

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

ДЕМЧИШИН Владислав Володимирович

**Система прогнозування концентрації CO₂ за допомогою IoT для
моніторингу якості повітря в приміщеннях / CO₂ concentration
forecasting system using IoT for indoor air quality monitoring**

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

Дипломний проект

Виконав студент групи КН-42
В.В. Демчишин

Науковий керівник:
к.т.н., доцент І.В. Турченко

Дипломний проект допущено до
захисту

«___»_____ 2023 р.

Завідувач кафедри
_____ М.П. Комар

Тернопіль – 2023

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ М.П. Комар
« _____ » _____ 2022 р.

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ
ДЕМЧИШИН Владислав Володимирович
(прізвище, ім'я, по батькові)

1. Система прогнозування концентрації CO₂ за допомогою IoT для моніторингу якості повітря в приміщеннях / CO₂ concentration forecasting system using IoT for indoor air quality monitoring
керівник проекту к.т.н., доцент І.В. Турченко

затверджені наказом по університету від 08 грудня 2022 р. № 491.

2. Строк подання студентом закінченого проекту 01 червня 2023 р.

3. Вихідні дані до проекту: технічне завдання.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

- опис характеристик систем контролю якості повітря;
- опис і характеристика існуючих аналогів систем моніторингу CO₂ та прогнозування його вмісту;
- постановка задачі та формування вимог до системи прогнозування концентрації CO₂ за допомогою ІОТ для моніторингу якості повітря в приміщеннях;
- розробка архітектури системи прогнозування концентрації CO₂ за допомогою ІОТ;
- розробка алгоритмічного забезпечення системи;
- розробка моделі прогнозування на основі lstm;
- дослідження моделі стаціонарного стану CO₂
- розробка програмно технологічного забезпечення системи;

- проведення тестування системи прогнозування концентрації CO₂
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
- архітектура системи прогнозування концентрації CO₂ за допомогою іот для моніторингу якості повітря в приміщеннях;
 - алгоритм роботи програми контролера для обробки даних с сенсора CO₂.

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Н. контроль	к.т.н., доцент І.В. Турченко		

7. Дата видачі завдання 08 грудня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту	Строк виконання етапів проекту	Примітка
1	Аналіз систем контролю та прогнозування якості повітря в приміщеннях	30.12.2022	
2	Розробка архітектури, алгоритмічного та інформаційного забезпечення системи	24.03.2023	
3	Програмно-технологічне забезпечення системи прогнозування концентрації co ₂ за допомогою іот для моніторингу якості повітря в приміщеннях	12.05.2023	
4	Повне завершення та оформлення дипломного проекту	01.06.2023	

Студент _____ В.В. Демчишин
(підпис)

Керівник проекту _____ І.В. Турченко
(підпис)

РЕФЕРАТ

Пояснювальна записка до дипломного проекту: 57 с., 37 рис., 3 додатки, 30 джерел.

Метою є розробка системи прогнозування концентрації CO₂ за допомогою IoT для моніторингу якості повітря в приміщеннях.

В роботі використовувалися методи системного аналізу і синтезу, порівняння, логічного узагальнення результатів, методи аналізу даних.

Було розроблено: архітектуру системи прогнозування концентрації CO₂ за допомогою IoT; алгоритмічне забезпечення системи; алгоритм читання даних з сенсора та відправки їх на сервіс; алгоритм роботи серверної частини; адаптовано модель прогнозування на основі LSTM; підібрано програмно-апаратні засоби для розробки системи; розроблено ПЗ для модуля визначення концентрації CO₂; розроблено ПЗ та модель мережі розумних сенсорів CO₂; створено канал для збору даних від розумних сенсорів; налаштовано логіку спрацьовування реакцій на каналі thinkspeak; інтегровано скрипт прогнозування в канал thinkspeak; протестовано роботу системи в цілому

Ключові слова: IOT, МОДЕЛЬ ПРОГНОЗУВАННЯ, THINKSPEAK, СИСТЕМА, КОНТРОЛЬ ЯКОСТІ ПОВІТРЯ, ПРОГНОЗУВАННЯ, CO₂.

ABSTRACT

The bachelor's thesis report: 57 pages, 37 figures, 3 appendices, 30 sources.

The aim is to develop a system for predicting CO₂ concentration using IoT to monitor indoor air quality.

Methods of system analysis and synthesis, comparison, logical generalization of results, data analysis methods were used in the work.

The following were developed: architecture of the CO₂ concentration forecasting system using IoT; algorithmic support for the system; algorithm for reading data from the sensor and sending it to the service; algorithm for the server part; adapted the forecasting model based on LSTM; selected software and hardware for system development; software for the module for determining the CO₂ concentration was developed; software and a model of the network of smart CO₂ sensors were developed; a channel for collecting data from smart sensors was created; the logic of triggering reactions on the thinkspeak channel was configured; the forecasting script was integrated into the thinkspeak channel; the system as a whole was tested

Keywords: IOT, FORECASTING MODEL, THINKSPEAK, SYSTEM, AIR QUALITY CONTROL, FORECASTING, CO₂.

ТЕХНІЧНЕ ЗАВДАННЯ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

1.1 Система прогнозування концентрації CO₂ за допомогою IoT для моніторингу якості повітря в приміщеннях.

1.2 Область застосування –аналіз та прогнозування концентрації CO₂ в приміщеннях.

2. ОСНОВА ДЛЯ РОЗРОБЛЕННЯ

Основою для розроблення є завдання на дипломний проект, затверджене кафедрою інформаційно-обчислювальних систем і управління факультету комп'ютерних інформаційних технологій Західноукраїнського національного університету.

3. ПРИЗНАЧЕННЯ РОЗРОБЛЕНОГО КОМПЛЕКСУ

Метою дипломного проекту є розробка системи прогнозування концентрації CO₂ за допомогою IoT для моніторингу якості повітря в приміщеннях.

4. ДЖЕРЕЛА РОЗРОБЛЕННЯ

Джерелами даної розробки є матеріали навчальної і реферативної літератури, технічна документація, науково-дослідні статті, журнали, Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

5.1 Основні функціональні вимоги до програмної системи:

- Використання датчиків Інтернету речей для системи моніторингу повітря в приміщенні в режимі реального часу.

- Застосування LSTM для прогнозування або передбачення майбутніх рівнів CO₂ на основі зібраних або історичних даних.

– Визначення стаціонарного рівня CO₂ в приміщенні за допомогою датчиків.

5.2 Вимоги до апаратних засобів:

– Сенсор BME680 IAQ, Adafruit SCD-30 – NDIR, Arduino, ESP8266.

5.3 Вимоги до програмних засобів:

– Autodesk Tinkercad Circuit, ThingSpeak, Cisco Packet Tracer

6. ПОРЯДОК КОНТРОЛЮ

6.1 Представлення дипломного проекту на попередній захист.

6.2 Представлення дипломного проекту на захист.

Завдання прийняв до виконання _____ В.В. Демчишин
(підпис) (прізвище та ініціали)

Керівник дипломного проекту _____ І.В. Турченко
(підпис) (прізвище та ініціали)

ЗМІСТ

Вступ.....	9
1 Аналіз систем контролю та прогнозування якості повітря в приміщеннях.	10
1.1 Характеристики систем контролю якості повітря.....	10
1.2 Огляд і характеристика існуючих аналогів систем моніторингу CO ₂ та прогнозування його вмісту.	12
1.3 Постановка задачі та формування вимог до системи прогнозування концентрації CO ₂ за допомогою іот для моніторингу якості повітря в приміщеннях.....	17
2 Розробка архітектури, алгоритмічного та інформацій- ного забезпечення системи	19
2.1 Архітектура системи прогнозування концентрації CO ₂ за допомогою ІоТ	19
2.2 Алгоритмічне забезпечення системи	23
2.3 Модель прогнозування на основі lstm	26
2.4 Модель стаціонарного стану CO ₂	28
3 Програмно-технологічне забезпечення системи прогнозування концентрації co ₂ за допомогою іот для моніторингу якості повітря в приміщеннях.....	31
3.1 Вибір та аналіз програмно-апаратних засобів для розробки системи....	31
3.2 Програмно технологічне забезпечення системи.....	37
3.3 Тестування системи прогнозування концентрації CO ₂	44
Висновки	48
Список використаних джерел	50
Додаток А код програми реалізації алгоритму LSTM	54
Додаток Б архітектура системи прогнозування концентрації CO ₂ за допомогою іот для моніторингу якості повітря в приміщеннях	56
Додаток В алгоритм роботи програми контролера для обробки даних с сенсора CO ₂	57

					<i>ДП.КН.9500086.057.ПЗ</i>			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Демчишин В.В.			Система прогнозування концентрації CO ₂ за допомогою ІоТ для моніторингу якості повітря в приміщеннях	Літ.	Арк.	Акрушів
Перевір.		Турченко І.В.					8	57
Реценз.						ЗУНУ.ФКІТ.КН-42		
Н. Контр.		Турченко І.В.						
Затверд.		Комар М.П.						

ВСТУП

За звичкою чи необхідністю люди проводять більшу частину свого часу в приміщенні. Накопичений вуглекислий газ (CO₂) може призвести до низки негативних наслідків для здоров'я, таких як нудота, головний біль, втома тощо. Таким чином, якість повітря в приміщенні необхідно контролювати з різних причин, пов'язаних зі здоров'ям. На ринку доступні різні системи моніторингу якості повітря. Однак, оскільки вони дорогі і важкодоступні, їх нечасто використовують в побуті. З появою Інтернету речей (IoT) системи моніторингу якості повітря в приміщенні (IAQ) спростилися. Зараз кожен може купити готовий розумний сенсор контролю якості повітря з можливістю моніторингу через інтернет і за допомогою мобільного додатку. Крім того існують так звані набори для розробників, де можна самому зібрати подібну систему з заданими характеристиками. Тому розробка недорогих систем моніторингу IAQ на основі Інтернету речей з використанням штучного інтелекту (ШІ) для аналізу вмісту CO₂ є актуальною задачею і на ринку таких систем спостерігається бурхливий розвиток.

Метою дипломного проекту є розробка системи прогнозування концентрації CO₂ за допомогою IoT для моніторингу якості повітря в приміщеннях.

Об'єктом дослідження є системи моніторингу якості повітря в приміщенні.

Предметом дослідження процес моніторингу та прогнозування CO₂ в приміщеннях.

Методи та технології розробки. Технологією розробки було обрано сервіс ThingSpeak для збору, аналізу та візуалізації даних, програму для симуляції мереж Cisco Packet Tracer.

Практичне значення одержаних результатів. В результаті проведеної роботи буде реалізовано: систему прогнозування концентрації CO₂ за допомогою IoT для моніторингу якості повітря в приміщеннях

					ДП.КН.9500086.057.ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ СИСТЕМ КОНТРОЛЮ ТА ПРОГНОЗУВАННЯ ЯКОСТІ ПОВІТРЯ В ПРИМІЩЕННЯХ

1.1 Характеристики систем контролю якості повітря

Термін "в приміщенні" означає простір всередині будівлі або споруди, наприклад, кафе, ресторану, будинку, офісу або навіть автомобіля. Незалежно від особистих чи професійних причин, погоди чи інших факторів, більшість людей проводять більшу частину свого часу в приміщенні. Згідно з дослідженням Агентства з охорони навколишнього середовища (EPA), американці проводять 93% свого часу в приміщенні [1]. Середньостатистичний громадянин Великобританії проводить 53 роки свого життя в приміщенні. Оскільки ці дослідження були опубліковані до пандемії COVID-19, цифри могли значно зрости через вплив факторів, пов'язаних з пандемією [2]. Дійсно, для деяких людей стало очевидним, що вони можуть працювати вдома замість того, щоб їздити на роботу.

Через те, що люди проводять більшу частину свого часу в приміщенні, якість повітря всередині будівель та інших споруд має значний вплив на їхнє здоров'я, комфорт і самопочуття. Погана якість повітря в приміщенні (IAQ) може викликати або сприяти розвитку алергії, астми та інших респіраторних захворювань. Вона також може викликати головний біль, запаморочення, нудоту та втому [3].

На IAQ можуть впливати гази, частинки, пліснява, бактерії та інші забруднювачі. Кілька джерел забруднення відповідають за поганий IAQ, як всередині, так і зовні приміщень.

Зовнішні джерела забруднення повітря, такі як транспортні та промислові викиди, можуть потрапляти в будівлі через двері, вікна та вентиляційні системи [4].

Внутрішні джерела забруднення включають прилади для спалювання, будівельні матеріали, засоби для чищення та офісне обладнання. CO₂ вважається важливим і поширеним показником для вимірювання якості

					ДП.КН.9500086.057.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

повітря [5,6]. Викиди CO₂ утворюється кожного разу, коли людина видихає. Він також виробляється приладами для спалювання палива та деяким офісним обладнанням. Рівень CO₂ у повітрі в приміщенні, як правило, вищий, ніж у зовнішньому повітрі. Багато досліджень показали, що CO₂ може призвести до зниження когнітивних здібностей [7], а високі концентрації CO₂ призводять до ряду проблем, пов'язаних зі здоров'ям, таких як нудота, головний біль і втома [8]. Нещодавні дослідження також показують, що концентрація CO₂ може впливати на ризик зараження COVID-19 [9]. Це пов'язано з тим, що як патогени, так і CO₂ видихаються інфікованими людьми, що призводить до того, що CO₂ в приміщенні ефективним проксі для визначення потенційного ризику інфікування. Тому моніторинг рівня CO₂ було запропоновано, як індикатор для непрямої оцінки потенційного ризику передачі респіраторних інфекційних захворювань [10]. Вивчення рівня CO₂ в приміщенні також має відношення до питання підрахунку зайнятості приміщень, що може в подальшому вплинути на енергоспоживання будівлі [11].

Використання CO₂ для визначення зайнятості є перспективним підходом, який вивчався в декількох дослідженнях [12,13,14]. Точне визначення зайнятості будівлі є важливим для досягнення енергозбереження, а в деяких випадках може призвести до зменшення споживання енергії на 30-40% [15]. Розуміння викидів CO₂ в приміщеннях в приміщенні має вирішальне значення для забезпечення здоров'я та безпеки людей, які там перебувають. Урядові нормативні акти та промислові інструкції встановлюють різні допустимі рівні CO₂ для приміщень. Наприклад, безпечна межа становить не більше 1000 ppm CO₂ зазвичай використовується в багатьох сферах [16], в той час, як європейські стандарти встановлюють 1500 ppm як максимально допустиму концентрацію CO₂ для IAQ в приміщеннях [17]. Оскільки значна кількість CO₂ виділяється при диханні мешканців, необхідно мати ефективну систему очищення повітря від CO₂ а також систему прогнозування для ефективного запобігання пов'язаних з цим ризиків для здоров'я.

