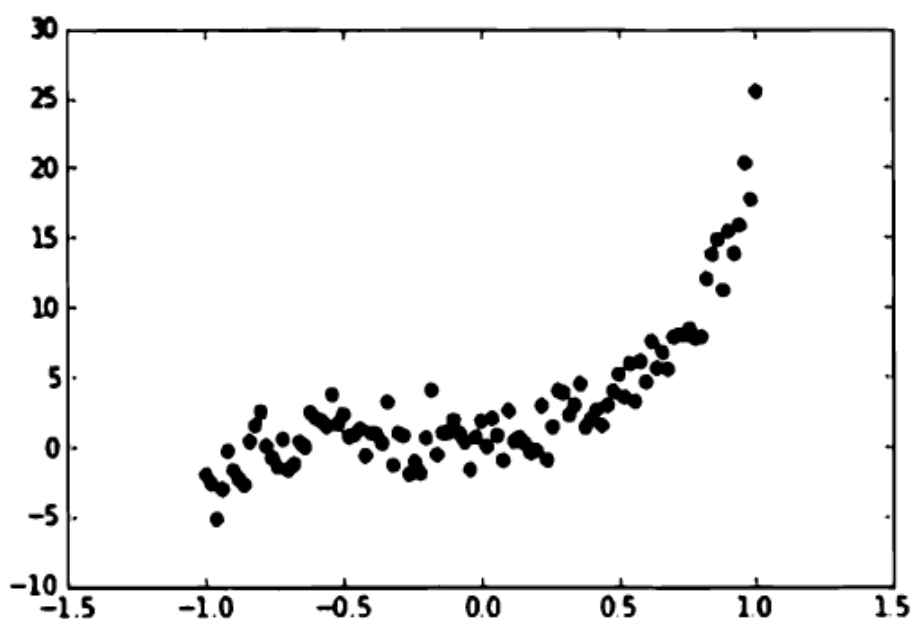


Рисунок 2.11 – Приклад графіку з точками даних поліноміальної функції

Бібліотека TensorFlow для використання елементів машинного навчання має функцію, що може апроксимувати дані та видавати поліном певної степені. На рис. 2.12 окрім точок даних зображена крива, що найкраще наближена до даних нелінійної залежності [9].

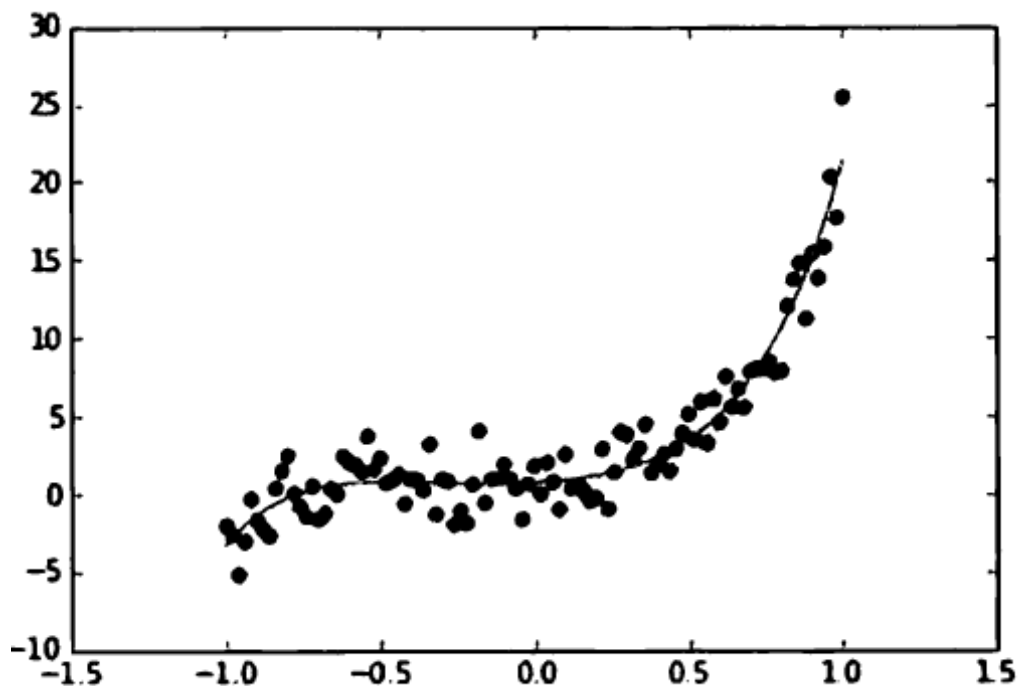


Рисунок 2.12 – Крива найкраще наближена до даних нелінійної залежності

2.5 Бібліотека TensorFlow і її використання у пропонованій системі

Бібліотека реалізована на основі мови C++ і має зручний інтерфейс прикладного програмування для Python, а також користується меншою популярністю інтерфейс для C++. Завдяки простим залежностям TensorFlow може бути швидко розгорнуто в різних архітектурах.

Аналогічно Theano (популярна обчислювальна бібліотека для мови Python) обчислення описуються як блок-схеми, відокремлюючи проектування від реалізації.

Створення прототипу в TensorFlow значно легше та швидше, ніж в Theano (програма ініціюється за секунди), оскільки багато операцій надходять попередньо скомпільованими. Оскільки TensorFlow використовується не тільки в нейронних мережах, вона також містить готовий пакет матричних обчислень і інструменти маніпулювання даними.

Більшість бібліотек, таких як Torch і Caffe, призначені виключно для глибоких нейронних мереж, але TensorFlow є більш універсальною, а також має можливість масштабування. Ця бібліотека добре задокументована і офіційно підтримується Google. Машинне навчання є складним предметом, тому тільки компанія з високою репутацією може підтримувати TensorFlow.

Було обрано саме дану бібліотеку, тому що її використання просте, що дозволяє зменшити час розробки. Також важливим є те, що бібліотека є великою та багатофункціональною [9].

					<i>КПТР.2017019.01.01 ПЗ</i>	<i>Лист</i>
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		41

Висновки до другого розділу

У другому розділі проведено огляд задач та сфер застосування машинного навчання у системах розумного будинку. Наведено класифікацію алгоритмів машинного навчання та описано їх можливості.

Було розроблено та описано структурну схему розумного будинку, що відповідає системі з можливостями реалізації машинного навчання за методом лінійної регресії.

Запропоновано спосіб реалізації алгоритму машинного навчання та наведено приклад реалізації коду лінійної регресії з використанням спеціалізованої бібліотеки. Окрім того наведено метод нелінійної регресії, для інших випадків та сценарії.

Описано особливість бібліотеки створеної для машинного навчання – TensorFlow, наведено її аналоги та описано плюси її використання.

					<i>КПТР.2017019.01.01 ПЗ</i>	<i>Лист</i>
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		42

РОЗДІЛ 3

ОПИС ВЛАСНОЇ РЕАЛІЗАЦІЇ СИСТЕМИ SMART HOUSE З ЕЛЕМЕНТАМИ САМОНАВЧАННЯ

3.1 Апаратна реалізація системи розумного будинку

Структура системи розумного будинку складається з великої кількості блоків, модулів, датчиків та керуючих пристроїв. Тому, нижче описано апаратну реалізацію кожного з елементів системи.

Контролер Smart House

Контролер є основним елементом усієї системи розумного будинку. Програмне забезпечення, що виконуватиметься даним пристроєм буде виконувати алгоритми машинного навчання.

Важливим фактором вибору для контролера, мікрокомп'ютер Raspberry Pi, є те що мікропроцесори можуть виконувати декілька операцій одночасно. В даному випадку Raspberry Pi повинен виконувати паралельно ряд таких функцій:

- Спостерігати за потоком даних протоколу MQTT;
- Виконувати функцію сервера;
- Реалізовувати алгоритм навчання системи, що включає у себе математичні обрахунки.

Зв'язок з всіма блоками керування здійснюватиметься за допомогою мережі Ethernet по протоколу MQTT.

MQTT протокол дозволяє здійснювати публікацію та отримання даних, отриманих по мережі Ethernet.

Мережа Ethernet забезпечує швидкий та легкий доступ до світу інтернету, що дає поштовх розвитку інтернету речей. Варто зазначити, що MQTT зв'язок може бути зашифрованим з використанням протоколів TLS / SSL, що забезпечуючи безпеку зв'язку.

