

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Системотехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Розроблення системи моніторингу для зарядної системи електромобілів з застосуванням методів машинного навчання
(тема)

Виконав:
студентка 2 курсу, групи СПРМ-22-1
Єрошенко С.О.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
—
(код і повна назва спеціальності)

Тип програми освітньо-наукова
Освітня програма Системне проектування
—
(повна назва освітньої програми)

Керівник д.т.н., проф. Мінухін С.В
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ проф. Гребеннік І.В
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет _____ *Комп'ютерних наук* _____

Кафедра _____ *Системотехніки* _____

Рівень вищої освіти _____ *другий(магістерський)* _____

Спеціальність _____ *122 Комп'ютерні науки* _____
(код і повна назва)

Тип програми _____ *освітньо-наукова* _____

Освітня програма _____ *Комп'ютерні науки* _____

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ *Єрошенко Софії Олександрівні* _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ *Розроблення системи моніторингу для зарядної системи електромобілів з застосуванням методів машинного навчання* _____

затверджена наказом університету від _____ *1 квітня 2024* р. № _____ *259Ст*

2. Термін подання студентом роботи до екзаменаційної комісії _____ *15 06 2024* р.

3. Вихідні дані до роботи _____ *розробити системи моніторингу для зарядної системи електромобілів з застосуванням методів машинного навчання* _____

4. Перелік питань, що потрібно опрацювати в роботі _____ *Вступ. Аналіз існуючих методів до вирішення ICEing. Збір та обробка даних системи в різних умовах паркування. Розробка та інтеграція моделі машинного навчання у мікроконтролер. Висновки.* _____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) _____ *схема тестового стенду, сирий сигнал з ультразвукового сенсору та сигнал магнітометра, відфільтрований сигнал за допомогою Фільтра Савіцького-Голея, алгоритм маркування даних, сигнал з ультразвукового сенсору для першого і другого паркувального місця, першого магнітометра та камери, сигнал з*

ультразвукового сенсору для третього і четвертого паркувального місця, другого магнітометра та камери, матриця невідповідностей Random Forest для лівого паркувального місця, матриця невідповідностей Random Forest для правого паркувального місця, схема використання моделі у реальному середовищі, схема зв'язку моделі у STM магнітометрів

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

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
	Проф. Мінухін С.В.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	<i>Отримання завдання на виконання роботи</i>	<i>01.03.2024</i>	
2	<i>Аналіз існуючих методів до вирішення ICEing</i>	<i>02-15.03.2024</i>	
3	<i>Збір та обробка даних системи в різних умовах паркування</i>	<i>16.03-09.04.2024</i>	
4	<i>Розробка та інтеграція моделі машинного навчання у мікроконтролер</i>	<i>10.04-17.05.2024</i>	
5	<i>Оформлення пояснювальної записки</i>	<i>18.05-01.06.2024</i>	
6	<i>Представлення на рецензування</i>	<i>12.06.2024</i>	

Дата видачі завдання 1 березня 2024 р.

Студент _____

 (підпис)

Єрошенко С.О.
 (прізвище, ініціали)

Керівник роботи _____
 (підпис)

д.т.н., проф. Мінухін С.В.
 (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до магістерської кваліфікаційної роботи: с.82, 30 рис., 10 табл., 54 джерела інформації., 3 додатки.

МОНІТОРИНГ, МАШИННЕ НАВЧАННЯ, МАГНІТОМЕТР, ФІЛЬТР САВІЦЬКОГО-ГОЛЕЯ, ICEing, RANDOM FOREST

Об'єктом дослідження є процес моніторингу електричних станцій для заряджання електричних транспортних засобів (EVCI), яка включає в себе всі компоненти цієї системи.

Предметом дослідження є модель машинного навчання для вирішення завдання заборони доступу (ICEing), яке полягає у паркуванні транспортних засобів з внутрішнім горінням (ICE) на місцях для заряджання електричних транспортних засобів (EV).

Метою дослідження є розробка системи моніторингу електричних станцій для заряджання електричних транспортних засобів для запобігання заборони доступу (ICEing) на базі магнітометрів та моделей машинного навчання.