					ДП.КН.9500086.057.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

З розвитком Інтернету речей, недорогі датчики та платформи з відкритим вихідним кодом стали широко доступними і можуть бути інтегровані разом з технологіями штучного інтелекту в системи IAQ. Тому до таких систем ставляться вимоги не тільки моніторингу викидів, а й прогнозування можливих піків концентрації CO₂ і крім того вони мають підтримувати можливість доступу і передачі даних через протокол TCP/IP.

На даний час майже усі системи контролю якості повітря використовують датчики Інтернету речей для створення системи моніторингу повітря в приміщенні в режимі реального часу. Для прогнозування концентрації газів використовуються різні алгоритми прогнозування, наприклад LSTM для прогнозування або передбачення майбутніх рівнів CO₂, а системи надають рекомендації щодо здоров'я та благополуччя.

1.2 Огляд і характеристика існуючих аналогів систем моніторингу CO₂ та прогнозування його вмісту.

У сфері будівель, моніторинг викидів CO₂ важливий для різних застосувань, включаючи управління системами опалення, вентиляції та кондиціонування повітря (HVAC) [18] (рисунок 1.1), прогнозування заповнюваності приміщень, набори даних про експлуатацію будівель, а також аналіз обчислювальної гідродинаміки (CFD).

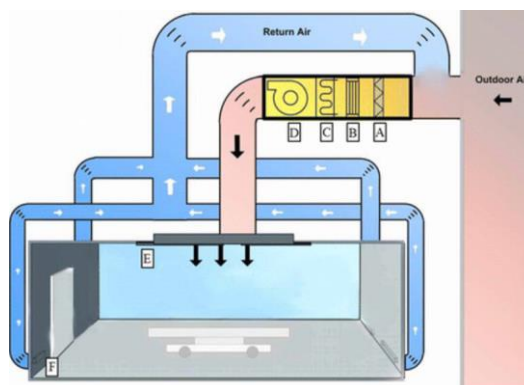


Рисунок 1.1 - Система опалення, вентиляції та кондиціонування повітря

					ДП.КН.9500086.057.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

Використовуючи CO₂ у поєднанні з іншими даними про будівлю, можна покращити експлуатаційні характеристики будівлі та її енергоефективність. Автори роботи [19] використовували генетичні алгоритми та варіації викидів CO₂ для оптимізації роботи стандартних систем опалення, вентиляції та кондиціонування повітря з точки зору економії енергії. Поєднуючи генетичні алгоритми з CO₂ вони змогли досягти зниження витрат на кулер на 21% і зменшити максимальний потік води через охолоджувальний змішувач на 83%.

CO₂ також відіграє ключову роль у прогнозуванні заповнюваності будівлі. Крім того, емпіричні результати, наведені в [20], показали, що концентрація CO₂ в приміщенні постійно входив до топ-15 найважливіших характеристик для прогнозування заповнюваності для всіх типів приміщень. Точне визначення заповнюваності будівлі може призвести до значного скорочення споживання енергії з потенційною економією до 30-40% [21].

Автори роботи [22] створили комплексні набори даних про експлуатацію будівель, що містять дані про викиди CO₂ з метою забезпечення унікальної перспективи експлуатації будівлі з нульовим споживанням енергії та встановлення корисного еталону для порівняння з існуючими будівлями.

У дослідженні [23] (рисунок 1.2), був розроблений недорогий вимірювальний пристрій для дистанційного моніторингу IAQ. Крім того, для дослідження потоків повітря в приміщенні та визначення потенційних місць вимірювання було використано CFD-моделювання.

CO₂ є найпоширенішим параметром повітря в приміщенні, пов'язаним з IAQ, і широко використовується для визначення комфорту та виявлення якості навколишнього середовища в приміщенні (IEQ) [24].

					ДП.КН.9500086.057.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

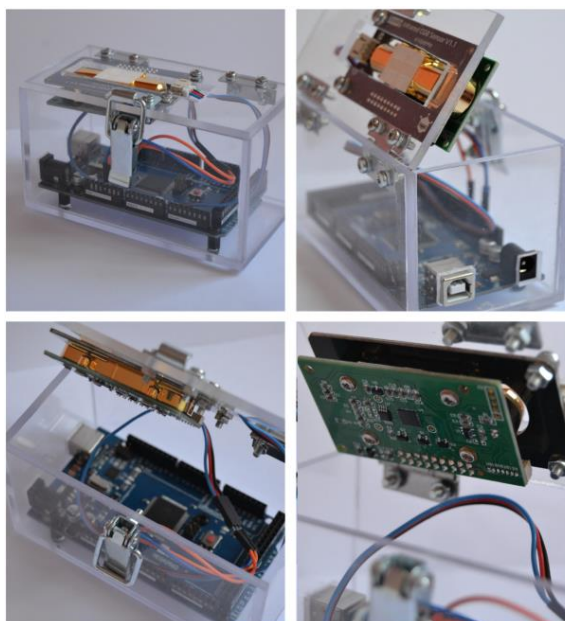


Рисунок 1.2 - Пристрій для дистанційного моніторингу IAQ

На основі датчиків IAQ та стратегії машинного навчання автори [25] (рисунок 1.3) запропонували рішення для моніторингу IAQ, що полягає в наданні користувачам доступу до веб-порталу та мобільного додатку, які відображають візуальне представлення якості повітря. У їхньому дослідженні загалом п'ять параметрів повітря (CO_2 , CO , NO_2 , CH_4) були розраховані та класифіковані відповідно до рівня IAQ.

Використовуючи нейронні мережі, автори класифікували умови IAQ з точністю 99,1%. Довготривала короткочасна пам'ять (LSTM) також була використана для прогнозування майбутніх викидів CO_2 у майбутньому. Однак отримані результати класифікації IAQ базувалися на індексі якості зовнішнього повітря (AQI), який залежить від багатьох інших параметрів і може бути непридатним для внутрішнього середовища. Крім того, їхній метод може становити проблему через термін служби датчика та фактори калібрування.

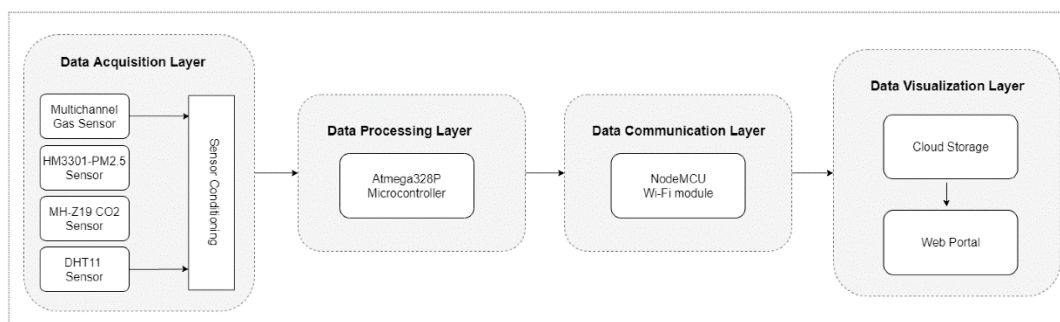


Рисунок 1.3 - Апаратна архітектура вузла моніторингу якості повітря в приміщенні

З іншого боку, на основі LSTM і його варіанту Gated Recurrent Unit (Повний рекурентний вузол) в дослідженні [26] автори розробили систему прогнозування IAQ (рисунок 1.4).

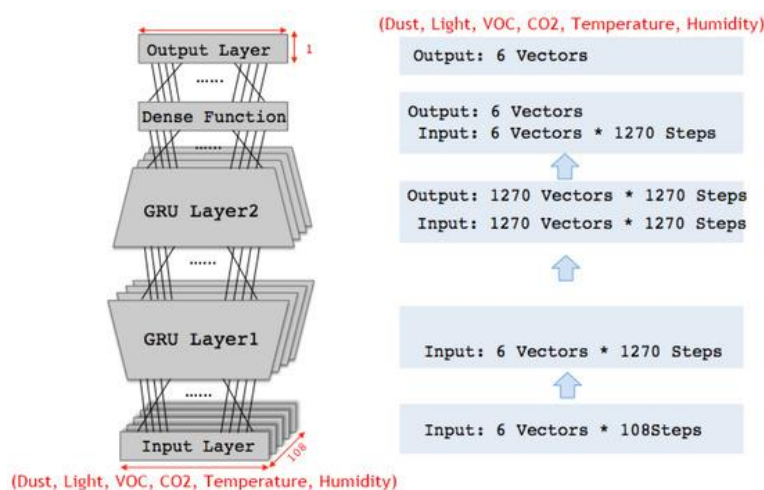


Рисунок 1.4 - Мережа закритих рекурентних одиниць (GRU) для прогнозування якості повітря

У цій роботі автори виявили, що GRU має кращу продуктивність, ніж LSTM, і досягли точності до 84,69% при використанні GRU. Однак ця модель витрачає 38 годин на пошук розміру крок. Крім того, архітектура для збору даних через мережу IoT була запропонована в роботі [27], де також було побудовано дві моделі прогнозування: одна для прогнозування комфортних умов протягом дня щодо температури, вологості та CO2 а інша - для

прогнозування викидів CO2 за допомогою нейронних мереж. Згідно з їхніми результатами, середньоквадратична похибка (MSE) протягом тестового періоду становить близько 75 ppm (10,6%) порівняно з середньою концентрацією CO2.

В іншому дослідженні [28] (рисунок 1.5) дослідники порівнювали дві різні концентрації CO2 і виявили, що дерево рішень є більш ефективним, ніж штучна нейронна мережа (ШНМ) з точки зору обчислень та споживання енергії.

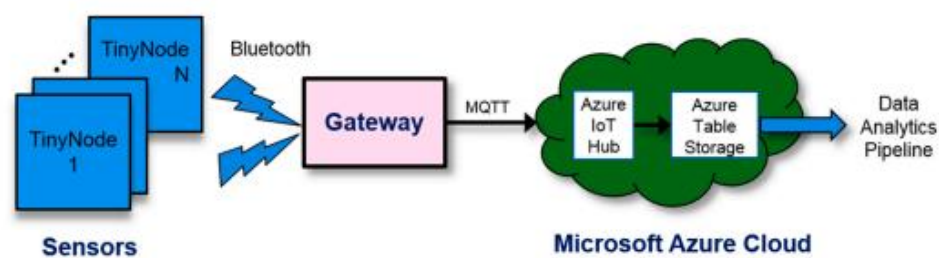


Рисунок 1.5 – Система збору даних

Крім того, застосування стратегії часового вікна прогнозування на одну хвилину вперед має найвищу точність порівняно з часовим вікном на десять або п'ятнадцять хвилин вперед. Однак додавання інших змінних, таких як температура і вологість, не покращує точність прогнозування. На основі динамічного мобільного вікна (рисунок 1.6) автори [29] запропонували

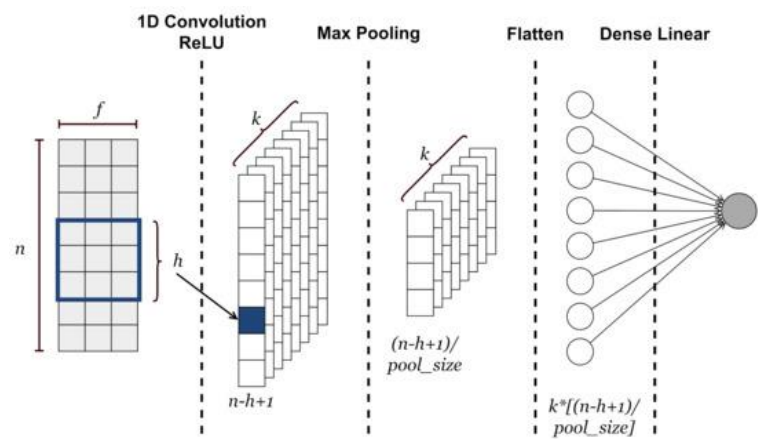


Рисунок 1.6 - Архітектура нейронної мережі

поступово оновлювану модель прогнозування викидів CO₂ модель прогнозування викидів CO₂. Ця модель може покращувати свою точність на щоденній основі. Використовуючи власну модель прогнозування викидів CO₂ вони інтегрували цю модель на периферійному пристрої, який здатен оновлювати модель і прогнозувати майбутні рівні CO₂.

У більшості досліджень основна увага приділяється створенню системи збору даних для внутрішнього середовища та прогнозуванню параметрів повітря, а також класифікації рівня забрудненості на основі прогнозованих параметрів повітря. Однак не існує єдиного стандарту для класифікації IAQ, а результати прогнозування на довшому часовому проміжку є гіршими, ніж на коротшому.

1.3 Постановка задачі та формування вимог до системи прогнозування концентрації CO₂ за допомогою IoT для моніторингу якості повітря в приміщеннях

У даному дипломному проекті пропонується створити недорогу систему моніторингу якості повітря на основі Інтернету речей з використанням штучного інтелекту (ШІ) для надання рекомендацій.

У системі датчики Інтернету речей передаватимуть дані через протокол передачі телеметрії в черзі (Message Queuing Telemetry Transport, MQTT), які можна візуалізувати в режимі реального часу на зручній для користувача інформаційній панелі, яку надає ThingSpeak.

Крім того, до зібраних даних про викиди CO₂ буде застосовуватись метод штучного інтелекту, який називається довготривалою короткочасною пам'яттю (LSTM) з метою прогнозування майбутніх викидів CO₂.

На основі прогнозованої концентрації CO₂ система може обчислити концентрацію CO₂ заздалегідь з рекомендованою похибкою (приблизно 5,5%).

					ДП.КН.9500086.057.ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

В даній роботі буде розроблена не тільки IoT-орієнтована система викидів CO₂ в реальному часі, що працює в режимі реального часу але й інтелектуальна підпрограма прогнозування викидів CO₂ та концентрації CO₂ стаціонарного стану або рівноваги. Тобто пропонується по суті система яка складається з двох частин:

- Використання датчиків Інтернету речей для створення системи моніторингу повітря в приміщенні в режимі реального часу.
- Застосування LSTM для прогнозування або передбачення майбутніх рівнів CO₂ на основі зібраних або історичних даних. Після цього стаціонарний рівень CO₂ в приміщенні може бути визначений за допомогою датчиків. Далі система може надати рекомендації щодо здоров'я та благополуччя.