					КПТР.2017019.01.01 ПЗ	Лист
Зм.	Лист	№ докум.	Підп.	Дата		43

Принцип дії MQTT полягає в тому, що пристрій програмно підписується на теми, які публікуються пристроєм, що виконує роль відправника. І навпаки, пристрій публікує тему, яку можна бачити тільки пристроєм який підписаний на дану тему.

Маршрутизатор

Маршрутизатор – пристрій, що об'єднує пристрої в одну мережу надаючи їм різні IP-адреси. У даній кваліфікаційній роботі у якості маршрутизатора виступає «Xiaomi Mi WiFi Router 4C Global» (рис. 3.1).



Рисунок 3.1 – Зображення Xiaomi Mi WiFi Router 4C Global

Завдяки наявності WIFI у даному маршрутизаторі є можливість безпроводного зв'язку контролера з іншими керуючими пристроями що дає можливість гнучкого розміщення.

Блоки керування

У кваліфікаційній роботі використовується 3 блоки керування, з яких один є основним. Саме тому на відміну від інших двох блоків у яких керуючий пристрій це плата розробника Arduino UNO, у даного блока використовується функціональна плата розробника - Arduino MEGA (рис. 3.2). Використання більш доскональної плати викликано великою кількістю інформації що обробляє даний блок.

Блок керування температурою котла

Даний блок є одним з основних елементів, що використовується у алгоритмі машинного навчання. Система розумного будинку вмикає опалення за годину до приходу власника квартири.

Керування температурою котла реалізоване за допомогою плати розробника Arduino NANO (рис. 3.4). Розміри плати є зручними для реалізації основної функції блоку. Зв'язок з блоком керування здійснюється по SPI-інтерфейсу.

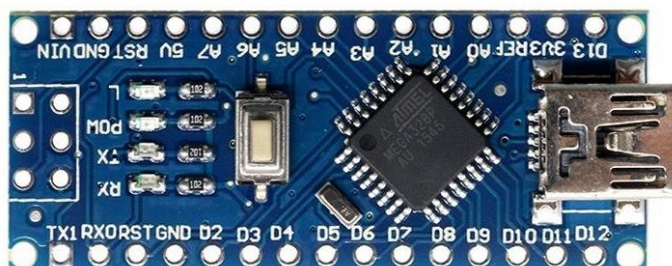


Рисунок 3.4 – Зображення плати Arduino NANO

Пристрій керування заслінкою вентиляції на кухні

Пристрій реалізований на базі плати Arduino NANO (рис. 3.4) та сервомотора SG90. Пристрій відкриває заслінку вентиляції при збільшенні концентрації диму та збільшенні вологості на кухні. Пристрій та блок керування здійснюють зв'язок по SPI-інтерфейсу.

Пристрій керування відкриття вікна на кухні

За допомогою плати Arduino NANO (рис. 3.4) пристрій отримує команду відкриття вікна. Відкриття здійснюється за тих самих умов, що й у пристрою керування заслінкою вентиляції. Для реалізації роботи пристрою використовується шаговий двигун 28BYJ-48 (рис. 3.5). Блок керування надсилає команди по SPI-інтерфейсу.

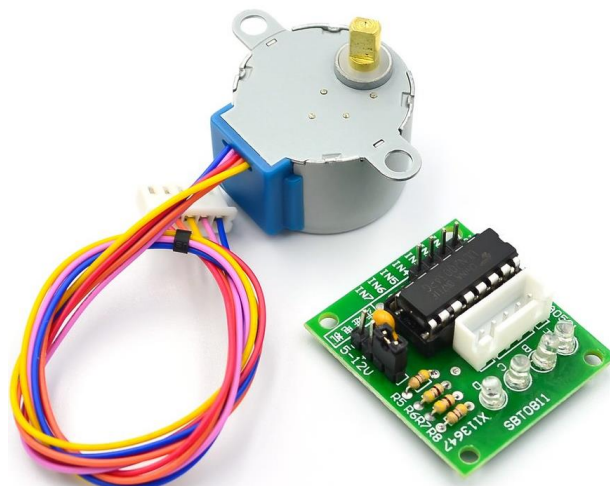


Рисунок 3.5 – Зображення шагового двигуна 28BYJ-48

Пристрій керування відкриття вікна у кімнаті

Плата Arduino NANO (рис. 3.4) по TWI-інтерфейсу отримує команду відкриття вікна. Для реалізації пристрою використовувався DC-DC перетворювач напруги LM2596S, так як для драйвера шагового двигуна A4988 (рис.3.6). На драйвер подається напруга перетворена з 5В на 9В, що необхідно для живлення обмоток двигуна NEMA17.

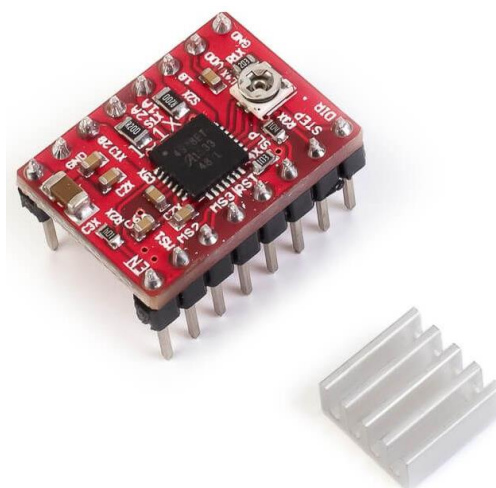


Рисунок 3.6 – Зображення драйвера шагового двигуна A4988

Датчик вологи

Датчик вологи HDS10 (рис. 3.7) призначений для відслідковування витоку води. Датчик встановлений у трьох місцях: ванна кімната, кухня та комора, де знаходяться клапани перекриття води. У разі аварійної ситуації система автоматично перекриває клапани холодної та гарячої води.



Рисунок 3.7 – Зображення датчику вологи HDS10

Датчик вологи та температури

Комбінований датчик вологи та температури DHT11 (рис. 3.8) допомагає регулювати температуру та вологу квартири. Система регулює температуру за допомогою відкриття вікна та керування котлом опалення. Керування вологою може виконуватись лише закриттям або відкриттям вікна. Даний датчик розміщений у спальній кімнаті та кухні.

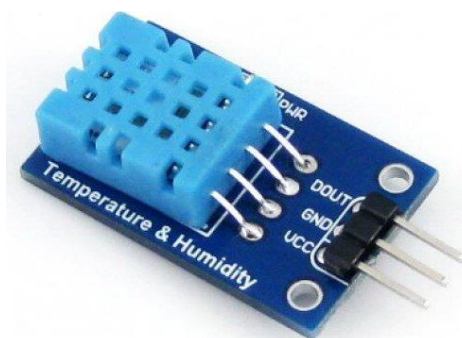


Рисунок 3.8 – Зображення датчику вологи та температури DHT11

Кінцевий вимикач

Для алгоритмів машинного навчання використовується лише кінцевий вимикач відкриття входних дверей KW12-3. Також у системі присутні два кінцевих вимикача відкриття вікна.

Датчик присутності (ІЧ)

Дозволяє виявляти рух людини на відстані до 7 метрів. Датчик присутності HC-SR501(рис. 3.9) є допоміжним для алгоритму машинного навчання.



Рисунок 3.9 – Зображення датчику присутності HC-SR501

Нижче на рис. 3.10 зображено прототип системи, що має підключення основних елементів які використовуються для машинного навчання. На фото прототипу є два блоки керування один з яких підключений до модуля реле та пристрою керування температурою опалення. До другого блоку підключені основні сенсори, що оповіщають контролер системи про повернення власника у помешкання.

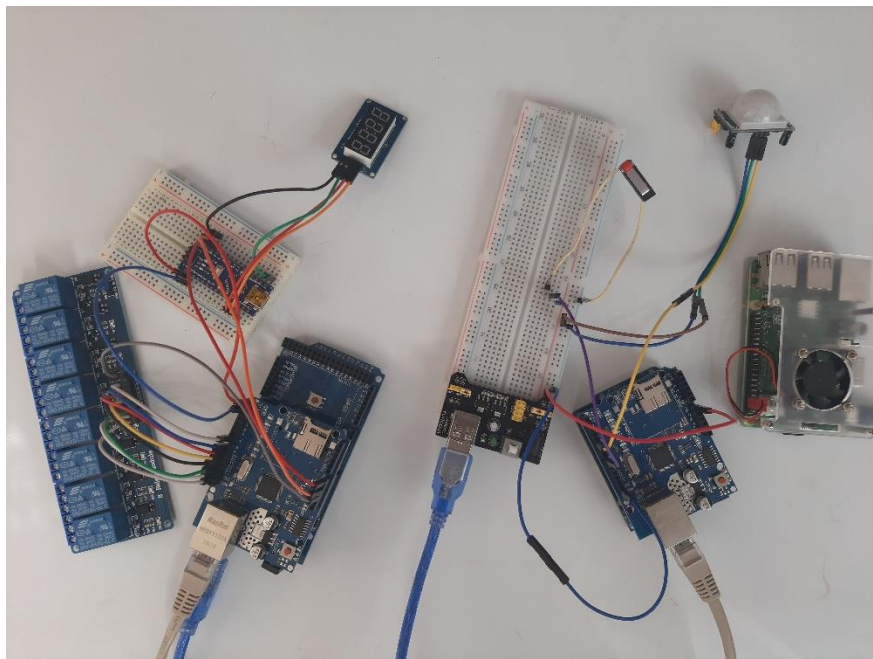


Рисунок 3.10 – Зображення прототипу системи з підключенням основних елементів.