Методи дослідження: системний аналіз – робота з науково-технічною літературою; моделювання – процеси обробки даних; експериментальна оцінка – оцінювання ефективності рішення, метрики для оцінки класифікаційної продуктивності; статистичний аналіз отриманих результатів.

Наукова новизна роботи полягає у вирішенні проблеми заборони доступу, визначення її причин та на пошук інноваційних методів та технологій для боротьби з нею, зокрема, використання магнітометрів для ідентифікації зайнятості паркових місць та застосування моделі машинного навчання для управління цим процесом та перевірка інтеграції моделі з STM32 мікроконтролером.

Галузь застосування – електричні станції для заряджання електричних транспортних засобів.

ABSTRACT

Master`s Thesis: 82 pages, 10 tables, 30 figures, 54 sources of information, 3 appendix.

MONITORING, MACHINE LEARNING, MAGNETOMETER, SAVITSKY-GOLAY FILTER, ICEING, RANDOM FOREST

The object of study is the EVCI monitoring system (Electric Vehicle Charging Infrastructure), which encompasses all components of this system.

The subject of the study is a machine learning model for resolving ICEing, which is the parking of internal combustion vehicles (ICE) at electric vehicle (EV) charging points.

The purpose of the study is to develop a system for monitoring electrical stations for charging electrical vehicles to prevent the ICEing using magnetometers and machine training models.

Research methods: System analysis – review of scientific and technical literature; Modeling – data processing techniques; Experimental evaluation – assessing the effectiveness of the solution, metrics for classification performance evaluation; Statistical analysis of the results obtained.

The scientific novelty of the work lies in investigating the ICEing issue, identifying its causes, and searching for innovative methods and technologies to combat it, including the use of magnetometers to identify parking spot occupancy and the application of machine learning models for managing this process and verifying the integration of the model with an STM32.

The field of application is electric vehicle charging stations.

ЗМІСТ

ВСТУП.....	9
1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ДО ВИРІШЕННЯ ЗАБЛОКУВАННЯ ЗАРЯДНИХ СТАНЦІЙ ДЛЯ ЕЛЕКТРОМОБІЛІВ.....	12
1.1 Огляд існуючих підходів до вирішення заблокування зарядних станцій	12
1.2 Огляд сучасних датчиків для виявлення руху	16
1.3 Порівняльний аналіз існуючих систем для вирішення ICEing	20
1.4 Огляд методів обробки сигналів	27
1.5 Огляд методів машинного навчання.....	28
1.6 Класифікаційні показники	33
1.7. Висновки до розділу 1	34
2 ТЕХНОЛОГІЯ ТА МЕТОДИ ОБРОБЛЕННЯ ДАНИХ ЩОДО ПАРКУВАННЯ ТРАНСПОРТНИХ ЗАСОБІВ	36
2.1 Постановка задачі та визначення вимог.....	36
2.2 Встановлення тестового стенду та збір даних.....	37
2.3 Апаратне забезпечення	40
2.4 Створення та управління базою даних	42
2.5 Обробка та аналіз сигналів	45
2.6 Маркування та підготовка даних для моделювання.....	47
2.7 Висновки до розділу 2.....	51
3 ЗАСТОСУВАННЯ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ ДЛЯ ІНТЕГРАЦІЇ З МІКРОКОНТРОЛЕРОМ У ЗАРЯДНІЙ СТАНЦІЇ ДЛЯ ЕЛЕКТРОМОБІЛІВ	52
3.1 Метод підготовки даних та застосування алгоритмів машинного навчання	52
3.2 Тренування та валідація моделі Random Forest.....	53
3.3 Тренування та валідація моделі SVC.....	57
3.4 Тренування та валідація моделі MLP	61
3.5 Тренування та валідація моделі CNN.....	63
3.6 Інтеграція моделі машинного навчання у мікроконтролер.....	66
3.7 Перспективи та рекомендації для майбутніх досліджень	70
3.8 Висновки до розділу 3.....	71
ВИСНОВКИ.....	74
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	76