Оскільки метою дипломного проекту є створення системи прогнозування концентрації CO₂ за допомогою IoT для моніторингу якості повітря в приміщеннях:

- дослідити системи контролю та прогнозування якості повітря в приміщеннях;
- розробити архітектуру системи прогнозування концентрації CO₂ за допомогою IoT;
- розробити алгоритми функціонування ПЗ для системи;
- інтегрувати модель прогнозування на основі LSTM;
- інтегрувати модель стаціонарного стану CO₂ для системи;
- провести аналіз програмно-апаратних засобів для розробки системи
- розробити апаратно програмне забезпечення «клієнта» системи
- розробити програмне забезпечення «клієнта» системи для імітування мережі сенсорів
- розробити програмне забезпечення «сервера» системи
- провести тестування системи

2 РОЗРОБКА АРХІТЕКТУРИ, АЛГОРИТМІЧНОГО ТА ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

2.1 Архітектура системи прогнозування концентрації CO₂ за допомогою IoT

Враховуючи потсановку задачі в п 1.3 та опису вимог до системи загальна архітектура системи буде поділятися на дві глобальні частини (рисунок 2.1):

- клієнтська
- серверна



Рисунок 2.1 – Загальна архітектура системи

Як показано на рисунку клієнтська частина складається з наступних елементів:

- Група сенсори включає в себе сенсори CO₂ та по можливості сенсори температури і вологості, так як сама концентрація CO₂ також дуже залежить від цих двох параметрів, але групи температура-вологість може і не бути, тому що основний акцент все-таки на сенсор CO₂.

- Група мікроконтролер. В дану групу входять засоби на базі мікроконтролера які мають на борту АЦП або порти загального призначення для підключення різних типів сенсорів та іншого периферійного обладнання. Можуть використовуватись як недорогі MCU, які мають тільки декілька аналогових каналів та пару загального призначення, як і складні плати для

розробників які дозволяють інтегрувати в себе додаткові обчислювальні модулі, наприклад контролери мережевих пристроїв, бездротовий/дротовий зв'язок.

- Конвертація даних. В більшості випадків контролер не отримує від сенсора або АЦП точного значення фізичної величини, яка вимірюється, тому є необхідність набору правил отримання даних та переведення їх у реальну фізичну величину, наприклад CO2. На рисунку 2.2 представлено послідовність таких дій в залежності від типу сенсора (аналоговий або цифровий).

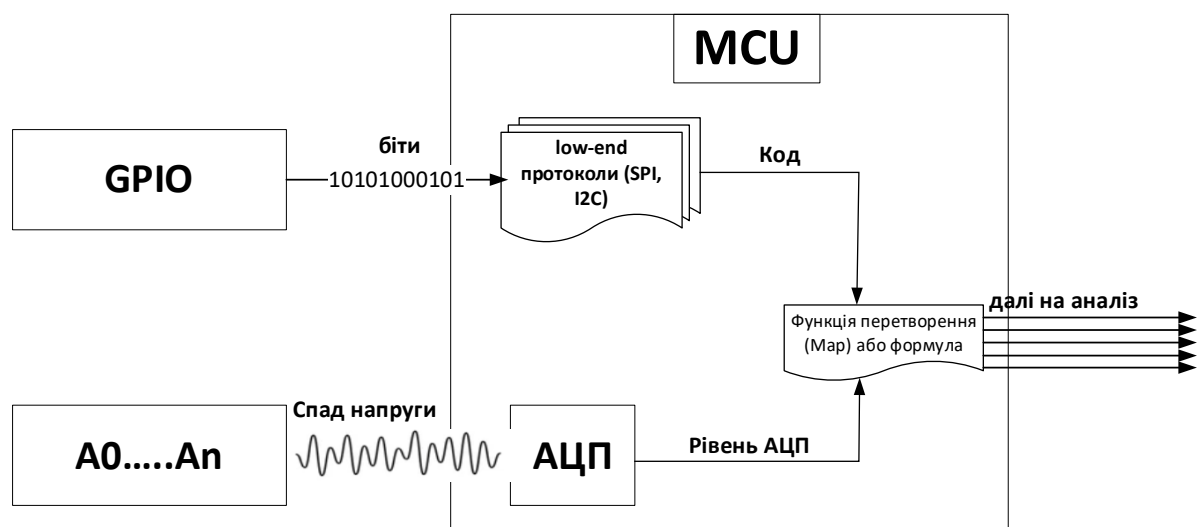


Рисунок 2.2 - Послідовність дій для перетворення фіз. величин в залежності від типу сенсора

- Група фізичні величини. Дана група вибрана тому, що в більшості випадків вимірювальні системи т системи аналізу оперують не тільки одним параметром, а цілою групою. Хоча в роботі 90% буде присвячено CO2 сенсорам і отримання саме цієї фізичної величини не можна ігнорувати інші супутні величини так як вони діють в комплексі і потрібно їх також враховувати.

Серверна частина складається з наступних частин:

Зберігання. Ця група забезпечує зберігання даних в БД. Вона може складатись з декількох підмодулів і залежати від конкретної серверної

архітектури. На рисунку 2.3 представлено архітектуру такого модуля.

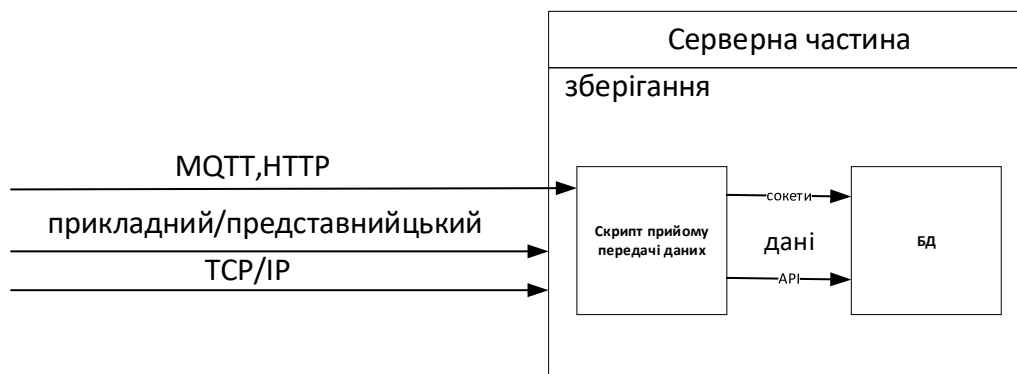


Рисунок 1.3 – Зберігання даних на стороні сервера

Як видно з рисунку безпосередньо зданими працює скрипт прийому-передачі даних, він обробляє запит відповідно до протоколу, через спеціальні API з'єднується з БД і записує дані, якщо в серверній частині не передбачено внутрішньої БД то скрипт може через сокети зберегти дані в іншому місці, наприклад на БД іншого сервера за допомогою сокетів.

Другим важливим елементом архітектури, яка була запропонована є візуалізація даних (рисунок 2.4)

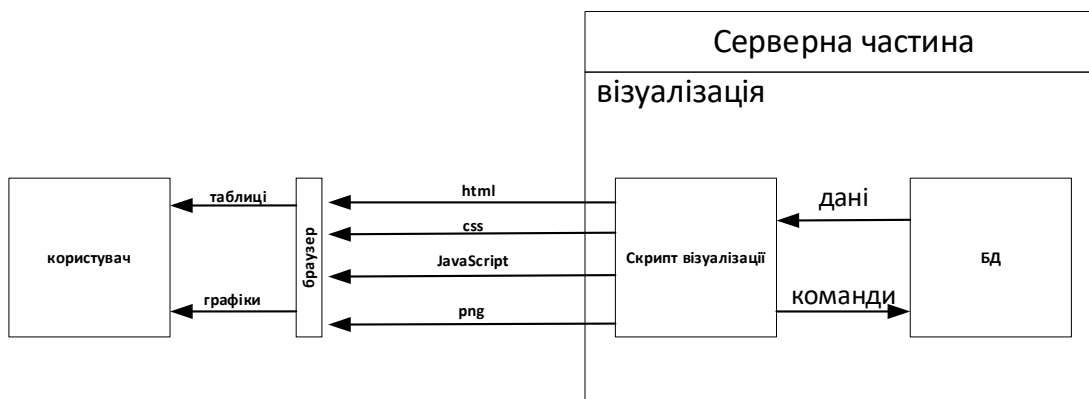


Рисунок 2.4 – Візуалізація даних

Основною задачею візуалізації є представлення інформації користувачеві у зрозумілому виді, тобто в якості графіків, діаграм чи таблиць. Сам процес візуалізації полягає в тому, що так само, як в прийомі і передачі даних працює скрипт-програма, яка отримує статистичні дані і генерує html

код сторінки, який передається вже на переглядач користувача. Сам процес візуалізації відбувається на стороні користувача.

Група зворотній зв'язок (рисунок 2.5). Дана група слугує для комунікації між аналітикою та зовнішніми модулями (клієнтською частиною) для відправки рекомендацій або команд.

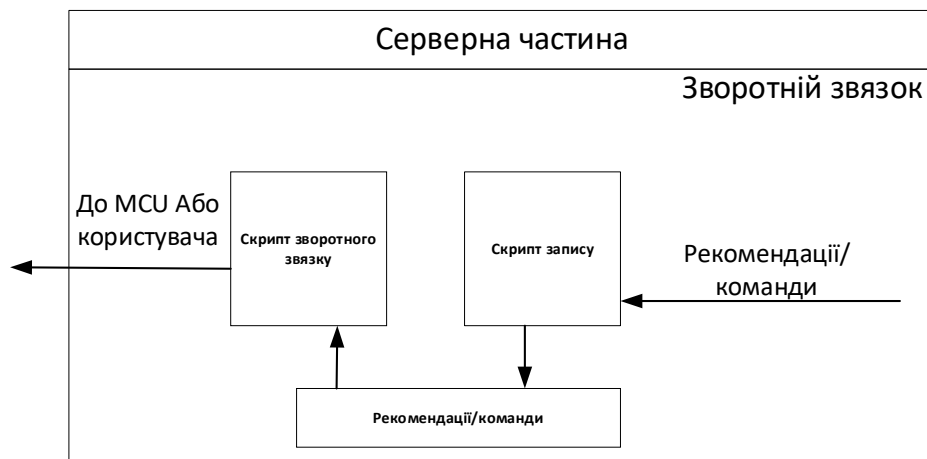


Рисунок 2.5 – Зворотній зв'язок

Група аналіз/передбачення (рисунок 2.6).

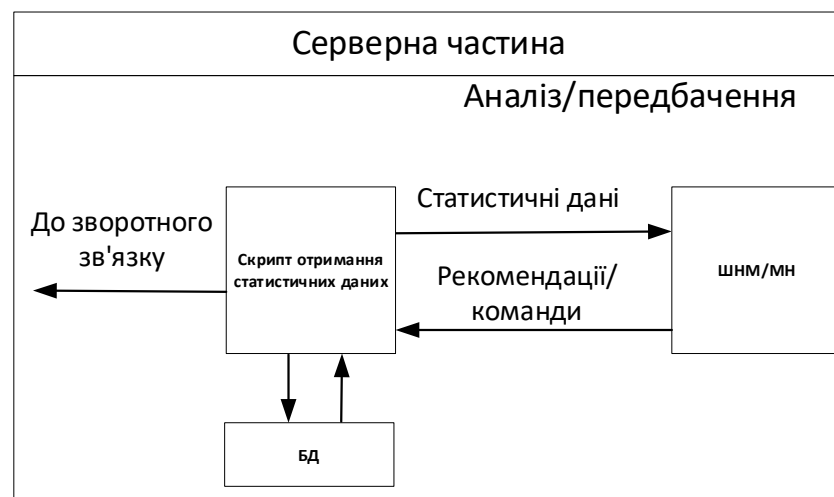


Рисунок 2.6 – Аналіз передбачення

Як показано на рисунку вище цей модуль має декілька каналів зв'язку (API або сокети). В блоці працює скрипт який по суті є двигуном, тобто він

сам нічого не вирішує, він тільки приєднується до БД статистичних даних, бере пакет даних, відправляє на ШМН, отримує результат і відправляє його далі по ланцюжку на блок зворотного зв'язку.

Зовнішні комунікації (рисунок 2.7). Основною задачею цього блоку є надання можливості використовувати додаткові можливості інших сервісів через протокол http.

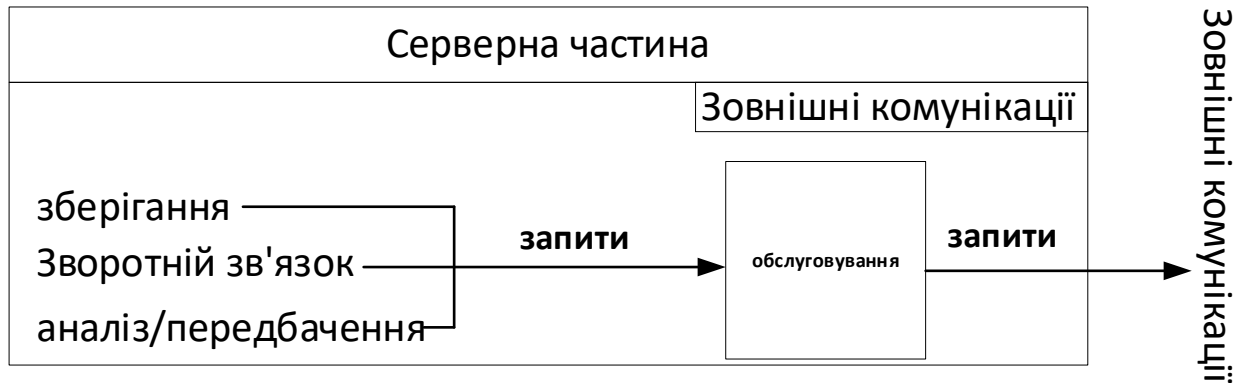


Рисунок 2.7 – Блок зовнішніх комунікацій

Тобто якщо сервіс не надає певного спектру послуг можна використати цей блок для підключення зовнішніх сервісів, виконанню на них додаткових завдань а повернення результатів. Даний блок не є обов'язковим, якщо система не планується розширятись і не змінювати архітектуру. У випадку гнучкої масштабованої системи даний блок є необхідний, тому що тільки декілька компаній в світі мають повнофункціональні хмарні сервіси, в інших випадках завжди є потреба в пошуку додаткового зовнішнього функціоналу.

2.2 Алгоритмічне забезпечення системи

Враховуючи вимоги до системи які були описані в розділі 1.3 та архітектуру в розділі 2.1 можна зрозуміти яким чином має працювати програмне забезпечення

Так як і в архітектурі системи, алгоритми функціонування також будуть поділятися на дві великі групи – це серверне і користувача, хоча в даному контексті користувачем може бути як людина так і мікроконтролер або зовнішній сервіс.