3.2 Розробка принципової схеми

Принципова електрична схема системи SmartHouse розміщена у додатку Б та зображена на рис. 3.12.

Нижче будуть зображені таблиці відповідності назв контактів плат елементів до назв вказаних у принциповій схемі.

Живлення всіх елементів об'єднано у одну лінію «5VDC». Дане живлення забезпечує блок живлення на 100 Ватт(рис. 3.11), що живиться від мережі перемінної напруги 220В. При виборі блоку враховувались затрати енергії усіх елементів що споживають максимальний струм, таких як Raspberry Pi – 3А, блок керування відкриттям вікна у кімнаті – 4А та було вирішено взяти блок з потужністю у два рази більшою для зняття навантаження з нього.



Рисунок 3.11 – Зображення блоку живлення на 100 Ватт

Назви контактів блока живлення відповідають назвам на принциповій схемі.

Позначення елементів на принциповій схемі мають відповідність з найменуваннями на листі переліку елементів

Ethernet роз'єми

Позначення маршрутизатору на принциповій схемі - M1. Ethernet є стандартним з'єднанням яке має стандартну порядок контактів зображену у табл. 3.1.

Таблиця 3.1 –Стандартне з'єднання роз'ємів Ethernet

Контакт	Коло	Опис контактів роз'єму
1	TX+	Transmit data+
2	TX-	Transmit data-
3	RX+	Receive data+
4	N/A	-
5	N/A	-
6	RX-	Receive data-
7	N/A	-
8	N/A	-

У даній системі існує 8 таких роз'ємів. 4 з них знаходяться у маршрутизаторі (M1), і по одному на блоках керування 1-3(A1, A4, A6), та основному контролері(A5).

Блок керування 1

Позначення першого блоку керування на принциповій схемі – A1. Плата Arduino UNO, що входить до складу блоку має порядок контактів, що зображена на рис. 3.13.

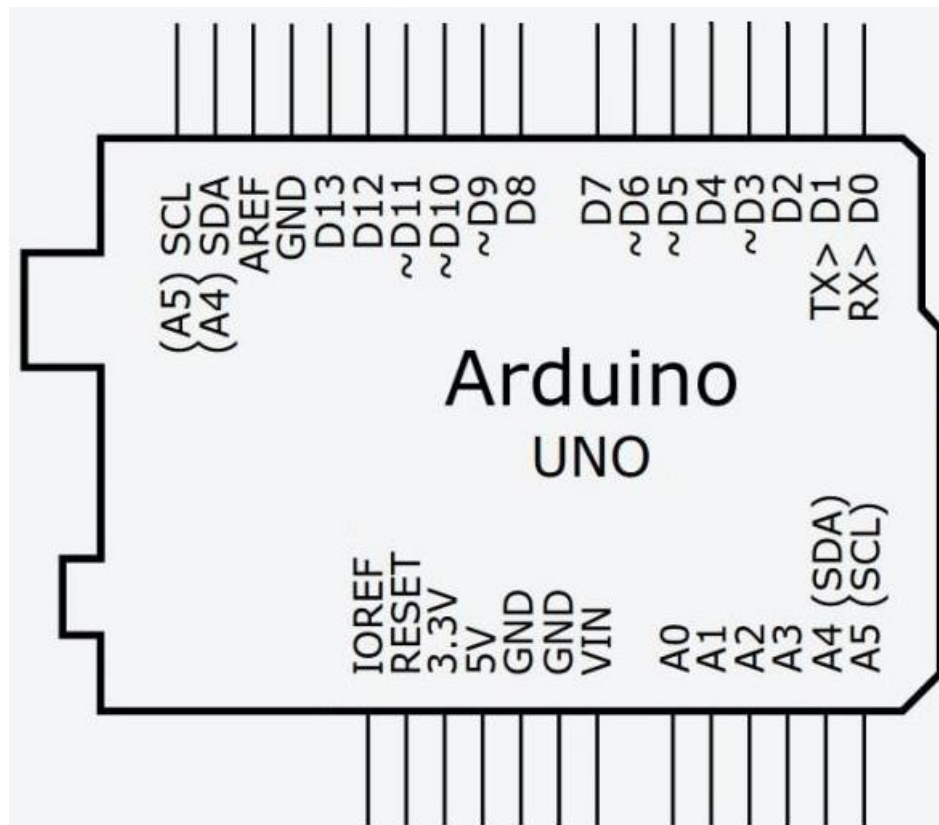


Рисунок 3.13 – Зображення порядку контактів плати розробника Arduino UNO

Датчики, кінцевий вимикач та пристрої керування, що обробляються у першому блоці описані нижче:

- Датчик вологи HDS10;
- Кінцевий вимикач KW12-3;
- Датчик вологи та температури DHT11;
- Датчик витоку газу MQ-5;
- Датчик диму MQ-5;
- Датчик вогню KY-026;
- Датчик присутності (ІЧ) HC – SR501;
- Пристрій керування заслінкою вентиляції на кухні;
- Пристрій керування відкриття вікна на кухні.

У табл. 3.2 наведено відповідність порядку контактів плати мікроконтролера Atmega 328P до порядку контактів, що вказана у принциповій схемі системи Smart House.

Таблиця 3.2 – Таблиця відповідності елемента А1

Контакт	Коло на принциповій схемі	Відповідність порядку контактів на Arduino UNO
1	5VDC	5V
2	GND	GND
3	X3IN1	D13
4	X3IN2	D12
5	X3IN3	D11
6	X3IN4	D10
7	X3IN5	D9
8	SWITCH1SIG	D8
9	X3SIG1	D7
10	X3SIG2	D6
11	X3SIG3	D5
12	X3SIG4	D4
13	X3SIG5	D3
14	X3SIG6	D2
15	X3SIG7	D1

Блок керування 2

Позначення другого блоку керування на принциповій схемі – А4. Плата Arduino MEGA, що входить до складу блоку має порядок контактів, що зображений на рис. 3.14.

Таблиця 3.3 – Таблиця відповідності елемента А4

Контакт	Коло на принциповій схемі	Відповідність порядку контактів на Arduino UNO
1	5VDC	5V
2	GND	GND
3	X14IN1	D14
4	X14IN2	D13
5	X14IN3	D12
6	X14IN4	D11
7	X14IN5	D10
8	X14IN6	D9
9	X14IN7	D8
10	X14IN8	D7
11	X14IN9	D6
12	X14IN10	D5
13	X14IN11	D4
14	X14IN12	D3
15	X14IN13	D2
16	TEMP TX	D1 (TX)
17	TEMP RX	D0 (RX)
18	X14SIG1	D15
19	X14SIG2	D16
20	X14SIG3	A0
21	X14SIG4	A1
22	SWITCH2SIG	D17

Блок керування 3

Позначення третього блоку керування на принциповій схемі – А6. Плата Arduino UNO, що входить до складу блоку має порядок контактів, що зображена на рис. 3.13.

Елементи, що обробляються у третьому блоці керування описані нижче:

- Кінцевий вимикач KW12-3;
- Датчик вологи та температури DHT11;
- Датчик присутності (ІЧ) HC – SR501;
- Пристрій керування відкриття вікна у спальній кімнаті.

У табл.3.4 наведено відповідність порядку контактів Arduino UNO до порядку контактів, що наведена у принциповій схемі.