ДОДАТОК А ГРАФІЧНИЙ МАТЕРІАЛ КВАЛІФІКАЦІЙНОЇ РОБОТИ.....	7 82
ДОДАТОК Б ТЕКСТ ПРОГРАМИ.....	95
ДОДАТОК В ВІДОМІСТЬ КВАЛІФІКАЦІЙНОЇ РОБОТИ.....	112

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД – база даних;

EVCI – електричних станцій для заряджання електричних транспортних засобів;

EV – електромобіль;

ICEing – проблема заборони доступу;

STM32 - сімейство 32-бітових мікроконтролерів виробництва STMicroelectronics;

CW - радар з постійною хвилею;

Random Forest - випадковий ліс;

SVC - класифікатор на основі опорних векторів;

MLP - багатошаровий перцептрон;

CNN - згорткова нейронна мережа;

SPS - система розумного паркування;

LED - Світлодіодні індикатори.

ВСТУП

Останні роки відзначилися значним зростанням популярності EV, завдяки їхній ефективності, екологічним перевагам та простоті обслуговування. Зі зростанням попиту на EV виникає критична потреба розширення інфраструктури зарядних станцій, спеціально призначених для електромобілів. Незважаючи на ці досягнення, постійною проблемою є несанкціоноване зайняття призначених місць для зарядки транспортними засобами, які не потребують зарядження, що призводить до неможливості заряджати EV і можливих втрат доходів для операторів зарядних станцій.

Проблема ICEing, або проблема паркування транспортних засобів з внутрішнього згорання на місцях для заряджання електромобілів, виникає, коли неелектричні транспортні засоби займають визначені місця для заряджання EV, роблячи їх недоступними для власників електромобілів. Причини можуть бути різними: деякі водії не усвідомлюють важливості цих місць, інші ігнорують правила паркування.

В різних регіонах, щоб запобігти ICEing, впроваджують закони, які передбачають штрафи або навіть евакуацію транспортних засобів, що порушують правила паркування на місцях для заряджання EV.

Серед шляхів вирішення проблеми ICEing є системи виявлення та управління паркувальними місцями. Такі системи використовують датчики, які можуть виявляти наявність автомобіля на зарядному місці і відрізнити електричні транспортні засоби від автомобілів з внутрішнім згоранням. У разі виявлення порушення система може автоматично повідомляти водія, відправляти сповіщення відповідним службам або, навіть, ініціювати евакуацію транспортного засобу.

Системи виявлення та управління паркувальними місцями у більшості випадків використовують методи машинного навчання та обробки зображень, що проявляють високу ефективність у виявленні доступних парковочних місць. Більшість з цих систем досягають точності понад 90%, а деякі навіть досягають точності понад 95% і навіть 96%. Такі результати свідчать про успішність

використання сучасних методів машинного навчання, зокрема глибокого навчання, у вирішенні проблем управління паркуванням і забезпеченні максимальної ефективності використання парковочних місць. Крім того, системи орієнтовані на роботу в реальному часі, що є важливим аспектом для забезпечення актуальної інформації про доступність парковочних місць у режимі онлайн.

Однак у випадку нашої задачі використання методів машинного навчання в парі з обробкою зображень є недоцільним з технічних обмежень. Це вимагає додаткового аналізу систем, які використовують не тільки обробку зображень, а також інші методи виявлення транспортного засобу, зокрема виявлення за допомогою безконтактних датчиків.

Метою роботи є дослідження методів безконтактного виявлення руху та їх використання в моніторингу дорожнього руху та транспорту. Для створення системи виявлення транспортних засобів на основі цих датчиків будуть використані методи обробки сигналів та машинного навчання.

Основне застосування цього дослідження - у зарядних станціях для електромобілів, яке не вимагає необхідності складного калібрування або конфігурації. Наразі не існує рішення, яке б не вимагало додаткової установки для виявлення транспортного засобу, який не заряджається. Це дослідження може допомогти розвинути розумні зарядні станції, які будуть відомі про свій стан зайнятості навіть у випадку, якщо транспортний засіб не розпочав зарядку, що можливо шляхом додавання датчика в систему.