2.2.1 Середовище користувача

На риснку 2.8 представлено псевдокод роботи програми контролера для обробки даних с сенсора CO2

```

початок
налаштування
{
    Налаштування портів (2,3,4)
    Налаштування послідовного інтерфейсу (9600 бод)
    Налаштування клієнта WiFi
    цикл { wifi статус != приєднано}
    {
        затримка (500)
        вивід в послідовний порт «...»
    }
    Вивід в послідовний порт «Приєднано до WiFi»
}
головний цикл {}
{
    якщо (приєднання до сервера !=0)
    {
        Дані з порта = SPI
        стрічка запиту="ключ доступу"+" дані з порта "
        вивід в послідовний порт «приєднано до сервісу»
        запис в сокет "POST /update HTTP/1.1"
        запис в сокет стрічка запиту
        відповідь=відповідь сервера
        час=таймер
        якщо (час >20000 мс або відповідь == 200)
        {
            Отримання відповіді від сервера
        }
    }
    Стоп клієнт
    вивід в послідовний порт відповіді від сервера
    інакше
    {
        вивід в послідовний порт «неможливо приєднатись»
    }
    затримка (160000 мс)
}

```

Рисунок 2.8 – псевдокод програми читання даних з сенсора та відправки їх на сервіс

					ДП.КН.9500086.057.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

2.2.2 Серверна частина

Після того як дані отримані сервером йде наступний крок а саме збереження, аналіз даних та отримання рекомендацій. На рисунку 2.9 показано алгоритм роботи серверної частини.

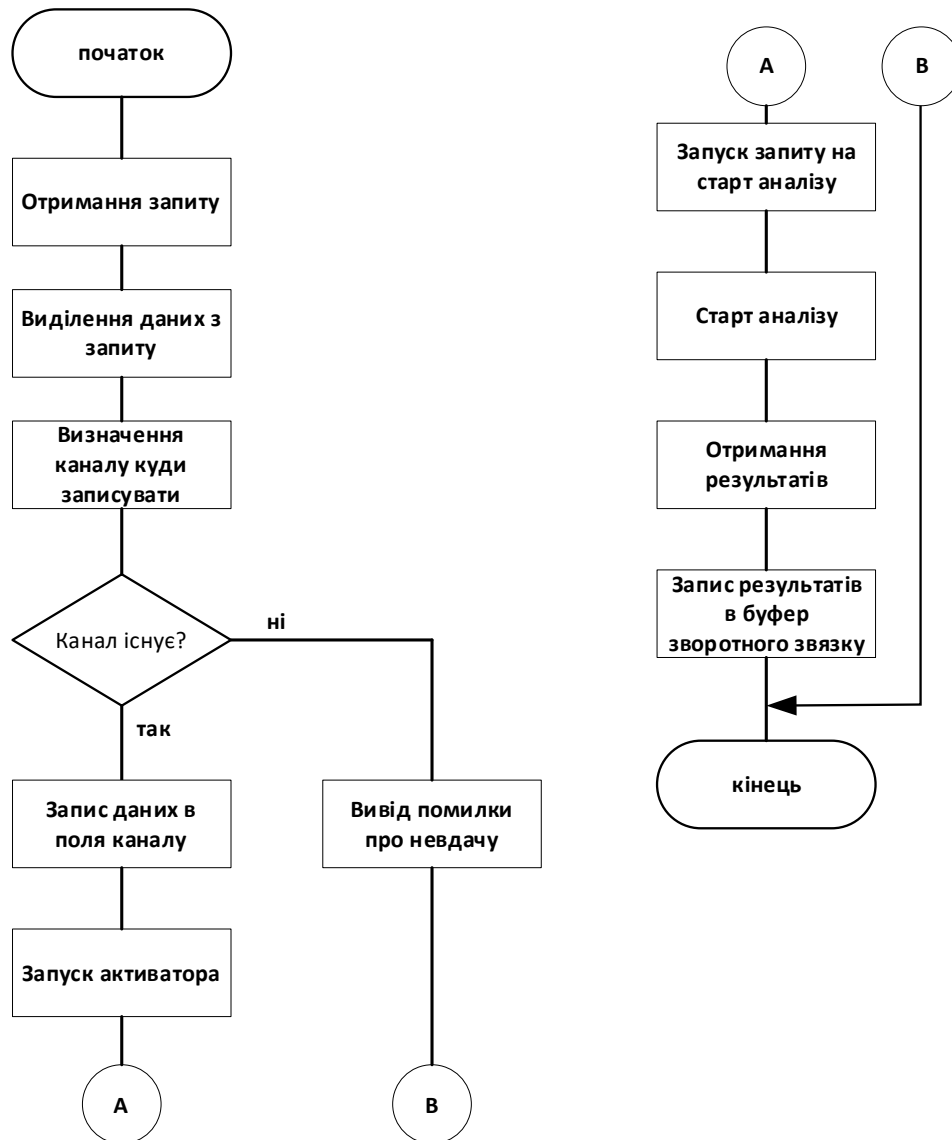


Рисунок 2.9 – Алгоритм роботи серверної частини

Даний алгоритм представляє роботу усього сервісу, по суті кожен блок даного алгоритму це окремий програмний модуль який працює в спільному середовищі сервісу.

Крім того щоб усі модулі потрібно дотримуватись певного алгоритму побудови ланцюгу виклику модулів інакше система не буде коректно

працювати. Нижче представлено алгоритм створення функціоналу в середовищі сервісу thingspeak:

1. Створення каналу
2. Призначення полів
3. Налаштування форми запиту для запису даних
4. Створення спеціального тригера
5. Створення скрипта аналізу
6. Створення запиту запуску скрипта
7. З'єднання тригера з запитом
8. Налаштування тригера на певну реакцію

Після усіх налаштувань усі елементи будуть працювати, як єдина система.

2.3 Модель прогнозування на основі LSTM

Рекурентні нейронні мережі (RNN) - це поширена модель прогнозування часових рядів, яка надає більше переваг, ніж інші типи нейронних мереж, що працюють з даними часових послідовностей. RNN складається з одношарових мереж з петлями, які можуть передавати часову інформацію від однієї мережі до іншої.

Вона може вивчати інформацію з попередніх даних і поєднувати її з поточними вхідними даними для прийняття рішення. Історичним обмеженням ШНМ є проблема зникнення градієнта, особливо при роботі з великою кількістю даних часових рядів [30]. Однак цю проблему можна подолати, використовуючи LSTM, який є оновленою версією RNN з додатковими елементами пам'яті.

LSTM (рисунок 2.10) складається з комірки та чотирьох шарів, а саме: вхідного шару, шару із затримкою, оновленого шару та вихідного шару. Додатковий вентиль забування обробляє вхідну інформацію в пам'яті протягом більш тривалого періоду часу в порівнянні з RNN. Використовуючи

					ДП.КН.9500086.057.ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

вентилі для керування потоком інформації та запобігання зникненню градієнтів.

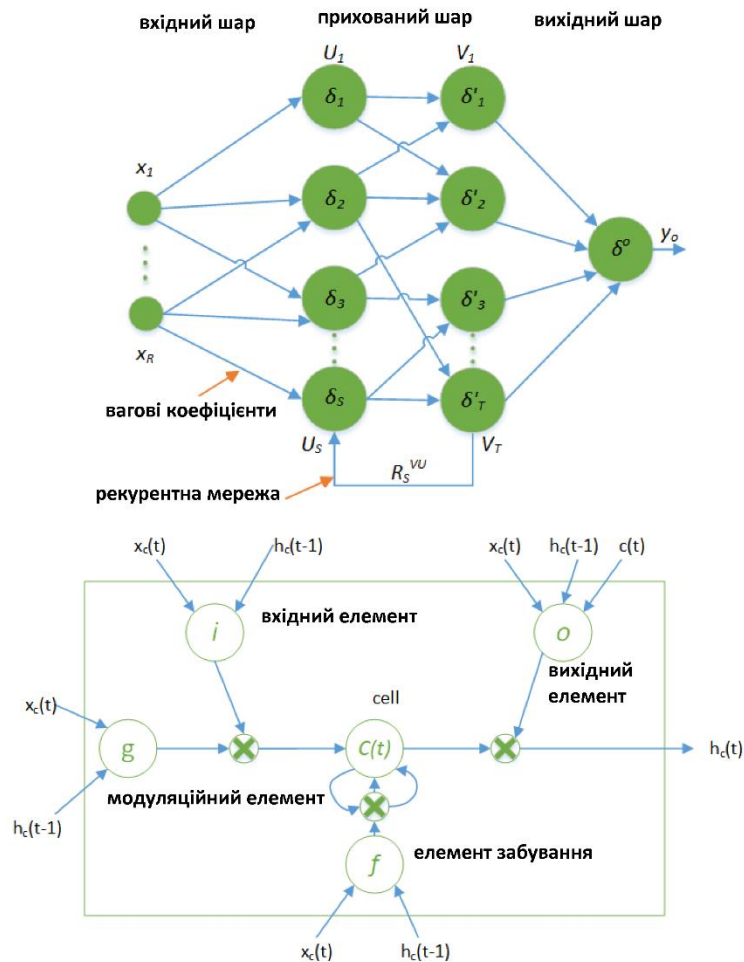


Рисунок 2.10 - Базова структура комірки LSTM-NN

Тут g , i , o та f позначають модуляційний елемент, вхідний елемент, вихідний елемент та елемент забування відповідно. У модуляційному елементі вхід оновлюється в поточний момент часу. Всі елементи працюють на поточному вході ($x(t)$) та попередніх значеннях станів ($h(t-1)$). Виходи елемент обчислюються наступним чином:

$$i(t) = f(W_i x_c(t) + U_i h_c(t-1) + b_i) \quad (1)$$

$$g(t) = \tanh (W_c x_c(t) + U_c h_c(t - 1) + b_c) \quad (2)$$

$$g(t) = f(W_f x_c(t) + U_f h_c(t - 1) + b_f) \quad (3)$$

де, W та U - ваги для вхідного та попереднього стану відповідно. Зсуви позначено символом b , індекс використовується для позначення конкретного входу. Оновлене значення клітинки обчислюється як:

$$c(t) = f(W_o x_c(t) + U_o h_c(t - 1) + V_o c(t) + b_o) \quad (4)$$

$$h(t) = O(t) \tanh (c(t)) \quad (5)$$

Як правило, якщо прихованих шарів більше одного, архітектура мережі може бути класифікована як глибока мережа.

LSTM здатні ефективно зберігати інформацію на тривалих часових інтервалах і вирішувати складні задачі з часовими рядами. Це робить їх кращими за традиційні ШНМ, які борються з цими проблемами. Тому LSTM вважається одним з найсучасніших алгоритмів у вирішенні проблем прогнозування часових рядів.

2.4 Модель стаціонарного стану CO₂

Стаціонарний стан є ключовим фактором у викидах CO₂ рівняння балансу маси, яке представляє максимальну кількість CO₂ в приміщенні. Він визначається кількістю людей та швидкістю вентиляції. Важливість вивчення рівняння балансу CO₂ в стаціонарному стані полягає в тому, що він використовується для вирішення питання заповнюваності, яке безпосередньо пов'язане з енергоспоживанням житлових та офісних будівель. Європейський стандарт передбачає, що концентрація 1500 ppm CO₂ є прийнятною верхньою межею IAQ. Тому, якщо максимальне значення CO₂ можна передбачити

					ДП.КН.9500086.057.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

заздалегідь, то проблеми зі здоров'ям, спричинені CO₂ можна запобігти. У приміщенні CO₂ утворюється в першу чергу через дихання мешканців, згідно з даними CO₂ відповідно до теорії балансу маси. CO₂ всередині будівлі залишається в динамічній рівновазі з зовнішнім повітрям протягом достатнього періоду часу, і цей рівень називається стаціонарним. Його диференціальне рівняння показано в рівнянні (6):

$$C(t) = (C_0 - C_{ss})e^{-\frac{Q}{V}t} + C_{ss}, \quad (6)$$

де C_0 - концентрація CO₂ в приміщенні при $t=0$; C_{ss} - стаціонарний стан CO₂; V - об'єм приміщення; C_{ot} - концентрація CO₂ ззовні; Q та G - об'ємна витрата повітря в приміщенні та швидкість генерації CO₂. Загалом, C_{ot} , Q та G є функціями часу, але в цій моделі вони вважаються постійними. C_{ss} можна визначити наступним чином:

$$C_{ss} = C_{ot} + \frac{G}{Q}, \quad (7)$$

Рисунок 2.11 ілюструє тенденцію рівнів концентрації CO₂ в приміщенні.

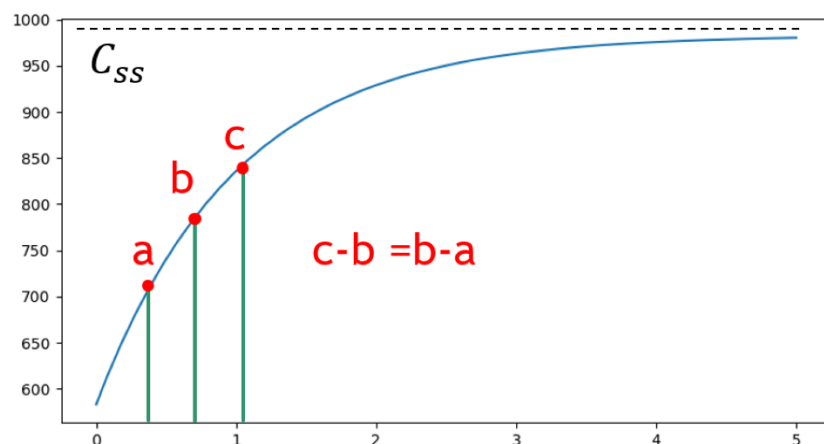


Рисунок 2.11 - Тенденції щодо викидів CO₂ в приміщеннях

Спостереження за кривою тренду CO₂ протягом достатньої кількості часу є одним із способів отримати стаціонарні значення C_{ss} . Однак рівняння (7) показує, що визначення стаціонарної концентрації зазвичай вимагає багато часу, оскільки вона залежить від частоти дихання G і потоку повітря Q . Другий метод отримання стаціонарних значень заздалегідь полягає у визначенні рівноважного рівня на основі кривої тренду. Це рівняння можна виразити наступним чином:

$$C_{ss} = \frac{C_b^2 - C_a C_c}{2C_b - C_a - C_c}, \quad (8)$$

де C_a , C_b і C_c - концентрації в рівномірно віддалених точках часу a , b і c . Теоретично, як видно з рисунку 3, стаціонарне значення C_{ss} може бути розраховане з будь-яких трьох рівномірно віддалених точок часу на всій осі часу. Для того, щоб уникнути помилок, спричинених фактичним вимірюванням CO₂, інтервал між C_a , C_b та C_c становить десять часових точок.