Таблиця 3.4 – Таблиця відповідності елемента А6

Контакт	Коло на принциповій схемі	Відповідність порядку контактів на Arduino UNO
1	5VDC	5V
2	GND	GND
3	STEP	D6
4	DIR	D5
5	SWITCH3SIG	D4
6	X27SIG1	D2
7	X27SIG2	D1
8	X27SIG3	D3

3.3 Програмна реалізація машинного навчання, опис алгоритму та програмного забезпечення

Програма реалізації машинного навчання написана на мові програмування Python. За основу коду взято бібліотеку “tensorflow”. Дана бібліотека створена для реалізації існуючих методів машинного навчання, метод лінійної регресії включно.

Перед написанням основного програмного забезпечення на мові Python, необхідно створити систему розумного будинку з можливістю надсилання всіх важливих “тем” MQTT протоколу. Для цього використовуються елементи програмування на мові Java, що присутні у автоматичній побудові веб-дизайну розумного будинку в середовищі OpenHAB. Також для створення елементів системи за допомогою яких здійснюється взаємодія з навколишнім світом, відбувається написання коду на мові C/C++.

Створення веб-інтерфейсу системи Smart House

Для створення інтерфейсу здійснюється редагування файлів розширення `***.items`, `***.sitemap`. Дані розширення створені розробниками середовища Open HAB, та у написанні програмного коду у цих розширеннях використовуються елементи програмування мови Java.

Написання програмного коду з розширенням `***.items` використовується для створення тем MQTT протоколу для реалізації передачі даних з певною адресацією. Нижче зображені та описані частинки коду даного розширення:

```
Switch Relay1 "Світло на кухні"
{channel="mqtt:topic:55922470:Relay1"}
// Створення елемента протоколу "Switch", що відповідає за
перемикання з 0 на 1
.....
Number temp1 "Температура на кухні [%.1f °C]" <temperature>
{channel="mqtt:topic:55922470:temp1"}
// Створення елемента протоколу "number", що відповідає за
індикацію значення та розмірності даних.
.....
Dimmer TempK "Температура котла"
{channel="mqtt:topic:test:TempK"}
```

```
// Створення елемента протоколу "dimmer", для регулювання даних
вручну
.....
DateTime Date "Date [%1$td.%1$tm.%1$tY %1$tH:%1$tM]" {
channel="ntp:ntp:demo:dateTime" }
// Створення елемента "date", що відповідає за індикацію та
надсилає значення часу у момент запиту
```

Написання програмного коду з розширенням *****.sitemaps** використовується для автоматичного створення базового інтерфейсу. Нижче зображені та описані частинки коду:

```
sitemap demo label="HOME" // Створення вікна дизайну під назвою
"HOME"
{
    Frame label="HOME" // Звернення до дизайну
    {
        Switch item=Relay1 label="Світло на кухні"
// Автоматичне створення перемикача з назвою "Світло на кухні"
.....
        Text item=temp1
// Автоматичне створення комірки для виведення значень
.....
        Slider item=TempK label="Температура котла"
// Автоматичне створення регулюємої області з назвою
"Температура котла"
.....
    }
    Frame label="Температура у кухні" //Створення
вікна
    {
        Chart label="Last Hour Temp1" item=temp1
period=h refresh=30000
// Підключення та налаштування параметрів виведеного графіка
.....
    }
}
```

На рис.3.15 зображено базовий інтерфейс середовища OpenHAB, що створюється за допомогою написання описаного вище програмного коду.

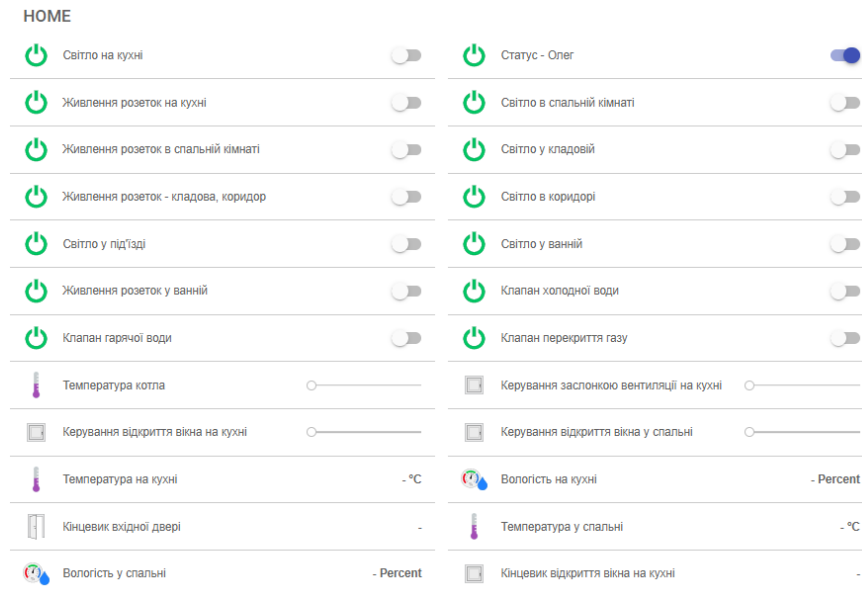


Рисунок 3.15 – Зображення вигляду базового інтерфейсу середовища OpenHAB

Опис використання MQTT протоколу у написанні програмного забезпечення для блоків керування 1-3

Компіляція програмного коду на мові C/C++ відбувається у середовищі ArduinoIDE. Для реалізації використання MQTT протоколу у мережі Ethernet у програмних забезпеченнях використовуються бібліотеки “Ethernet.h” та “PubSubClient.h”.

Нижче описані частини коду, що є спільними для всіх блоків керування.

```
#include <Ethernet.h> // підключення бібліотеки
#include <PubSubClient.h> // підключення бібліотеки
.....
// Встановлення індивідуальної Mac-адреси кожного блоку
byte mac[] = { 0x90, 0xB2, 0xFA, 0x0D, 0x4E, 0x49 };
// Встановлення індивідуальної IP-адреси кожного блоку
IPAddress ip(192, 168, 31, 210);
// Встановлення IP адреси MQTT брокера, тобто сервера системи
byte server[] = { 192, 168, 31, 217 };

const char* mqtt_username = "##"; // введення логіну
const char* mqtt_password = "##"; // введення паролю

// Функція зв'язку
void callback(char* topic, byte* payload, unsigned int length) {
    payload[length] = '\0'; // перевірка нових повідомлень у
    брокера
    Serial.print("Topic: ");
    Serial.print(String(topic));
    Serial.println(" - ");
}
```

```

.....
// Запис значення повідомлення у змінну
if (String(topic) == "/HOME/relay1") {
    String value = String((char*)payload);
    Relay1 = value.substring(0, value.indexOf(';')).toInt();
    Serial.print("Relay1: ");
    Serial.println(Relay1);
    digitalWrite(Relay1_pin, Relay1);
}
.....
Serial.println("Callback");
Serial.println();
}

EthernetClient ethClient; // створення ethernet-клієнта
// створення клієнта для протоколу MQTT
PubSubClient client(server, 1883, callback, ethClient);

// функція підписки на теми та підключення до брокера
boolean reconnect() {
    //Serial.println("reconnect...");
    if (client.connect("Arduino", mqtt_username, mqtt_password)) {
client.subscribe("/HOME/relay1"); Serial.println("Connected to:
/HOME/relay1"); // оформлення підписки на тему
        Serial.println("MQTT connected");
    }
    return client.connected();
}

.....
void setup()
{
    Serial.begin(9600);
    .....
    Ethernet.begin(mac, ip); // підключення до мережі
    Serial.print("My ip address: ");
    Serial.println(Ethernet.localIP());
    reconnect(); // Виклик функції вище
        Serial.print("MQTT connected");
        client.publish("/HOME/temp1", "0.00"); // публікація теми
MQTT
        client.publish("/HOME/humid1", "0.00"); // публікація теми
MQTT
}

```

Лістинг коду для блоків керування знаходиться у додатку Г.

Опис та зображення алгоритму роботи машинного навчання

Написання програмного забезпечення для машинного навчання відбувається на мові програмування Python. Алгоритм роботи навчання зображено на рис. 3.16.

					<i>КПТР.2017019.01.01 ПЗ</i>	Лист
Зм.	Лист	№ докум.	Підп.	Дата		61

Програмне забезпечення, що створене за алгоритмом на початку підключає всі необхідні бібліотеки та створює змінні до яких входить масив, що відповідає за години повернення власника квартири додому. Далі програма заходить в цикл для повторюваності дій. Після входу в цикл, програма очікує повідомлення від теми на яку вона підписана. При виконанні умови отримання повідомлення, починається отримання даних, після цього відбувається перевірка умов, що поставлені після отримання даних. До цих умов відноситься програмна перевірка часу та дня тижня. При не виконанні умов програма звертається до очікування отримання нового повідомлення. Якщо умови вище виконуються то програма записує дані до масиву, що відповідає за значення аргументу та плюсує день до змінної, що відповідає за кількість пройдених днів навчання. Потім йде умова, що перевіряє чи пройшов період навчання. При виконанні умови починаються обрахунки коефіцієнту нахилу за методом лінійної регресії та встановлення часу увімкнення опалення.