Зважаючи на поставлену мету дослідження щодо виявлення транспортних засобів, у даній роботі треба вирішити такі завдання:

1. провести огляд існуючих технологій виявлення транспортних засобів. Провести детальний огляд наявних методів безконтактного виявлення руху транспортних засобів, зокрема тих, що можуть бути застосовані для виявлення присутності транспортного засобу на зарядних станціях.

2. зробити вибір оптимального датчика. Проаналізувати різні типи датчиків (наприклад, радарні, ультразвукові, оптичні тощо) і визначити найбільш підходящий для використання на зарядних станціях.

3. розробити систему виявлення транспортних засобів. Розробити програмне забезпечення для обробки сигналів і виявлення присутності транспортних засобів на основі обраного датчика з використанням алгоритму обробки сигналів і машинну навчання

4. провести аналіз застосування різних методів машинного навчання, а саме Random Forest, SVC, MLP та CNN.

5. Виконати інтеграцію моделей машинного навчання у мікроконтролер STM32.

6. Провести аналіз результатів роботи. Виявити обмеження рішень, а також запропонувати можливі шляхи щодо його подальшого вдосконалення.

1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ДО ВИРІШЕННЯ ПРОБЛЕМИ ЗАБЛОКУВАННЯ ЗАРЯДНИХ СТАНЦІЙ ДЛЯ ЕЛЕКТРОМОБІЛІВ

1.1 Огляд існуючих підходів до вирішення заблокування зарядних станцій

ICEing, або «заморожування» - це практика заблокування зарядних станцій для електромобілів бензиновими або дизельними автомобілями. ICEing є проблемою для власників електромобілів, які не можуть заряджати свої автомобілі через зайняті зарядні станції [1]. Наприклад, під час свят або в торгових центрах місця для заряджання можуть бути зайняті традиційними авто, попри наявність відповідних знаків, що забороняють це робити [2].

Пропонується кілька підходів до вирішення цієї проблеми:

- законодавчі заходи. Законодавчі заходи є ефективним юридичним інструментом у боротьбі з проблемою ICEing. Шляхом прийняття відповідних законодавчих актів, які забороняють блокування зарядних станцій автомобілями з двигунами внутрішнього згоряння, створюється правова база для регулювання цієї проблеми. Такі закони можуть передбачати встановлення штрафів для порушників, що створюватимуть додатковий стимул для дотримання правил. Наприклад, у штатах Каліфорнія, Флорида та Нью-Йорк вже діють такі закони, що сприяють збереженню доступності інфраструктури для електромобілів [3];

- інформаційна кампанія: Інформаційна кампанія відіграє ключову роль у підвищенні свідомості серед водіїв про проблему ICEing та її вплив на власників електромобілів. Розповсюдження інформації про альтернативні місця для заправки та пояснення важливості уникання блокування зарядних станцій сприяє формуванню правильного ставлення до цієї проблеми;

- розвиток інфраструктури. Розвиток інфраструктури, включаючи збільшення кількості зарядних станцій та їх розміщення в стратегічних місцях, є ключовим для зменшення проблеми ICEing. Забезпечення доступності альтернативних місць зарядки сприяє розвитку ринку електромобілів і зменшує залежність власників електромобілів від існуючих зарядних станцій. Наприклад,

Європейський зелений план встановив амбіційні цілі створення одного мільйона зарядних пунктів для електромобілів до 2025 року [4];

– технологічні рішення. Розробка систем відеоспостереження або використання сенсорів є прикладом технологічного рішення проблеми ICEing, та дозволяє виявляти наявність автомобілів, які блокують зарядні станції, і автоматично повідомляти про це відповідним чином. Що робить процес контролю більш ефективним і зменшує ймовірність блокування зарядних станцій.

Саме технологічні рішення є найбільш перспективними у вирішенні проблеми заблокування зарядних станцій для електромобілів бензиновими або дизельними автомобілями. Використання технологій дозволяє контролювати виконання законодавчих заходів та використання спільного простору водіями, має більш швидке застосування, ніж інформаційна компанія, яка спрямована на зміну ставлення суспільства до електромобілів, та більш дешеве устаткування ніж при розвитку інфраструктури.