3 ПРОГРАМНО-ТЕХНОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ПРОГНОЗУВАННЯ КОНЦЕНТРАЦІЇ CO₂ ЗА ДОПОМОГОЮ ІoT ДЛЯ МОНІТОРИНГУ ЯКОСТІ ПОВІТРЯ В ПРИМІЩЕННЯХ

3.1 Вибір та аналіз програмно-апаратних засобів для розробки системи

У даному проєкті буде використовувється наступне апаратне та програмне забезпечення:

3.1.1 ESP8266

ESP8266 (рисунок 3.1) можна використовувати як зовнішній Wifi модуль, використовуючи стандартний набір прошивок AT Command Set, підключивши його до будь-якого мікроконтролера через послідовний UART, або безпосередньо як мікроконтролер з підтримкою Wifi, запрограмувавши нову прошивку за допомогою SDK, що входить до комплекту поставки.

ESP8266 Pinout

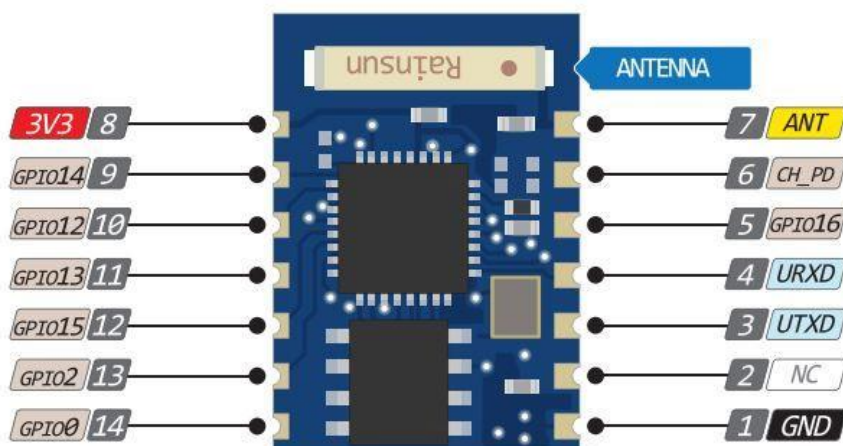


Рисунок 3.1 – Модуль WiFi ESP8266

Виводи GPIO забезпечують аналоговий і цифровий ввід/вивід, а також ШІМ, SPI, I2C тощо.

Ця плата в основному в контексті IoT, де потрібно додати підключення, до проекту Arduino.

Деякі з можливостей реалізації проектів:

- Реєстрація температури та веб-інтерфейс
- Ретро веб-браузер
- Димова сигналізація з підтримкою Інтернету
- Контроль якості повітря з підтримкою Інтернету
- Контроль CO2 з підтримкою Інтернету

Технічні характеристики

- 802.11 b / g / n
- Wi-Fi Direct (P2P), soft-AP
- Вбудований стек протоколів TCP/IP
- Вбудований TR-перемикач, LNA, підсилювач потужності та

узгоджувальна мережа

- Вбудований PLL, стабілізатор напруги та компоненти управління

живленням

- Режим 802.11b + вихідна потужність 19,5 дБм
- Вбудований датчик температури
- Підтримка різноманітності антен
- Струм витоку у вимкненому стані менше 10 мкА
- Вбудований малопотужний 32-розрядний процесор: може працювати

в якості процесора додатків

- SDIO 2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- Агрегація A-MPDU, A-MSDU та 0.4 протягом 2 мс
- 2 мс, підключення та передача пакетів даних
- енергоспоживання в режимі очікування менше 1,0 мВт (DTIM3)

Використання ESP8266 з Arduino

Для використання ESP8266, як модуля для Arduino є стандартні бібліотеки від виробника або можна портом Arduino IDE для безпосереднього

					ДП.КН.9500086.057.ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

програмування ESP. Для безпосереднього програмування ESP8266 можна встановити інструментарій та утиліту для завантаження прошивки (ESP Open SDK). З даною платою також можна взаємодіяти з ESP8266, надсилаючи AT-команди через передбачені виводи TX і RX, використовуючи інший мікроконтролер.

3.1.2 BME680 IAQ sensor

BME680 - це газовий датчик (рисунок 3.2), який поєднує в собі високолінійні та високоточні датчики газу, тиску, вологості та температури. Він спеціально розроблений для мобільних застосунків і пристроїв, де розмір і низьке енергоспоживання є критично важливими вимогами.

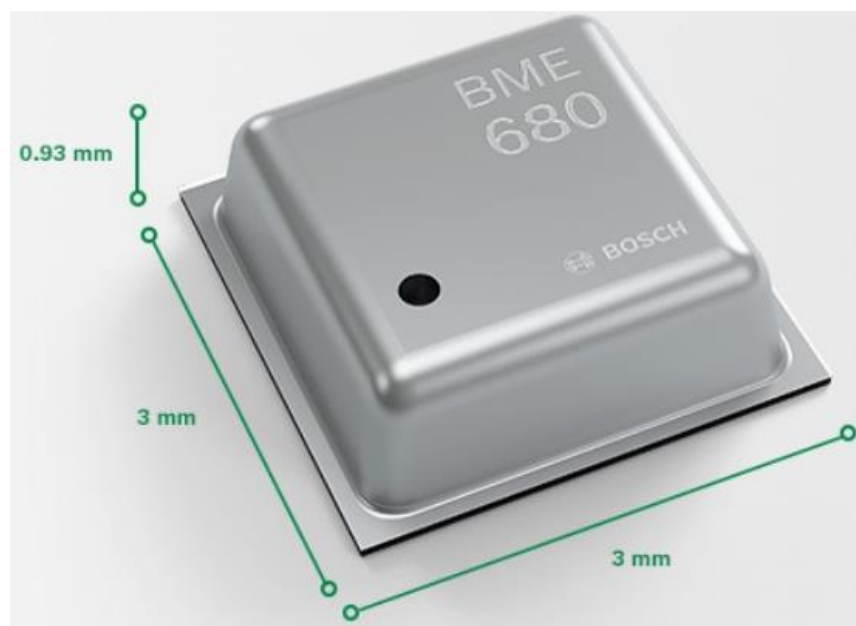


Рисунок 3.2 – Сенсор BME680 IAQ

BME680 гарантує - залежно від конкретного режиму роботи - оптимізоване споживання, довготривалу стабільність і високу стійкість до електромагнітних перешкод. Для вимірювання якості повітря для особистого благополуччя датчик газу в BME680 може виявляти широкий спектр газів, таких, як леткі органічні сполуки (ЛОС).

					ДП.КН.9500086.057.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

3.1.3 Adafruit SCD-30 - NDIR датчик температури та вологості CO₂

SCD-30 - це NDIR-датчик (рисунок 3.3), який є "справжнім" датчиком CO₂, який показує склад CO₂ PPM (частин на мільйон) у навколишньому повітрі. Цей датчик не наближено вимірює концентрацію VOC-газів - він дійсно вимірює концентрацію CO₂. Ідеально підходить для зондування навколишнього середовища, наукових експериментів, досліджень якості повітря та вентиляції тощо.

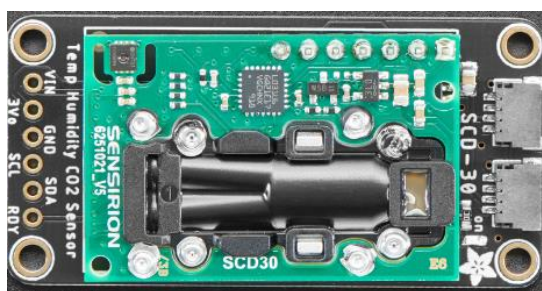


Рисунок 3.3 - SCD-30 датчик CO₂

Дані зчитуються через I2C, тому він працює практично з будь-яким мікроконтролером або мікрокомп'ютером.

3.1.3 Autodesk Tinkercad Circuit

Arduino та Tinkercad (рисунок 3.4) - два потужні інструменти, які набувають все більшої популярності у світі електроніки та інженерії.

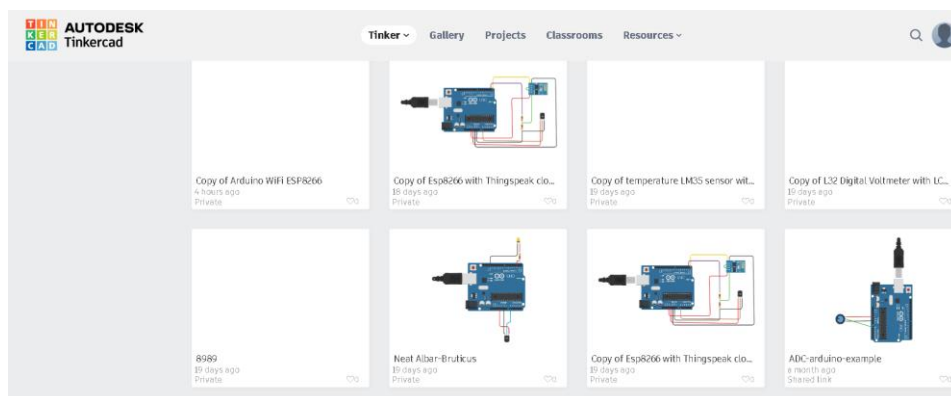


Рисунок 3.4 - Autodesk Tinkercad Circuit

					ДП.КН.9500086.057.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

Arduino - це електронна платформа з відкритим вихідним кодом, яка призначена для створення та програмування електронних пристроїв. Вона складається з програмованого мікроконтролера або комп'ютерного чіпа, який можна використовувати для керування електронними компонентами, такими як світлодіоди, двигуни, датчики тощо.

Tinkercad - це онлайн-платформа, яка дозволяє користувачам проектувати та моделювати 3D-моделі. Однією з найцікавіших особливостей Arduino та Tinkercad є те, що їх можна використовувати разом. Tinkercad має ряд електронних компонентів, які можна використовувати для створення схем і керування пристроями, а також вбудований симулятор Arduino, який дозволяє користувачам програмувати і тестувати свої схеми у віртуальному середовищі.

3.1.4 ThingSpeak

ThingSpeak - це платформа, що надає різноманітні сервіси, призначені виключно для створення IoT-додатків (рисунк 3.5).

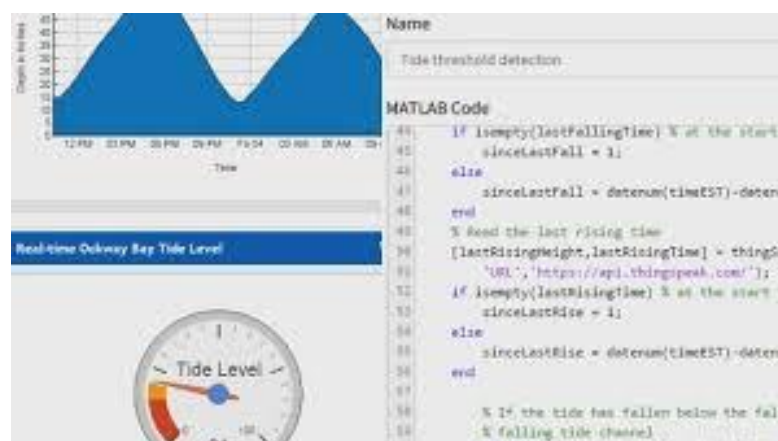


Рисунок 3.5 – Платформа ThingSpeak

Вона пропонує можливості збору даних у режимі реального часу, візуалізації зібраних даних у вигляді графіків, можливість створювати плагіни

					ДП.КН.9500086.057.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

та додатки для співпраці з веб-сервісами, соціальними мережами та іншими API. Основним елементом ThingSpeak є "канал ThingSpeak". Канал зберігає дані, які надсилаються до ThingSpeak, і складається з наведених нижче елементів:

8 полів для зберігання даних будь-якого типу - їх можна використовувати для зберігання даних з датчика або вбудованого пристрою.

3 поля розташування - можна використовувати для зберігання широти, довготи та висоти над рівнем моря. Вони дуже корисні для відстеження рухомого пристрою.

1 поле стану - коротке повідомлення для опису даних, що зберігаються в каналі.

Щоб використовувати ThingSpeak, потрібно зареєструватися і створити канал. Після створення каналу можна надсилати дані, дозволяти ThingSpeak обробляти їх, а також отримувати їх.

3.1.5 Cisco Packet Tracer

Дане середовище є потужним інструментом для побудови та імітації роботи комп'ютерних мереж (рисунок 3.6).

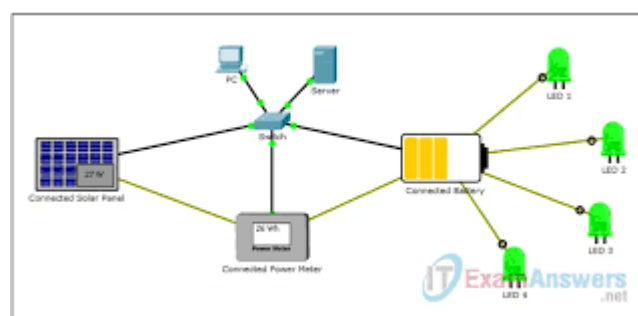


Рисунок 3.6 – Середовище Cisco Packet Tracer

На додаток до класичних мережевих пристроїв, таких як маршрутизатори і комутатори, доступних в попередніх версіях, Packet Tracer Components Box містить широкий спектр розумних речей і компонентів:

					ДП.КН.9500086.057.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

- Розумні речі - це фізичні об'єкти, які можуть підключатися до сервера реєстрації або домашнього шлюзу через мережевий інтерфейс. Вони розділені на 4 підкатегорії: Домашні, Розумне місто, Промислові та Енергосистема.

- Компоненти - це фізичні об'єкти, які підключаються до мікроконтролерів (MCU-PT) або одноплатних комп'ютерів (SBC-PT). Зазвичай вони не мають мережевого інтерфейсу і покладаються на MCU-PT або SBC-PT для доступу до мережі. Це прості пристрої, які взаємодіють лише через свої аналогові або цифрові порти.

Існує три підкатегорії для компонентів :

- Плати: мікроконтролери (MCU-PT), одноплатні комп'ютери (SBC-PT) і спеціальний пристрій під назвою Thing, які використовуються для створення автономних фізичних об'єктів, таких як кавоварки або сигналізатори диму.

- Приводи: ці компоненти маніпулюють навколишнім середовищем, самими собою або простором навколо себе.

- Датчики: ці компоненти сприймають навколишнє середовище (фотодетектори, температурні датчики), простір навколо себе (RFID, металеві датчики) або взаємодію (потенціометр, кнопка).