					<i>КПТР.2017019.01.01 ПЗ</i>	<i>Лист</i>
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		62

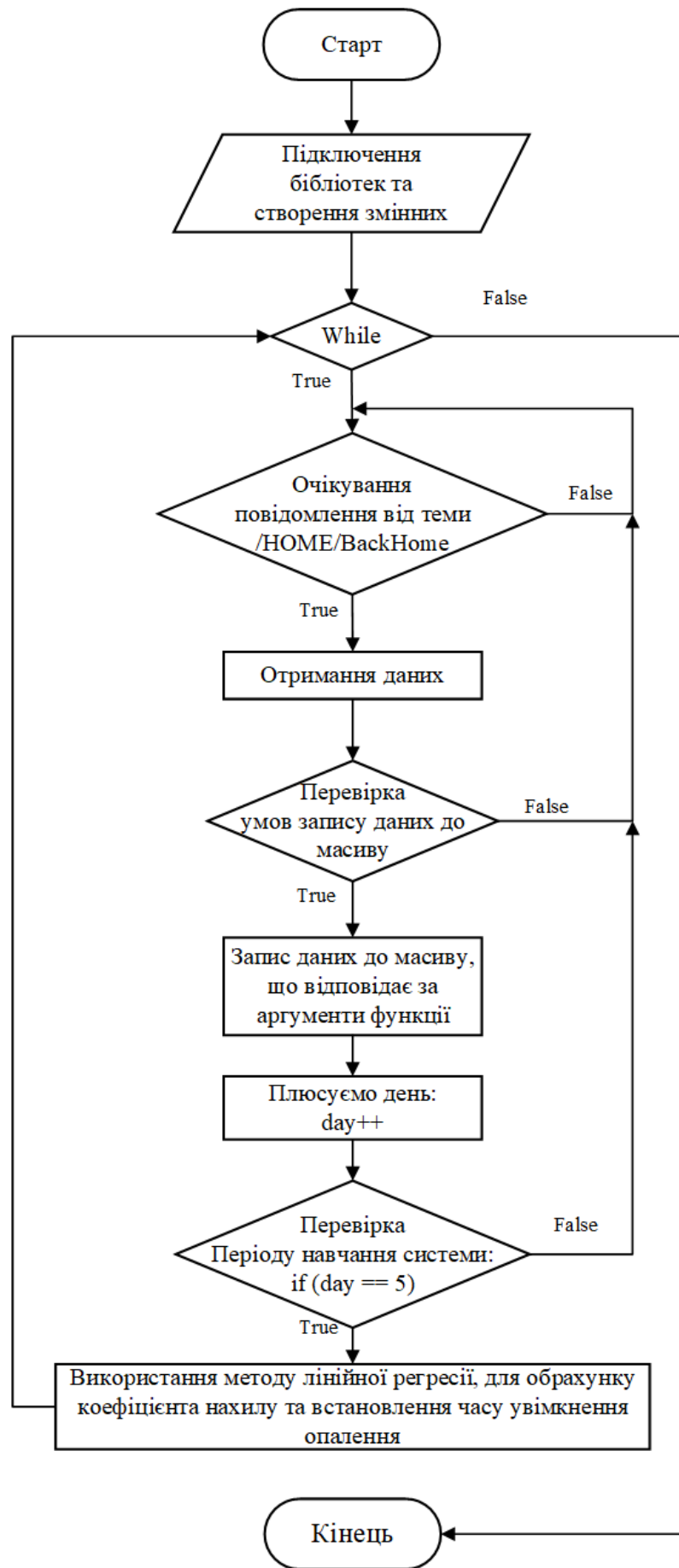


Рисунок 3.16 – Зображення вигляду базового інтерфейсу середовища OpenNAV

Зм.	Лист	№ докум.	Підп.	Дата

Програмне забезпечення машинного навчання

Для написання програми використовуються такі основні бібліотеки:

- Tensorflow;

Бібліотека створена для реалізації елементів машинного навчання на мові програмування Python. Містить у собі багато класів, що відповідають за методи машинного навчання. При вмінні користуватись даною бібліотекою реалізація алгоритмів машинного навчання суттєво спрощується.

- paho.mqtt.client;

Бібліотека створена для використання MQTT протоколу на мові програмування Python.

- datetime.

Бібліотека дає можливість запису реального часу у змінні на мові програмування Python для подальших операцій.

Нижче будуть описані основні функції бібліотеки Tensorflow, що використовуються для реалізації машинного навчання.

```
X = tf.placeholder(tf.float32)
Y = tf.placeholder(tf.float32)
def model(X, w):
    return tf.multiply(X, w)
w = tf.Variable(0.0, name="weights")

y_model = model(X, w)
cost = tf.square(Y-y_model)

train_op =
tf.train.GradientDescentOptimizer(learning_rate).minimize(cost)

sess = tf.Session()
init = tf.global_variables_initializer()
sess.run(init)

for epoch in range(training_epochs):
    for (x, y) in zip(x_train, y_train):
        sess.run(train_op, feed_dict={X: x, Y: y})

w_val = sess.run(w)

sess.close()
plt.scatter(x_train, y_train)
y_learned = [0, 0, 0, 0, 0, 0 ]
y_learned[0] = x_train[0]*w_val
```

					<i>КПТР.2017019.01.01 ПЗ</i>	<i>Лист</i>
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		64

```

y_learned[1] = x_train[1]*w_val
y_learned[2] = x_train[2]*w_val
y_learned[3] = x_train[3]*w_val
y_learned[4] = x_train[4]*w_val
y_learned[5] = x_train[5]*w_val
print("w = ")
print(w_val)
print(y_learned)
plt.plot(x_train, y_learned, 'r')
plt.show()
publish(/HOME/timew, payload=w_val, qos=0, retain=False)

```

На рис. 3.17 зображено користувацький інтерфейс системи Smart House. Створений у програмному середовищі OpenHAB.

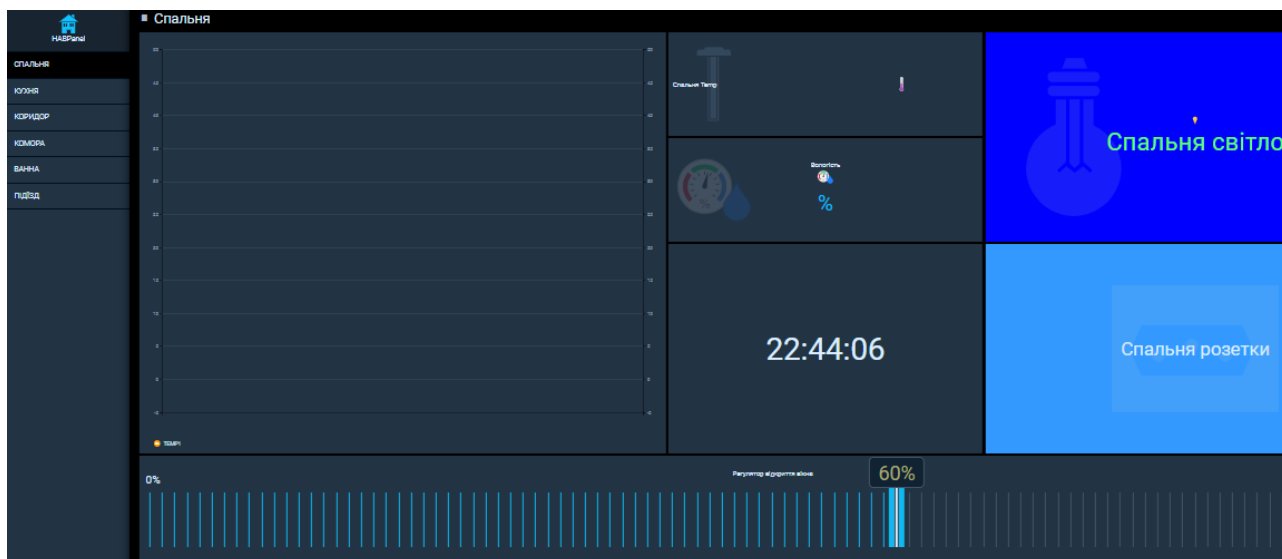


Рисунок 3.17 – Зображення вигляду користувацького інтерфейсу середовища OpenHAB

Висновки до третього розділу

У третьому розділі описано апаратну реалізацію системи Smart House. Описано складові та їх функції в системі.