Порівняльна таблиця найбільш ефективних технологічних рішень для вирішення проблеми ICEing надано у таблиці 1.1.

Таблиця 1.1 – Порівняльна таблиця ефективних технологічних рішень

Вид системи	Переваги	Недоліки
Інтелектуальні системи керування.	Інтелектуальні системи керування використовують штучний інтелект та алгоритми машинного навчання для аналізу даних з різних джерел. Вони забезпечують ефективне виявлення порушень та можуть передбачати можливі події.	Їх впровадження може бути дорогим, а також вони вимагають потужних обчислювальних ресурсів [5].

Продовження таблиці 1.1

Вид системи	Переваги	Недоліки
Системи відеоспостереження	<p>Системи відеоспостереження можуть забезпечити ефективний спосіб виявлення автомобілів, які неправомірно припарковані біля зарядних станцій. Встановлення камер в області зарядних станцій дозволяє відстежувати транспортні засоби та дії навколо них.</p> <p>Методи, що базуються на візуальному спостереженні, зараз стають більш доступними, особливо за допомогою розвитку класифікаторів зображень на основі машинного навчання, таких як MobileNet v2, які використовують згорткові нейронні мережі.</p>	<p>Це вимагає значних витрат на встановлення та обслуговування камер, а також може бути обмеженим у охопленні області зарядних станцій. Однак такі системи повинні впоратися з різними умовами освітлення, шумами, вітром і навіть артефактами, такими як відбиття або тіні. Але додатковим недоліком систем відеоспостереження є їх потенційна несумісність з вимогами загального регламенту з захисту персональних даних (GDPR). Збір, зберігання та обробка відеоданих можуть потрапляти під дію цих правил, особливо якщо на відеофрагменті зображені особи [6]</p>

Системи датчиків	Системи датчиків можуть швидко реагувати на наявність автомобілів поруч зі зарядними станціями і працювати в режимі реального часу.	Вони можуть бути схильні до помилок, особливо при поганій погоді або в присутності інших перешкод, і вимагають системи обробки даних для ефективного використання інформації. [7]
------------------	---	---

Продовження таблиці 1.1

Системи розпізнавання номерних знаків.	Системи розпізнавання номерних знаків дозволяють швидко ідентифікувати автомобілі та їх власників і можуть бути інтегровані з базами даних для автоматизації процесу ідентифікації.	Вони вимагають великої кількості обробки даних для точного розпізнавання номерних знаків і можуть бути неефективними в разі змінення номерних знаків або блокування камер [8]
--	---	---

Отже, використання датчиків з інтелектуальними системами керування виявляється найбільш оптимальним варіантом, оскільки цей підхід поєднує швидке виявлення автомобілів за допомогою сенсорів з аналізом даних і прогнозуванням можливих подій за допомогою інтелектуальних алгоритмів.

В даній роботі пропонується підхід до розв'язання проблеми ICEing шляхом розробки технології ідентифікації зайнятості місць для заряджання на точках заряджання. Цей підхід використовує бездротові датчики для автоматичного виявлення транспортних засобів на місцях для заряджання та застосовує штучний інтелект для прийняття рішень щодо їх зайнятості. Однією з переваг такого підходу є ефективне вирішення проблеми заборони доступу, що забезпечує рівний доступ до місць для заряджання електромобілів та сприяє переходу до чистих енергетичних технологій у транспорті. Для цього дані, зібрані за допомогою датчика, піддаються обробці за запропонованим потоком обробки сигналів, набору даних та машинного навчання.

Інтеграція цих технологій дозволяє автоматично визначати стан зайнятості місць для заряджання та управляти ними за допомогою інтелектуальних алгоритмів.

1.2 Огляд сучасних датчиків для виявлення руху

Для виявлення руху та присутності транспортних засобів інтелектуальні системи керування можуть використовувати різні типи датчиків. Один з найбільш перспективних варіантів - це бездротові датчики, які можуть автоматично виявляти

транспортні засоби на місцях для заряджання. Ці датчики можуть бути розміщені на поверхні дороги або біля зарядних станцій і спроможні надсилати сигнали про присутність автомобілів безпосередньо до системи керування.