3.2 Програмно технологічне забезпечення системи

Беручи до уваги розділ 2 де було описано архітектуру системи та основні алгоритми функціонування, можна приступати до програмної реалізації. Як і в архітектурі ПЗ буде поділятися на 2 основні функціональні групи – клієнтська частина, функціонал якої зосереджений на зборі даних від сенсорів, попередній обробці та відправці даних на сервер, крім того дане ПЗ отримувати керуючі команди, якщо буде потреба повністю автоматизувати процес контролю якості повітря.

					ДП.КН.9500086.057.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

3.2.1 Апаратно програмне забезпечення «клієнта» системи

Будь яка система збору даних базується на двох елементах – це апаратне забезпечення (сенсори, контролери, виконавчі механізми) та програмному забезпеченню яке працює на мікроконтролері та забезпечує алгоритми функціонування.

На рисунку 3.7 представлено апаратну реалізацію передачі даних в хмарний сервіс.

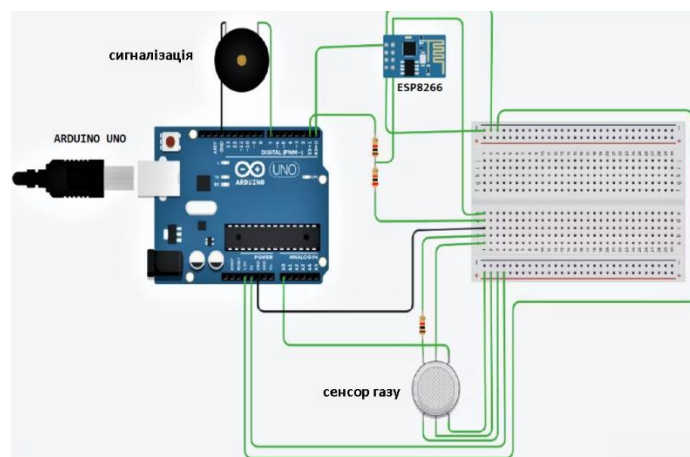


Рисунок 3.7 – Симуляція апаратної частини клієнта з газовим сенсором

На початку роботи контролер повинен налаштувати послідовний порт для можливості перегляду даних які будуть передаватись та для відлагодження програми. Крім того також налаштовується саме з'єднання через WiFi. На рисунку 3.8 представлено фрагмент коду налаштування.

```
1 void setup()  
2 {  
3   Serial.begin(115200);  
4   delay(100);  
5   Serial.println("AT+CWJAP=\"Simulator Wifi\", \"\"\\r\\n");  
6   delay(3000);  
7   pinMode(7, OUTPUT);  
8   pinMode(A0, INPUT);  
9 }  
10
```

Рисунок 3.8 – Початкові налаштування контролера

Функція void setup() відповідає за усі налаштування контролера. В ній описуються всі подальші налаштування всіх периферійних пристроїв.

Функція Serial.begin(115200) – відповідає за налаштування швидкості передачі даних через послідовний порт.

За допомогою функції pinMode(7,OUTPUT); налаштовується порт GPIO для надсилання управляючого сигналу до сигналізації.

Функція pinMode (A0, INPUT); активує роботу АЦП для отримання даних від сенсора CO2.

Далі контролер переходить в режим виконання безкінечного циклу void loop(), цей цикл виконується до тих пір поки контролер буде увімкнено.

В даному циклі покроково виконуються наступні команди (рисунок 3.9):

```
void loop()
{
    val = analogRead (A0);
    Serial.println (val);
    delay(1000); // Wait for 1000 millisecond(s)

    Serial.println("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", 80\r\n");
    delay(3000);
    int len = 60;
    Serial.print("AT+CIPSEND=");
    Serial.println(len);
    delay(1000);
    Serial.print ("GET /update?api_key=INO7SZ58XVOAZJYP&field1=");
    Serial.print (val);
    Serial.println (" HTTP/1.1\r\n");
    delay(1000);
    if (val > 80){
        tone(7, 1000);
    }
    Serial.println("AT+CIPCLOSE=0\r\n");
    delay(1000); // Thingspeak free account
}
```

Рисунок 3.9 – Основний код зчитування даних з сенсора та передачі даних на хмарний сервіс

Функція analogRead (A0) – отримує значення спаду напруги від АЦП, який потім можна конвертувати в рівень CO2 відповідно до специфікації сенсора.

Функція Serial.println виводить повідомлення в послідовний порт, в даній реалізації вона буде виводити AT команди для контролера ESP8266.

					ДП.КН.9500086.057.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

Функція `Serial.println("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", 80\r\n")` надсилає AT команду до контролера встановити TCP з'єднання з сервером `thingspeak.com` по порту 80;

Функція `Serial.print("AT+CIPSEND=")` – ініціює відправку даних до сервера.

Функція `Serial.println(len)` відправляє на контролер довжину даних яка буде відправлятися.

Функція `Serial.print("GET /update?api_key=INO7SZ58XVOAZJYP&field1=")` -створює запит GET для відправки запиту на конкретний канал куди будуть відправлятися дані.

Функції `Serial.print (val)` та `Serial.println (" HTTP/1.1\r\n")` – відповідно добавляють до запиту самі дані і версію протоколу по якому проводиться передача даних.

Функція `Serial.println("AT+CIPCLOSE=0\r\n")` – завершує з'єднання по протоколу TCP.

Після чого контролер повертається до виконання команди `analogRead (A0)`. Так буде відбуватись поки не сяде батарея або не буде вимкнено контролер.

3.2.2 Програмне забезпечення «клієнта» системи для імітування мережі сенсорів.

Так як `tinkercad` може моделювати тільки один пристрій, а загалом потрібно наприклад, отримувати дані з декількох кімнат чи великих приміщень, було вирішено окремо розробити програмну імітацію декількох таких пристроїв які працюють разом в середовищі `Cisco Packet Tracer` (рисунок 3.10).

Дане середовище може імітувати різні типи мереж і сенсорів. Тобто тут так само як і в реальних контролерах можна створювати код який потім можна переносити на реальні пристрої з мінімальними змінами.

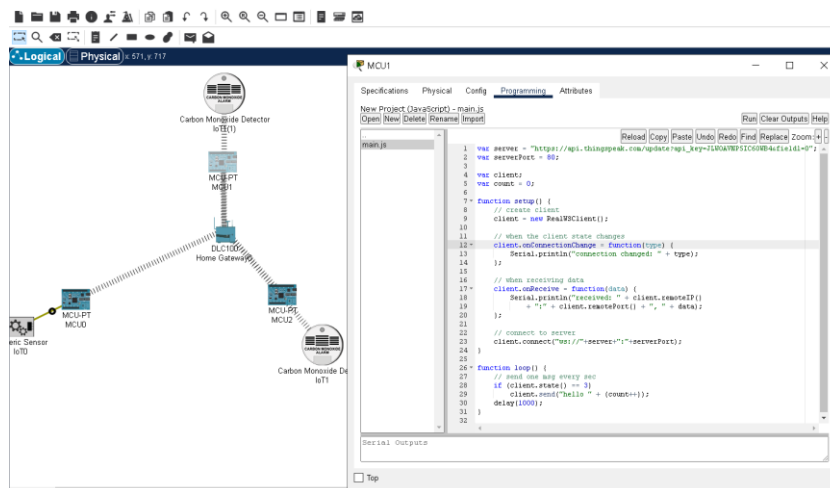


Рисунок 3.10 – Імітація мережі сенсорів CO2

Крім мережі на рисунку представлено код реалізації, так як він майже повторює функціонал.

Функція `new RealWSClient()` забезпечує спосіб обміну даними між браузером і сервером через постійне з'єднання. Дані можуть передаватися в обох напрямках у вигляді "пакетів", без розриву з'єднання і необхідності додаткових HTTP-запитів. WebSocket особливо добре підходить для сервісів, які вимагають безперервного обміну даними, наприклад, системи в режимі реального часу.

3.2.2 Програмне забезпечення «сервера» системи

В якості сервера та хмарного сервісу використано <https://thingspeak.com/> (рисунок 3.11).

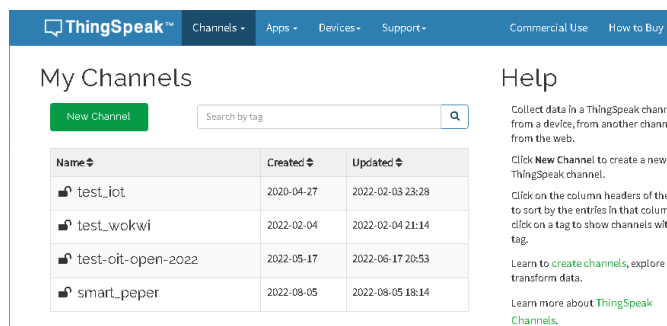


Рисунок 3.11 – Головне вікно сервісу

Сервіс підтримує ідеологію каналів для збору інформації. Кожен канал складається з 8 полів куди можна записувати як числові дані так і дані типу стрічка (рисунок 3.12).

Channel Settings

Percentage complete: 30%

Channel ID: 1737956

Name: test-oit-open-2022

Description:

Field 1: voise ☒

Field 2: react ☒

Field 3: Field Label 3 ☒

Field 4: Field Label 4 ☒

Field 5: Field Label 5 ☒

Field 6: Field Label 6 ☒

Field 7: Field Label 7 ☒

Field 8: Field Label 8 ☒

Help

Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

Channel Settings

- Percentage complete:** Calculated based on data entered into the various fields of a channel. Enter the name, description, location, URL, video, and tags to complete your channel.
- Channel Name:** Enter a unique name for the ThingSpeak channel.
- Description:** Enter a description of the ThingSpeak channel.
- Field#:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- Tags:** Enter keywords that identify the channel. Separate tags with commas.
- Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- Show Channel Location:**
 - Latitude:** Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072.
 - Longitude:** Specify the longitude position in decimal degrees. For example, the longitude of the city of London is -0.1275.

Рисунок 3.12 – Налаштування каналу

Для реалізації даного проекту не потрібно створювати декілька для різних пристроїв, як пропонується на рисунку 3.10, достатньо буде одного каналу та 3 полів, а кожне поле буде ідентифікатором для пристрою (рисунок 3.13).

```
Перший пристрій GET
https://api.thingspeak.com/update?api_key=JLW0AVNP5IC6OWB4&field1=0
Другий пристрій GET
https://api.thingspeak.com/update?api_key=JLW0AVNP5IC6OWB4&field2=0
Третій пристрій GET
https://api.thingspeak.com/update?api_key=JLW0AVNP5IC6OWB4&field3=0
```

Рисунок 3.13 – REST API для відправки даних на канал

Для активації дій (тригерів) можна використовувати застосунок react, який приєднується до певного каналу і поля та аналізувати дані, після чого може запускати скрипт аналізу та отримувати результати (рисунки 3.14).

The screenshot shows the 'React' configuration page in the ThingSpeak web interface. The top navigation bar includes 'Channels', 'Apps', 'Devices', 'Support', 'Commercial Use', and 'How to Buy'. The main content area is titled 'React Settings' and includes a 'Help' link. The configuration fields are as follows:

- React Name:** React 8
- Condition Type:** Numeric
- Test Frequency:** On Data Insertion
- Condition:** If channel: test-oit-open-2022 (1737956)
- field:** 3 (Field Label 3)
- Comparison:** is greater than
- Action:** MATLAB Analysis
- Code to execute:** Predict
- Options:** Run action each time condition is met (selected)

A green 'Save React' button is located at the bottom of the configuration area.

Рисунок 3.14 – Створення реакції

По суті реакція це є налаштування спеціального скрипта, який буде запускатись відносно отриманих даних та активувати інший скрипт аналізу та прогнозування. В реакції є наступні поля налаштувань:

React Name – назва ідентифікатора реакції для звернення через API сервісу;

Condition Type – тип значень, може бути числовий, стрічковий, геолокація,

Test Frequency – частота спрацьовування, коли дані надійшли, чи по таймеру;

Condition If channel – вибір ідентифікатора (назви каналу)

Field – номер поля з якого будуть отримуватись дані

Action – тип скрипта який буде використовуватись відносно умови, може бути RESTAPI або Matlab скрипт.

На рисунку 3.15 представлено скрипт прогнозування на основі алгоритму LSTM. Він запускається скриптом реакції який був описаний вище.

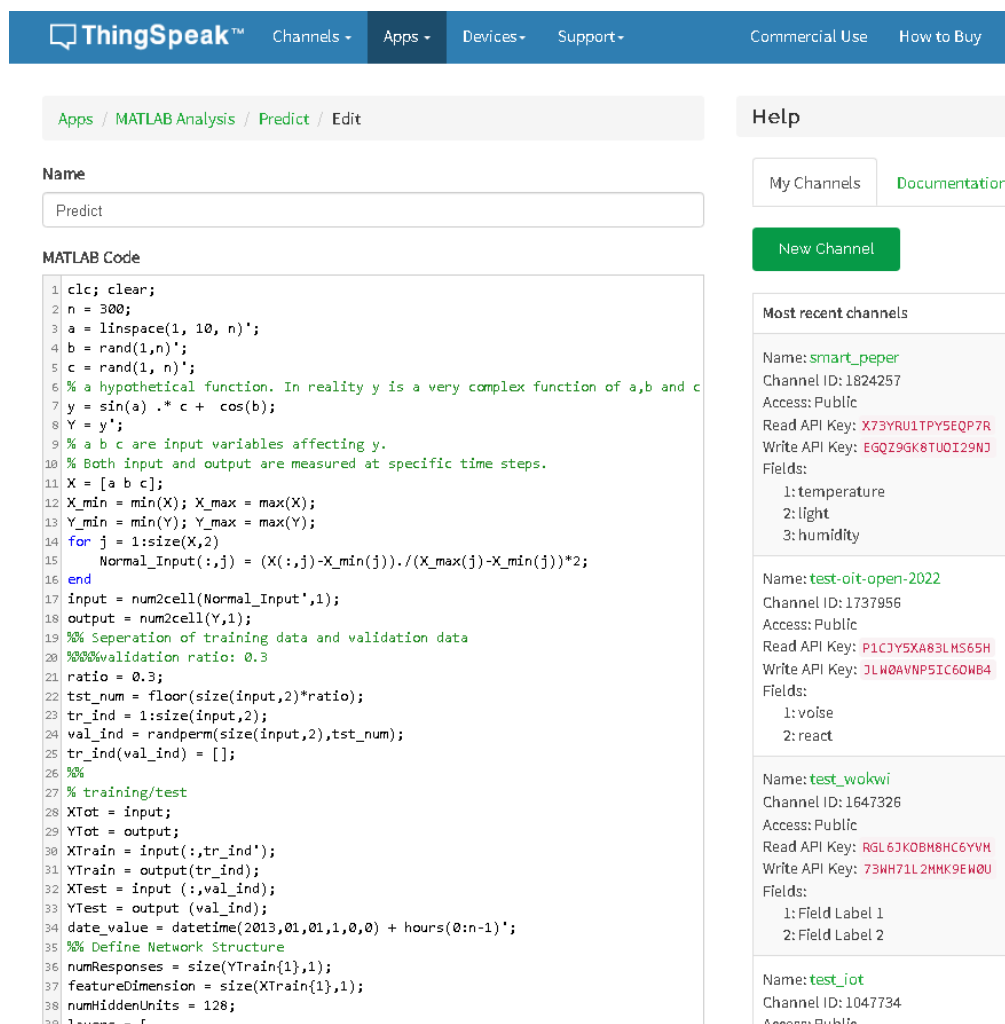


Рисунок 3.15 – Алгоритм LSTM

3.3 Тестування системи прогнозування концентрації CO2

Для тестування взаємодії клієнтської і серверної частини було вибрано також пакет Postman (рисунок 3.16).