Описано підключення основних елементів системи. Розроблено прототип на базі описаних компонентів апаратної реалізації. Створено базовий та користувацький інтерфейс системи у програмному забезпеченні OpenHAB. Написано програми відповідних розширень, що у подальшому дають можливість взаємодії сервера з зовнішніми пристроями системи.

Реалізовано та описано алгоритм програмного забезпечення, що відповідає за реалізацію методу лінійної регресії у кваліфікаційній роботі. Написано програмний код, що дає можливість передбачення поведінки мешканця будинку в умовах, що дають можливість використовувати метод лінійної регресії.

					<i>КПТР.2017019.01.01 ПЗ</i>	<i>Лист</i>
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		<i>66</i>

ВИСНОВКИ

Метою кваліфікаційного проекту була розробка апаратного та програмного забезпечення для системи Smart House на основі аналізу поведінки мешканців.

Було створено програмне забезпечення для машинного навчання системи Smart House з використанням методу лінійної регресії. Код програми писався на мові програмування Python з використанням бібліотеки створеної для машинного навчання – TensorFlow.

Апаратна реалізація здійснювалась за допомогою плат розробників та модулів Arduino, також мікрокомп'ютера Raspberry Pi. Мікрокомп'ютер виступав основним контролером реалізації алгоритмів машинного навчання та одночасно сервером системи.

Фіксація повернення до помешкання людини відбувається у робочі дні. Тобто завершення першого періоду машинного навчання буде через 5 робочих днів після запуску системи.

У момент максимального навантаження система споживає 50 Ватт/год. Та насправді в середньому система споживає 20 Ватт/год.

Виконання кваліфікаційної роботи, на прикладі наближення вихідних даних до бажаного користувачем часу на 20хв., при наявності різниці в масиві аргументу у 40 хв., показало, що використання методу лінійної регресії є ефективним у передбаченні поведінки людини.

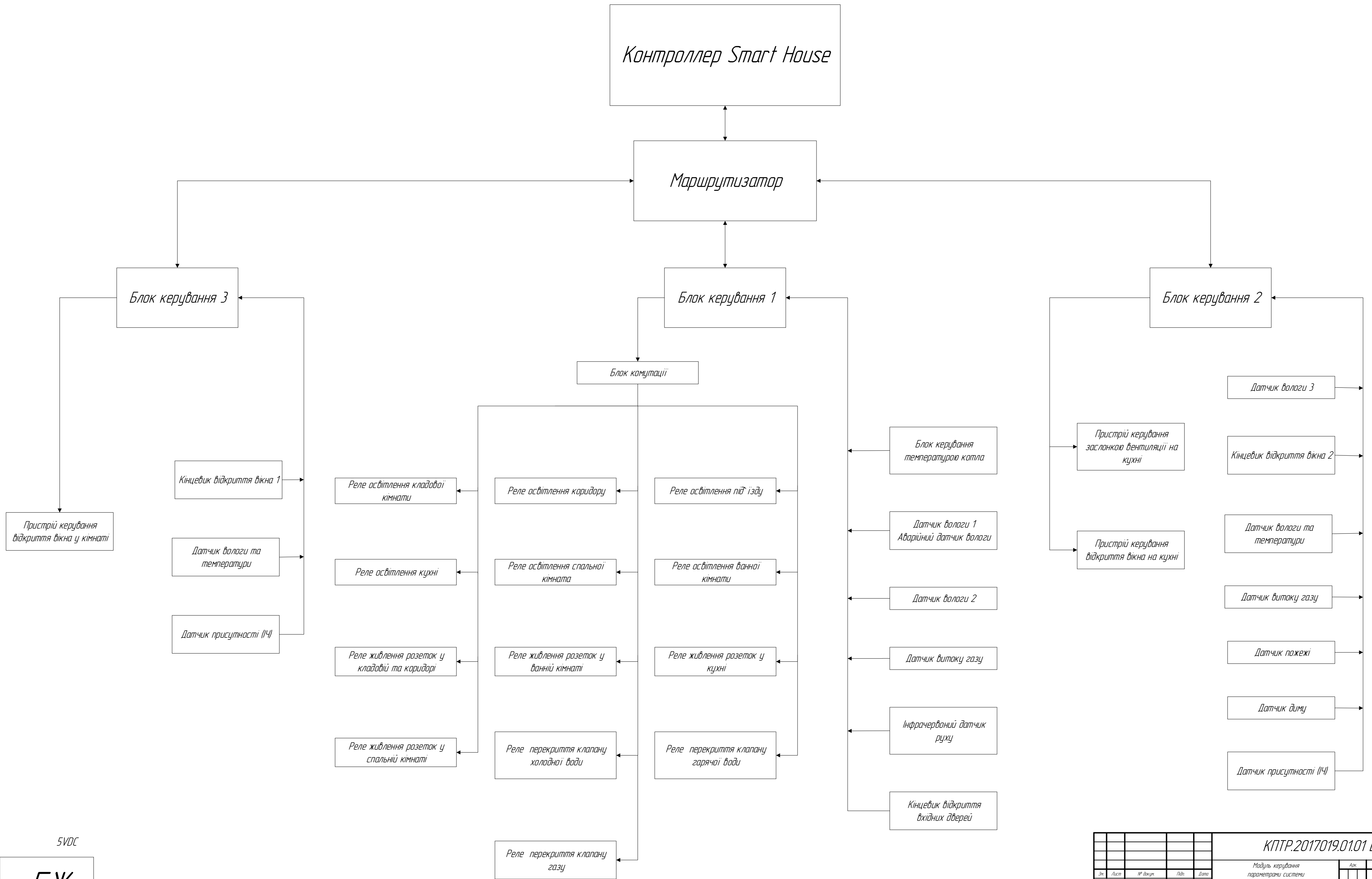
					<i>КПТР.2017019.01.01 ПЗ</i>	<i>Лист</i>
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		67

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. https://www.researchgate.net/publication/327136969_Smart_Home_Technology_An_Exploration_of_End_User_Perceptions
2. <https://vencon.ua/ua/articles/rejting-sistem-umnyy-dom-po-proizvoditelyam>
3. <https://ubidots.com/blog/open-source-home-automation/>
4. <https://ubidots.com/blog/open-source-home-automation/>
5. https://uk.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%B5_%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F
6. <https://www.expert.ai/blog/machine-learning-definition/>
7. <https://www.instructables.com/How-to-Use-MQTT-With-the-Raspberry-Pi-and-ESP8266/> - роз'яснення MQTT on Python
8. <https://webformyself.com/data-i-vremya-v-python/>
9. Машинное обучение и TensorFlow. - СПб.: Питер, 2019. - 336 с.: ил. - (Серия «Библиотека программиста»).

					<i>КПТР.2017019.01.01 ПЗ</i>	<i>Лист</i>
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		<i>68</i>