Postman - це платформа для створення та використання API, який спрощує кожен крок життєвого циклу API та оптимізує співпрацю, для пришвидшення створення API.

					ДП.КН.9500086.057.ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

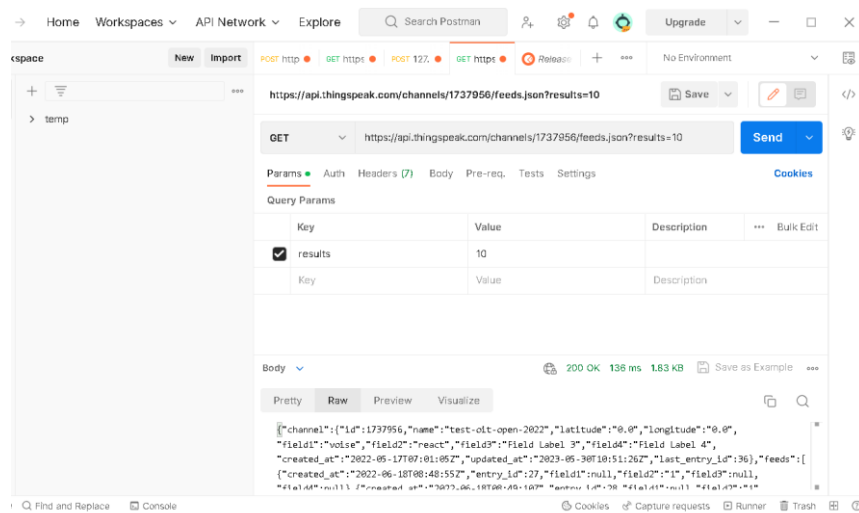


Рисунок 3.16 – Програма тестування Postman

Для тестування отримання даних від клієнта до сервера було протестовано наступний GET запит:

`https://api.thingspeak.com/update?api_key=JLW0AVNP5IC6OWB4&field1=1500.`

Даний запит складається з наступних частин:

- `https://api.thingspeak.com/update?` – адреса завантаження даних
- `api_key=JLW0AVNP5IC6OWB4` – унікальний ідентифікатор
- `field1=1500` – ім'я поля і дані

Якщо дані отримані, сервер відправить відповідь наступного вигляду.

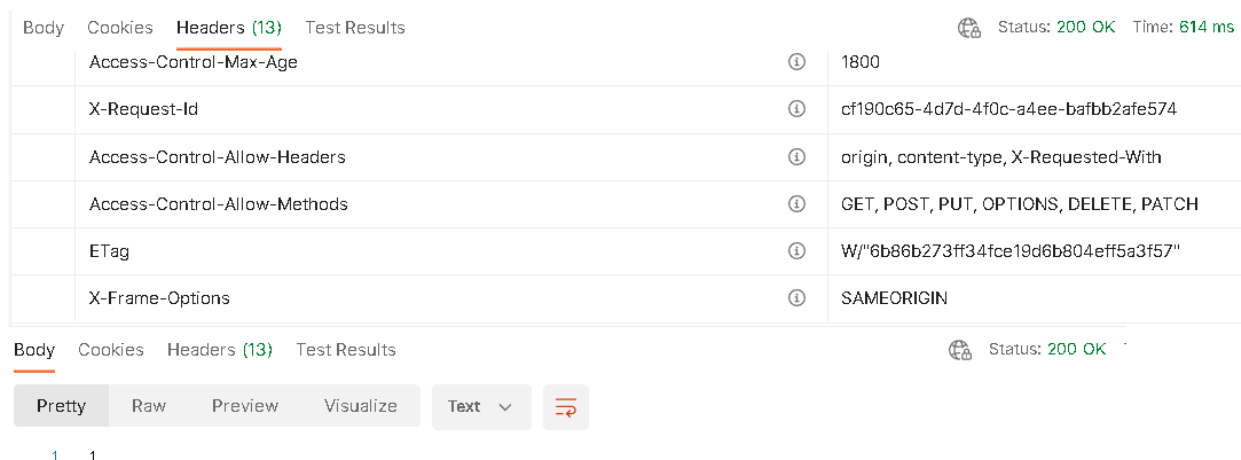


Рисунок 3.17 – Успішна відповідь сервера

					ДП.КН.9500086.057.ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

Сервер повертає статус 200 OK, що означає що запит прийнято та у тілі відповіді повертає значення. Якщо значення рівне нулю – дані не збережені в канал, якщо значення більше нуля – дані записані в канал.

На сторінці самого каналу в вкладці візуалізація можна побачити візуалізацію цих даних.

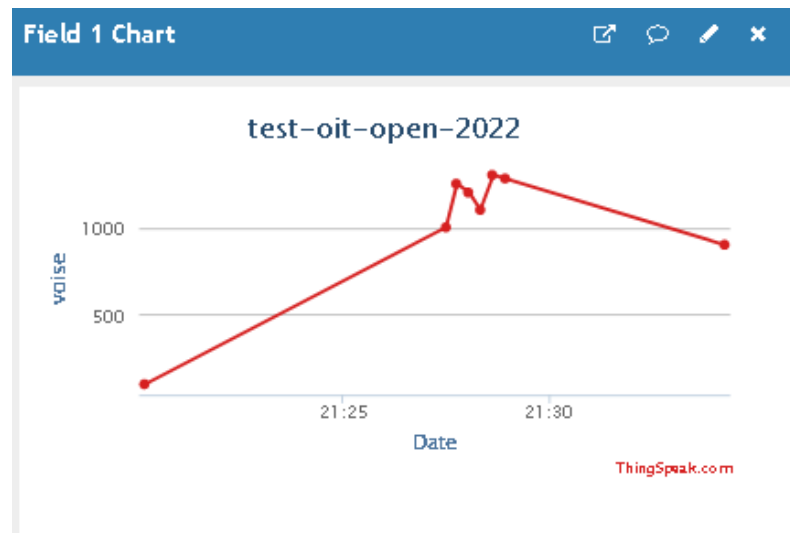


Рисунок 3.18 – Результат візуалізації

Після тестування сторони сервера, можна перейти до тестування клієнтської частини. В реалізації даного проекту контролер формує такий самий запит, як в Postman, що був показаний вище.

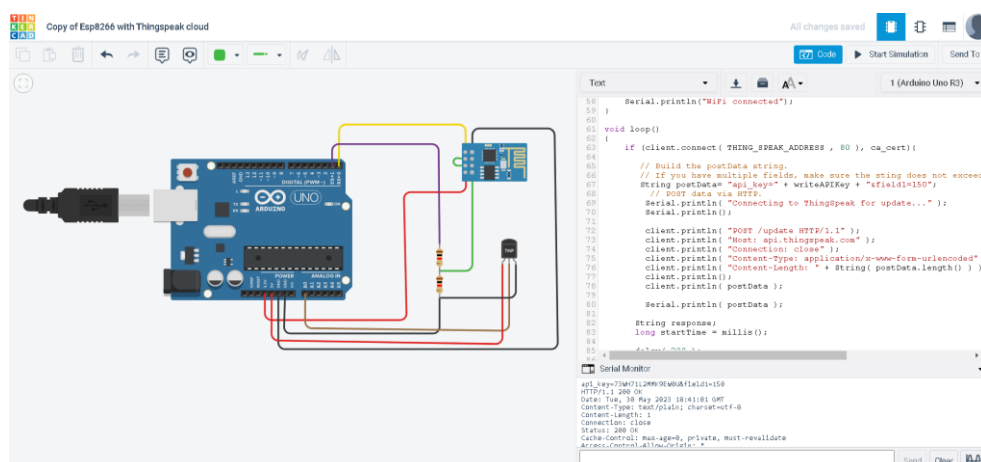


Рисунок 3.19 – Результат тестування клієнта

					ДП.КН.9500086.057.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

Після отриманні даних на сервері також спрацьовує скрипт прогнозу на базі алгоритму LSTM. В експерименті було отримано 30 точок та прогнозовано концентрацію CO₂, як показано на рисунку 3.20.

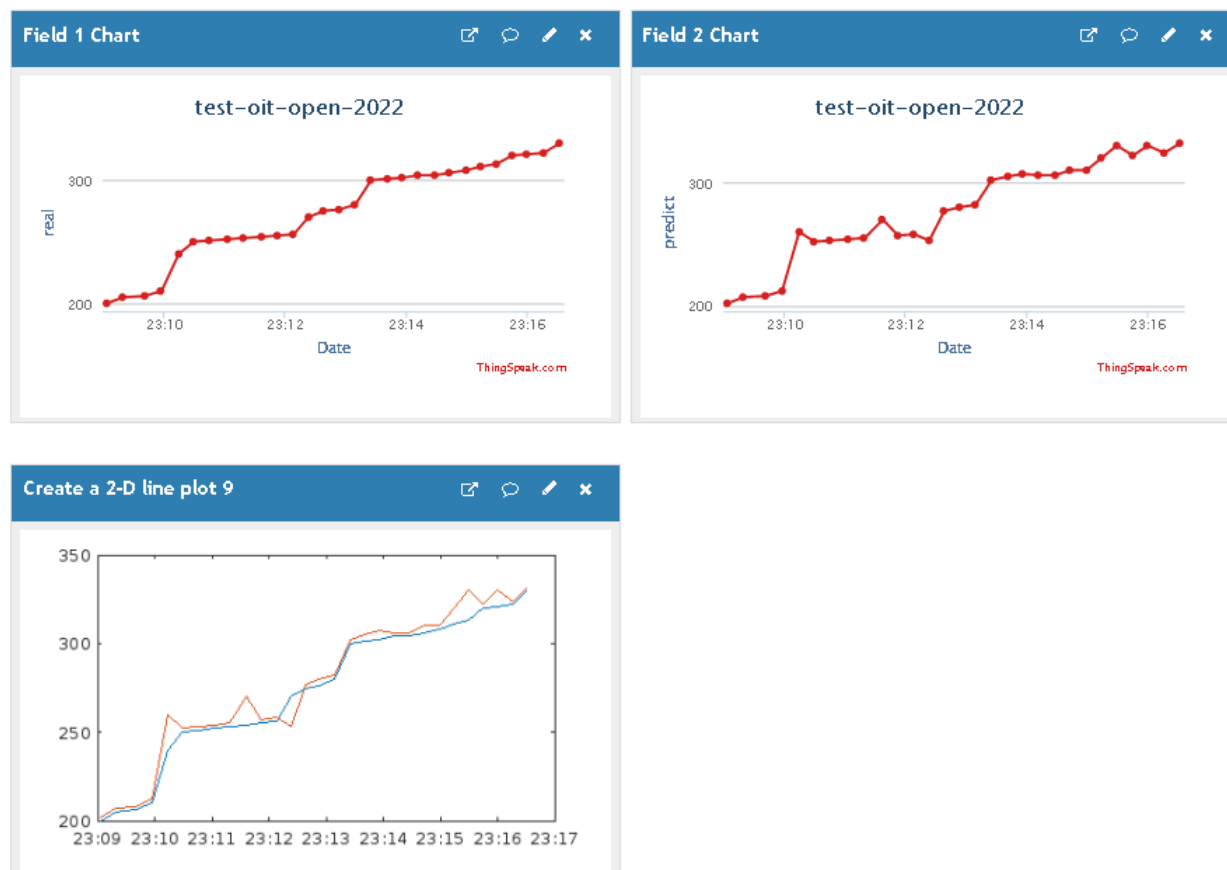


Рисунок 3.20 – Результати прогнозу та реальних даних

ВИСНОВКИ

У результаті виконання дипломного проекту було проведено аналіз сучасних напрямків забезпечення комфорту та управління якістю навколишнього середовища, а саме контролю концентрації вуглекислого газу.

Як показали дослідження при пандеміях рівень CO₂ відіграє дуже важливу роль в розповсюдженні захворювання. Крім того правильний мікроклімат та його прогнозування сприяє роботі та самопочуттю людей.

В роботі проаналізовано можливості інтегрування Інтернету речей в системах контролю якості повітря, що показало, що системи контролю, які були дорогі і не доступні для широкого загалу, тепер з розвитком концепції IoT, такі системи починають з'являтися практично у кожному домі.

Крім переваг IoT, яка може застосовуватись в системах контролю якістю повітря було проведено аналіз як дорогих систем так і дешевих, які збираються дослідниками, або аматорами.

В роботі було проаналізовано апаратне та програмне забезпечення таких систем, так як від архітектури та набору контролерів і сенсорів дуже суттєво буде залежати, як складність системи так і її точність.

Ще одним додатком до таких систем є можливість прогнозування концентрації CO₂ для вчасної реакції на ситуацію та попередження збільшення рівня CO₂.

В роботі описано основні переваги та недоліки IoT-орієнтованих систем контролю викидів CO₂ в реальному часі і запропоновано недорогу систему орієнтовану на концепцію IoT з можливістю прогнозу концентрації CO₂ та правильної реакції на цю концентрацію.

На основі проведеного аналізу та визначення вимог до системи було розроблено:

- архітектуру системи прогнозування концентрації CO₂ за допомогою IoT,
- алгоритмічне забезпечення системи,

					ДП.КН.9500086.057.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

- алгоритм читання даних з сенсора та відправки їх на сервіс,
- алгоритм роботи серверної частини
- адаптовано модель прогнозування на основі LSTM
- підібрано програмно-апаратні засоби для розробки системи
- розроблено ПЗ для модуля визначення концентрації CO2
- розроблено ПЗ та модель мережі розумних сенсорів CO2
- створено канал для збору даних від розумних сенсорів
- налаштовано логіку спрацьовування реакцій на каналі thinkspeak
- інтегровано скрипт прогнозування в канал thinkspeak
- протестовано роботу системи в цілому

Практичним результатом дипломного проекту є: створення системи прогнозування концентрації CO2 за допомогою IoT для моніторингу якості повітря в приміщеннях. Система отримує дані від мережі розумних сенсорів та зберігає їх одночасно система перевіряє якість повітря і прогнозує наступні значення, тим самим попереджаючи виникнення небезпечних ситуацій.

Однак було виявлено деякі недоліки системи під час тестування, а саме: виявилось, що є надлишковість даних які відправляються на сервер, для вирішення даної проблеми можна використати IoT шлюз для попередньої обробки та відправки тільки важливих даних. Ще одною проблемою стало те, що потрібно враховувати додаткові параметри якості повітря – запиленість, вологість, температура для адекватної реакції системи.

					ДП.КН.9500086.057.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Klepeis, N.E.; Nelson, W.C.; Ott, W.R.; Robinson, J.P.; Tsang, A.M.; Switzer, P.; Behar, J.V.; Hern, S.C.; Engelmann, W.H. The National Human Activity Pattern Survey (NHAPS): A resource for assessing exposure to environmental pollutants. *J. Expo. Sci. Environ. Epidemiol.* 2001, 11, 231–252.
2. Mumtaz, R.; Zaidi, S.M.H.; Shakir, M.Z.; Shafi, U.; Malik, M.M.; Haque, A.; Mumtaz, S.; Zaidi, S.A.R. Internet of things (IoT) based indoor air quality sensing and predictive analytic—A COVID-19 perspective. *Electronics* 2021, 10, 184
3. He, J.; Xu, L.; Wang, P.; Wang, Q. A high precise E-nose for daily indoor air quality monitoring in living environment. *Integration* 2017, 58, 286–294.
4. El-Sharkawy, M.F.; Noweir, M.E. Indoor air quality levels in a University Hospital in the Eastern Province of Saudi Arabia. *J. Fam. Community Med.* 2014, 21, 39.
5. Maduranga, M.; Kosgahakumbura, K.; Karunarathna, G. Design of an IoT based indoor air quality monitoring system. In *Proceedings of the National Engineering Research Symposium (NERS 2020)*, 2020; Volume 1, pp. 117–122. Available online: <http://ir.kdu.ac.lk/handle/345/4437> (accessed on 21 December 2022).
6. Satish, U.; Mendell, M.J.; Shekhar, K.; Hotchi, T.; Sullivan, D.; Streufert, S.; Fisk, W.J. Is CO₂ an indoor pollutant? Direct effects of low-to-moderate CO₂ concentrations on human decision-making performance. *Environ. Health Perspect.* 2012, 120, 1671–1677.
7. Allen, J.G.; MacNaughton, P.; Satish, U.; Santanam, S.; Vallarino, J.; Spengler, J.D. Associations of cognitive function scores with carbon dioxide, ventilation, and volatile organic compound exposures in office workers: A controlled exposure study of green and conventional office environments. *Environ. Health Perspect.* 2016, 124, 805–812.

					<i>ДП.КН.9500086.057.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

8. Persily, A. Challenges in developing ventilation and indoor air quality standards: The story of ASHRAE Standard 62. Build. Environ. 2015, 91, 61–69.

9. Peng, Z.; Jimenez, J.L. Exhaled CO₂ as a COVID-19 infection risk proxy for different indoor environments and activities. Environ. Sci. Technol. Lett. 2021, 8, 392–397.

10. Rudnick, S.; Milton, D. Risk of indoor airborne infection transmission estimated from carbon dioxide concentration. Indoor Air 2003, 13, 237–245.

11. Franco, A.; Leccese, F. Measurement of CO₂ concentration for occupancy estimation in educational buildings with energy efficiency purposes. J. Build. Eng. 2020, 32, 101714.

12. Wei, Y.; Wang, S.; Jin, L.; Xu, Y.; Ding, T. Indoor occupancy estimation from carbon dioxide concentration using parameter estimation algorithms. Build. Serv. Eng. Res. Technol. 2022, 43, 01436244211060903.

13. Jiang, C.; Masood, M.K.; Soh, Y.C.; Li, H. Indoor occupancy estimation from carbon dioxide concentration. Energy Build. 2016, 131, 132–141.

14. Pedersen, T.H.; Nielsen, K.U.; Petersen, S. Method for room occupancy detection based on trajectory of indoor climate sensor data. Build. Environ. 2017, 115, 147–156.

15. Erickson, V.; Cerpa, O. Occupancy-based system for efficient reduction of HVAC energy. In Proceedings of the 10th International Conference on Information Processing in Sensor Networks (IPSN'11), Stockholm, Sweden, 12–16 April 2010.

16. Persily, A.K. Indoor carbon dioxide concentrations in ventilation and indoor air quality standards. In Proceedings of the 36th AIVC Conference Effective Ventilation in High Performance Buildings, Madrid, Spain, 23–24 September 2015; pp. 810–819.

17. EN 13779; Ventilation for Non-Residential buildings—Performance Requirements for Ventilation and Room-Conditioning Systems. European Committee for Standardization (CEN): London, UK, 2007.

					ДП.КН.9500086.057.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

18. H. Mirinejad, K. C. Welch and L. Spicer, "A review of intelligent control techniques in HVAC systems," 2012 IEEE Energytech, Cleveland, OH, USA, 2012, pp. 1-5, doi: 10.1109/EnergyTech.2012.6304679.

19. Congradac, V.; Kulic, F. HVAC system optimization with CO2 concentration control using genetic algorithms. Energy Build. 2009, 41, 571–577.

20. Tekler, Z.D.; Chong, A. Occupancy prediction using deep learning approaches across multiple space types: A minimum sensing strategy. Build. Environ. 2022, 226, 109689.

21. Jiang, Z., Risbeck, M. J., Ramamurti, V., Murugesan, S., Amores, J., Zhang, C., ... & Drees, K. H. (2021). Building HVAC control with reinforcement learning for reduction of energy cost and demand charge. Energy and Buildings, 239, 110833.

22. Tekler, Z.D.; Ono, E.; Peng, Y.; Zhan, S.; Lasternas, B.; Chong, A. ROBOD, room-level occupancy and building operation dataset. In Building Simulation; Springer: Berlin/Heidelberg, Germany, 2022; Volume 15, pp. 2127–2137.

23. Pastor-Fernández, A.; Cerezo-Narváez, A.; Montero-Gutiérrez, P.; Ballesteros-Pérez, P.; Otero-Mateo, M. Use of Low-Cost Devices for the Control and Monitoring of CO2 Concentration in Existing Buildings after the COVID Era.

24. Tagliabue, L.C.; Cecconi, F.R.; Rinaldi, S.; Ciribini, A.L.C. Data driven indoor air quality prediction in educational facilities based on IoT network. Energy Build. 2021, 236, 110782.

25. Wei, W., Ramalho, O., Malingre, L., Sivanantham, S., Little, J. C., & Mandin, C. (2019). Machine learning and statistical models for predicting indoor air quality. Indoor Air, 29(5), 704-726.

26. Ahn, J.; Shin, D.; Kim, K.; Yang, J. Indoor air quality analysis using deep learning with sensor data. Sensors 2017, 17, 2476.

27. Tagliabue, L.C.; Cecconi, F.R.; Rinaldi, S.; Ciribini, A.L.C. Data driven indoor air quality prediction in educational facilities based on IoT network

					ДП.КН.9500086.057.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

28. Kallio, J.; Tervonen, J.; Räsänen, P.; Mäkynen, R.; Koivusaari, J.; Peltola, J. Forecasting office indoor CO2 concentration using machine learning with a one-year dataset. Build. Environ. 2021, 187, 107409.

29. Segala, G.; Doriguzzi-Corin, R.; Peroni, C.; Gazzini, T.; Siracusa, D. A practical and adaptive approach to predicting indoor co2. Appl. Sci. 2021, 11, 10771.

30. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. – Тернопіль: ЗУНУ, 2021. – 56 с.

					ДП.КН.9500086.057.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

Додаток А

Код програми реалізації алгоритму LSTM

```

clc; clear;
n = 300;
a = linspace(1, 10, n)';
b = rand(1,n)';
c = rand(1, n)';
y = sin(a) .* c + cos(b);
Y = y';
X = [a b c];
X_min = min(X); X_max = max(X);
Y_min = min(Y); Y_max = max(Y);
for j = 1:size(X,2)
    Normal_Input(:,j) = (X(:,j)-X_min(j))./(X_max(j)-X_min(j))*2;
end
input = num2cell(Normal_Input',1);
output = num2cell(Y,1);
ratio = 0.3;
tst_num = floor(size(input,2)*ratio);
tr_ind = 1:size(input,2);
val_ind = randperm(size(input,2),tst_num);
tr_ind(val_ind) = [];
XTot = input;
YTot = output;
XTrain = input(:,tr_ind');
YTrain = output(tr_ind);
XTest = input(:,val_ind);
YTest = output(val_ind);
date_value = datetime(2013,01,01,1,0,0) + hours(0:n-1)';
%% Define Network Structure
numResponses = size(YTrain{1},1);
featureDimension = size(XTrain{1},1);
numHiddenUnits = 128;
layers = [ ...
    sequenceInputLayer(featureDimension)
    lstmLayer(numHiddenUnits,'OutputMode','sequence')
    fullyConnectedLayer(featureDimension)
    dropoutLayer(0.3)
    fullyConnectedLayer(numResponses)
    regressionLayer];
maxepochs = 300;
options = trainingOptions('adam', ... %%adam
    'MaxEpochs',maxepochs, ...
    'MiniBatchSize',4, ...
    'GradientThreshold',1, ...
    'InitialLearnRate',1e-2, ...
    'LearnRateSchedule','piecewise', ...
    'LearnRateDropPeriod',125, ...
    'LearnRateDropFactor',1, ...
    'Verbose',0, ...
    'Plots','training-progress');
net = trainNetwork(XTrain,YTrain,layers,options);
YPred = predict(net,XTrain);
YPred_n = cell2mat(YPred(:));
% test

```

					ДП.КН.9500086.057.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

```

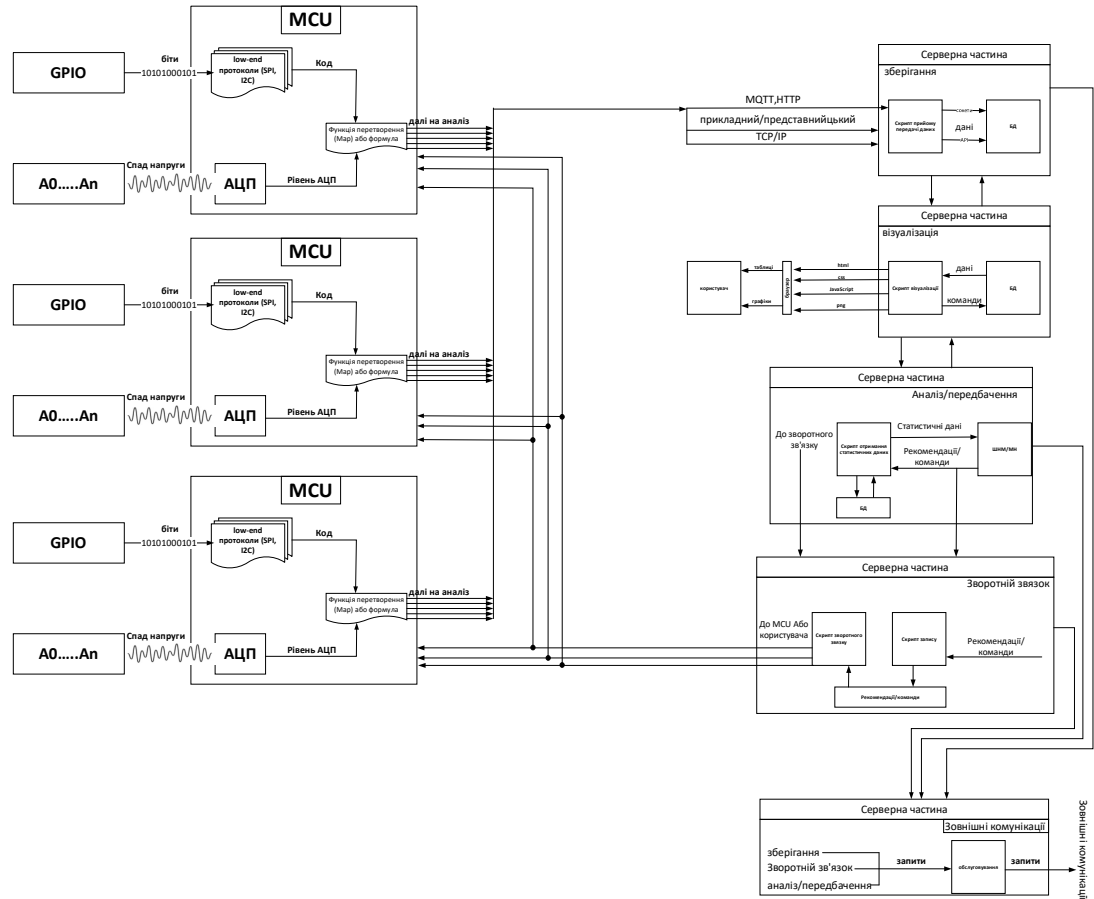
net = resetState(net);
YPred_t = predict(net,XTest);
YPred_tn = cell2mat(YPred_t(:));
%% Plotting
obs = Y';
sim = YPred_n;
sim_t = YPred_tn;
figure
plot(date_value,obs,'k-o','LineWidth',1)
hold on
plot(date_value(tr_ind),sim,'ro','LineWidth',0.9)
% plot(date_value(1:length(sim)),sim,'r--','LineWidth',0.9)
hold on
plot(date_value(val_ind),sim_t,'bo','LineWidth',0.9)
legend(['Observation' "Training" "Test"],'FontName','Times New
Roman','FontSize',12)
ylabel('ARGs(copies/mL)','FontName','Times New Roman','FontSize',14)
datetick('x','yyyy-mm-dd','keeplimits','keepticks')
title("Prediction of ARGs",'FontName','Times New Roman','FontSize',16)
hold off

```

					ДП.КН.9500086.057.ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток Б

Архітектура системи прогнозування концентрації CO₂ за допомогою IoT для моніторингу якості повітря в приміщеннях



Змн.	Арк.	№ докум.	Підпис	Дата

ДП.КН.9500086.057.ПЗ

Арк.

56

Додаток В

Алгоритм роботи програми контролера для обробки даних сенсора CO2