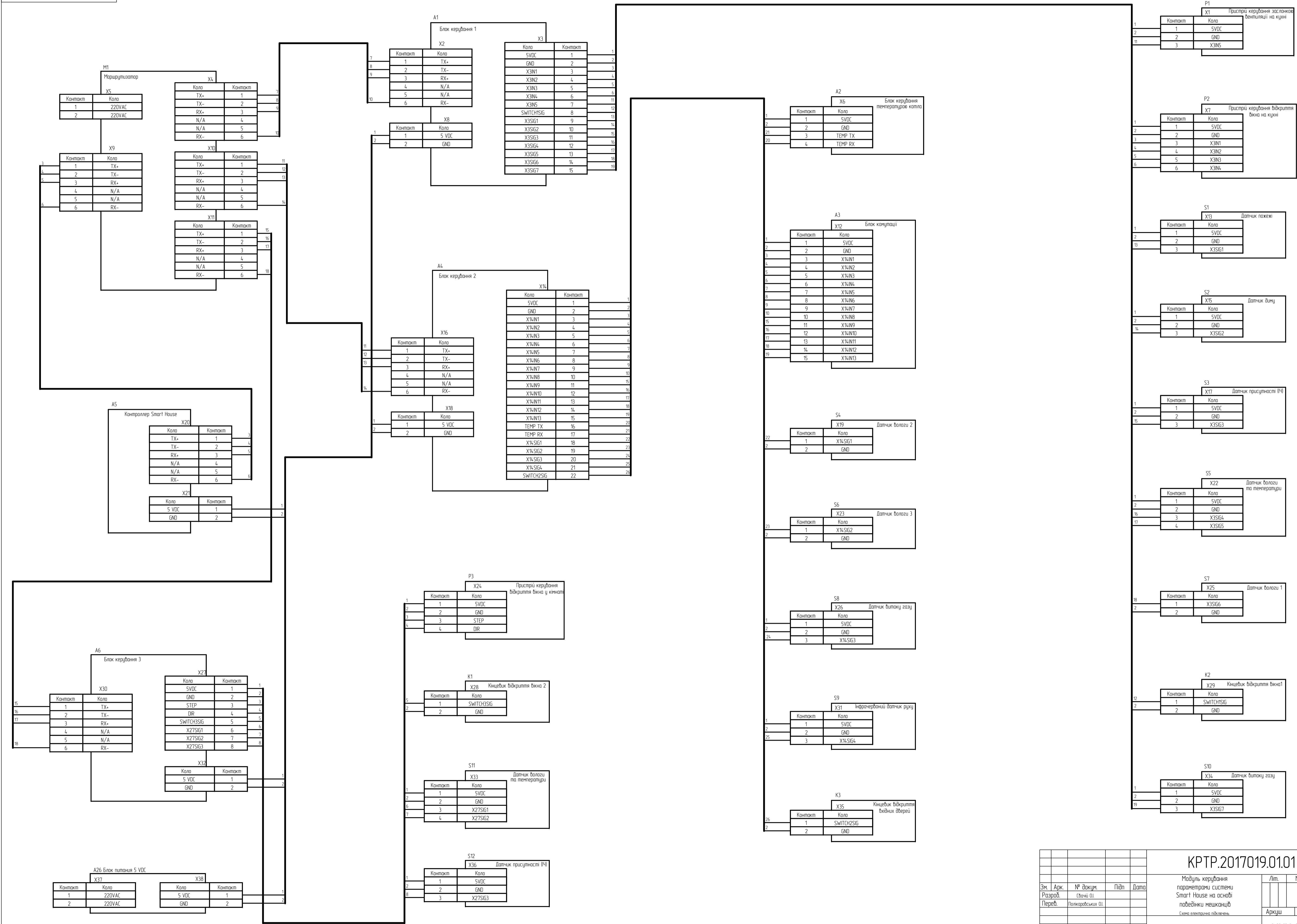
Додаток А
Структурна схема системи



5VDC
БЖ

					КІТР.2017019.01.01 Е1		
					Модуль керування параметрами системи Smart House на основі побувднки мешканцв		
					Структурна схема		
Арк.	Розроб.	Лист	№ докум.	Підп.	Дата	Розроб.	Масштаб
	Лавров	1	Полікарпович О.І.				
Масштаб	Арк.	Лист					
		1	ХНУ ТР-17-1				

Додаток Б
Схема електрична підключень



KPTR.2017019.01.01 E5					Модуль керування параметрами системи Smart House на основі побутових механізмів			Лист	Маса	Масштаб
Зм.	Арх.	№ док.	Підп.	Дата	Схема електричного підключення			Аркши	Аркши	1
Разроб.	Стечи ОІ									
Переб.	Полікарповських ОІ									
Нконтр.										
Замб.								ХНУ ТР-17-1		

Додаток В
Перелік елементів

Позначення	Найменування	Кільк.	Примітка
	<u>Маршрутизатори</u>		
M1	Xiaomi Mi WiFi Router 4C Global	1	
	<u>Блоки керування</u>		
A1, A6	Блок керування на базі ArduinoUNO та Ethernet Shield	2	
A2	Блок керування на базі ArduinoNANO	1	
A3	Блок комутації на базі реле SRD-05VDC-SL-C	1	
A4	Блок керування на базі ArduinoMEGA та Ethernet Shield	1	
A5	Контролер Smart House на базі мікрокомп'ютера RaspberryPI	1	
	<u>Датчики</u>		
S1	Модуль датчик вогню KY-026	1	
S2	Модуль датчику диму MQ-2	1	
S3, S12	Модуль датчика присутності (IIC) HC-SR501	2	
S4, S6, S7	Датчик вологи HDS10	3	
S5, S11	Модуль датчика температури та вологи DHT11	2	
S8, S10	Модуль датчику витoku газу MQ-5	2	
S9	Інфрачервоний датчик руху YL-63	1	
	<u>Кінцевики</u>		
K1, K2, K3	Кінцевик KW12-3	3	
	<u>Виконавчі пристрої</u>		
P1	Плата розробника ArduinoNANO, сервомотор SG90	1	
P2	Плата розробника ArduinoNANO, шаговий мотор 28BYJ-48	1	
P3	Плата розробника ArduinoNANO, драйвер A4988	1	
	<u>Блоки живлення</u>		
U1	Блок живлення 5VDC	1	

КПТР.2017019.01.01 ПЕЗ

Зм.	Арк.	№ докум.	Підп.	Дата
Розроб.		Свачій ОІ.		10.06.21
Перед.		Полікародських ОІ.		10.06.21
Н. контр.				10.06
Затв.		Пігученко		10.06

Модуль керування параметрами системи Smart House на основі поведінки мешканців
Перелік елементів

Літ.	Аркуш	Аркушів
		1

Додаток Г

ЛІСТИНГ КОДУ НА МОВІ Python

```
import tensorflow.compat.v1 as tf
tf.disable_v2_behavior()
import numpy as np
import matplotlib.pyplot as plt
import paho.mqtt.client as mqtt

from datetime import date
from datetime import time
from datetime import datetime

mqtt_username = "#####"
mqtt_password = "#####"
mqtt_topic = "/HOME/BackHome"
mqtt_broker_ip = "192.168.##.###"

client = mqtt.Client()

client.username_pw_set(mqtt_username, mqtt_password)

def on_connect(client, userdata, flags, rc):

    print ("Connected!", str(rc))
    client.subscribe(mqtt_topic)

learning_rate = 0.01
training_epochs = 6

i=1
w_val = 0
x_train = np.linspace(0,5,6)
```

```

x_nextweek = [0, 0, 0, 0, 0]
x_train = [0, 0, 0, 0, 0, 0 ]
y_train = [0, 1, 2, 3, 4, 5]

while 1:
    today = date.today()
    now = datetime.now()
    print ("Today weekday:", today.weekday())
    todayfm = today.weekday()
    hour = (now.strftime("%H"))
    print (todayfm, hour)

    def on_message(client, userdata, msg):

        print ("Topic: ", msg.topic + "\nMessage: " + str(msg.payload))

        if msg.payload == 1:
            if (int(hour) > 16):
                i=1

        client.on_connect = on_connect
        client.on_message = on_message

        client.connect(mqtt_broker_ip, 1883)

        client.loop_forever()
        client.disconnect()

        if i == 1:
            if todayfm == 0:
                x_train[1] = hour / 24
                i=0

            if todayfm == 1:
                h = int(hour)

```

```

        x_train[2] = 1 + h / 24
        i=0
    if todayfm == 2:
        h = int(hour)
        x_train[3] = 2 + h / 24
        i=0
    if todayfm == 3:
        h = int(hour)
        x_train[4] = 3 + h / 24
        i=0
    if todayfm == 4:
        h = int(hour)
        x_train[5] = 4 + h / 24
        i=0

if todayfm == 5:
    X = tf.placeholder(tf.float32)
    Y = tf.placeholder(tf.float32)
    def model(X, w):
        return tf.multiply(X, w)

    w = tf.Variable(0.0, name="weights")

    y_model = model(X, w)
    cost = tf.square(Y-y_model)

    train_op =
tf.train.GradientDescentOptimizer(learning_rate).minimize(cost)

    sess = tf.Session()
    init = tf.global_variables_initializer()
    sess.run(init)

    for epoch in range(training_epochs):

```

```
        for (x, y) in zip(x_train, y_train):
            sess.run(train_op, feed_dict={X: x, Y: y})

w_val = sess.run(w)

sess.close()
plt.scatter(x_train, y_train)
y_learned = [0, 0, 0, 0, 0, 0 ]
y_learned[0] = x_train[0]*w_val
y_learned[1] = x_train[1]*w_val
y_learned[2] = x_train[2]*w_val
y_learned[3] = x_train[3]*w_val
y_learned[4] = x_train[4]*w_val
y_learned[5] = x_train[5]*w_val
print("w = ")
print(w_val)
print(y_learned)
plt.plot(x_train, y_learned, 'r')
plt.show()
publish(/HOME/timew, payload=w_val, qos=0, retain=False)
```

ВІДГУК

на кваліфікаційний проект
студента групи ТР-17-1 Олега СВАЧІЯ

МОДУЛЬ КЕРУВАННЯ ПАРАМЕТРАМИ СИСТЕМИ SMART HOUSE НА ОСНОВІ ПОВЕДІНКИ МЕШКАНЦІВ

У проекті студента Свачія О.І. предметом дослідження є апаратна та програмна реалізація методу машинного навчання у системах Smart House. На сьогоднішній день, це доволі стрімко набуває популярності. Для подальшого розвитку даної сфери фірми починають розвивати напрямок машинного навчання. Основним з задач машинного навчання є передбачення поведінки людини.

Проектована система містить у собі такі інтерфейси передачі даних, як OneWire, I2C, SPI. Варто зазначити, на головному контролері знаходиться програмне забезпечення OpenHAB, зв'язок з яким відбувається за допомогою мережі Ethernet по MQTT протоколу.

Модуль реалізований на базі мікрокомп'ютера Raspberry Pi, а сама система розумного будинку реалізована за допомогою плат розробника Arduino та модулів, що орієнтовані на дані плати.

У ході виконання проекту було створену структурну схему та схему електричних підключень усієї системи Smart House.

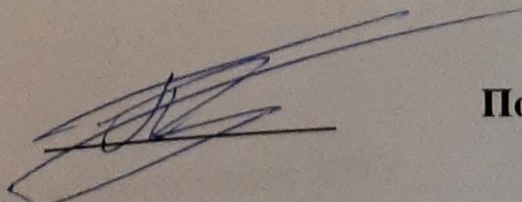
Для прогнозування поведінки людини використовується метод лінійної регресії. Написання програмного забезпечення відбувалось за допомогою бібліотек, що суттєво полегшували написання та реалізацію. TensorFlow у даному проекті виступала основною бібліотекою, що відповідає за машинне навчання методом лінійної регресії.

Під час виконання роботи студент Олег СВАЧІЙ проявив креативність, наполегливість, вміння застосовувати набуті знання для вирішення завдань, що були поставлені у кваліфікаційному проекті.

Кваліфікаційний проект на тему «Модуль керування параметрами системи Smart House на основі поведінки мешканців» виконано на високому рівні. Він є актуальним на сьогоднішній день, а студент Свачій О.І. заслуговує оцінки **«відмінно»**.

Керівник:

д.т.н., проф.



Полікаровських О.І.

РЕЦЕНЗІЯ

опонента на бакалаврський кваліфікаційний проект виконаний за темою
«Модуль керування параметрами системи Smart House на основі
поведінки мешканців» студента гр. ТР-17-1 Свачія О.І.

Тема кваліфікаційного проекту є доволі актуальною. У проекті студента Свачія О.І. проведено розробку апаратного та програмного забезпечення для системи Smart House. На сьогоднішній день, системи Smart House доволі стрімко набувають популярності. Та очевидно, що для того щоб бути конкуренто спроможними фірми починають розвивати напрямок машинного навчання. Це передбачує у собі автоматичне налаштування системи під різні сценарії, що передбачені у поведінці людини.

У ході виконання проекту було створено структурну схему та схему електричних підключень усієї системи Smart House. Проектована система містить у собі велику кількість інтерфейсів передачі даних, таких як SPI, I2C, OneWire. Варто зазначити, що на контролері який виконує одночасно роль сервера системи знаходиться програмне забезпечення OpenHAB, зв'язок з яким відбувається по MQTT протоколу.

Проектований модуль виконаний з мікрокомп'ютера Raspberry Pi, а сама система Smart House реалізована за допомогою плат розробника, таких як Arduino NANO, Arduino UNO, Arduino MEGA та модулів, що орієнтовані на плати Arduino.

Для прогнозування поведінки людини використовується метод лінійної регресії. Для написання програмного забезпечення використовувалось багато бібліотек для полегшення написання. Основною бібліотекою, що відповідає за машинне навчання методом лінійної регресії є TensorFlow.

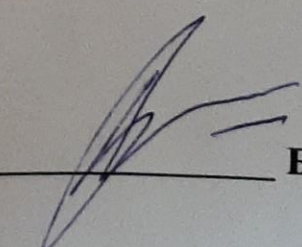
Система автоматично фіксує час коли мешканець повертається додому. Після чого контролер записує дані в масив що відповідає за аргумент функції. Навчання системи відбувається на протязі 5 днів. Після завершення навчання, написане програмне забезпечення розраховує коефіцієнт нахилу лінії та встановлює час увімкнення опалення.

Дана робота показує ефективність реалізації методів машинного навчання. Адже реальні значення відрізнялись від бажаних не більше як на 20хв.

В цілому проект на тему «Модуль керування параметрами системи Smart House на основі поведінки мешканців» виконано на рівні, що відповідає бакалаврському кваліфікаційному проекту. Він є актуальним у сучасному світі, а студент Свачій О.І. заслуговує оцінки «відмінно»

д.т.н., проф.

кафедри телекомунікацій та радіотехніки ХНУ


Бойко Ю.М.

Имя пользователя:
Kafedra TMIT KhNU

ID проверки:
1008250370

Дата проверки:
09.06.2021 22:03:49 EEST

Тип проверки:
Doc vs Internet

Дата отчета:
09.06.2021 22:09:21 EEST

ID пользователя:
100005657

Название файла: Свачий_TP17-1

Количество страниц: 83 Количество слов: 11568 Количество символов: 82219 Размер файла: 4.46 MB ID файла: 1008322295

Обнаружены модификации текста (могут влиять на процент совпадений)

5.2% Совпадения

Наибольшее совпадение: 1.01% с Интернет-источником (<https://vencon.ua/ua/articles/rejting-sistem-umnyy-dom-po-p..>

5.2% Источники из Интернета

757

Страница 85

Поиск совпадений с Библиотекой не производился

0% Цитат

Не найдено ни одной цитаты

Не найдено ни одной ссылки

0% Исключений

Нет исключенных источников

Модификации

Обнаружены модификации текста. Подробная информация доступна в онлайн-отчете.

Замененные символы

2

Подозрительное форматирование

15
страниц

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 1.0%

Словари проверки: en_US, ru_RU, ua_UA. Ошибок в документах: 9%

ID: 93112 Название: Модуль керування параметрами системи Smart House на основі поведінки мешканців Добавлено в БД: 2021-06-10 Авторы: Свачій Олег Ігорович Руководители: Полікарівських Олексій Ілліч Консультанты: Оponentы:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	51909	841	499 (1%)	9 (1%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

Завідувачу кафедри телекомунікацій,
медійних та інтелектуальних технологій
Д-р техн. наук, доц. Підченку С. К.
здобувача вищої освіти
Свачій О.І.
ФПКТС, гр. ТР-17-1

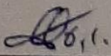
ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних проектів здобувачів вищої освіти на плагіат оповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах проектів.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія мого проекту збігається (ідентична) з друкованою.

25 травня 2021 р.
дата


підпис

РІШЕННЯ КАФЕДРИ ТЕЛЕКОМУНІКАЦІЙ, МЕДІЙНИХ ТА ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ ПРОЕКТУ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:
Назва: Модуль керування параметрами системи Smart House на основі поведінки мешканців

Автор: Свачій Олег Ігорович

Спеціальність: 172 Телекомунікації та радіотехніка

Науковий керівник: д.т.н., проф. Полікаровських Олексій Ілліч

Після аналізу звіту подібності зроблено такий висновок:

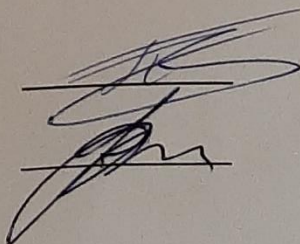
№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних). Робота приймається до захисту.	-
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	відповідає
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	-
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	-
5	Інше:	-

Підтвердження: Виявленні запозичення не є плагіатом так як розміщені в розділах, які не описують безпосередньо авторське дослідження (є власні терміни, визначення тощо), складають 5,2% та мають посилання на приведений список літературних джерел.

«0» червня 2021 р.

Науковий керівник

Завідувач кафедрою ТМІТ



Полікаровських О.І.

Підченко С.К.