При виборі датчика слід враховувати його характеристики [9], як показано у таблиці 1.2.

Таблиця 1.2 – Технології виявлення транспортних засобів

	Максимальна дальність виявлення	Ромір цілі	Монтаж
Бездротовий магнітометр	Залежить від розміру об'єкта	Всі розміри	Може бути встановлений вище або нижче рівня ґрунту.
Бездротовий ультразвуковий датчик	4 метри	Всі розміри	Повинен бути встановлений вище голови.
Радіолокаційний датчик	40 метрів	Передбачувані цілі	Мінімум 6 футів від цілі.
Оптичний датчик (ІЧ)	200 метрів	5 міліметрів і більше	Потребує установки як для передавача, так і для приймача.
Вимірювальна світлова сітка	2 метри	Всі розміри	Потребує установки для передавача і приймача.

Пасивні акустичні масиви можуть мати проблеми з виявленням тихих транспортних засобів, особливо тих, що рухаються на повільній швидкості. Вони не ідеальні, через те що система має бути встановлене на автостоянках і гаражах.

Іншим варіантом можуть бути оптичні сенсори, які виявляють транспортні засоби за допомогою інфрачервоного або ультразвукового випромінювання. Ці сенсори можуть бути розміщені на зарядних станціях або на стовпах поруч із ними.

Активні інфрачервоні детектори працюють за допомогою випромінювання близького інфрачервоного світла з довжиною хвилі 850 нм, інфрачервоне світло

розсівається та відбивається транспортним засобом, а потім збирається детектором за допомогою широкого оптичного поля. Пристрій може бути використаний для вимірювання швидкості, якщо передаються два або більше променів. Детектори монтувалися над дорогою або з боку дороги понад транспортними засобами. Цей метод схильний до погодних умов, часток у повітрі та блиску від сонця.

Ультразвукові детектори працюють за допомогою випромінювання звукової хвилі поза діапазоном слуху людини. Хвиля відбивається транспортним засобом назад до детектора, де її підбирає приймач. Існують два типи ультразвукових детекторів: режим імпульсного та постійної частоти. Детектор постійної частоти може обчислювати швидкість транспортних засобів за принципом Доплера, однак ця робота зосереджується на пульсуючих ультразвукових датчиках.

Також можуть використовуватися датчики, які здатні виявляти магнітні поля або зміни електричної ємності, які виникають при наявності автомобіля. Ці датчики можуть бути вбудовані безпосередньо у дорожню або стоянкову інфраструктуру [9].

Індуктивні петлі транспортних датчиків можуть виявляти транспортний засіб, який проходить або прибуває до певної точки. Вони використовуються в сучасних системах світлофорів для виявлення присутності транспортних засобів або на автострадах для моніторингу густини трафіку. Індуктивна петля встановлюється під дорожньою поверхнею і працює, спостерігаючи за індуктивністю петель, яка змінюється, коли залізний корпус транспортного засобу знаходиться у близькому контакті з петлею.

Мікрохвильовий радар - це широко використовуваний метод виявлення транспортних засобів [10]. Він працює за допомогою випромінювання радіохвиль та моніторингу хвиль, які відбиваються транспортними засобами. Мікрохвильовий радар працює в діапазоні від 1 до 30 ГГц; обрана робоча частота зазвичай залежить від місцевих регуляцій. Для операцій з управління трафіком зазвичай використовується CW або радар з частотно-модульованою.

CW виявляє переміщення транспортного засобу, що призводить до зміни отриманої частоти через ефект Доплера. FMCW випромінює сигнал, що змінює частоту з часом, і може використовуватися для виявлення навіть стоячих транспортних

засобів. Радарні детектори можуть бути встановлені зверху або з боку дороги. Існують варіанти як для моніторингу багатополосних, так і для моніторингу однополосних доріг.

Найпоширенішими сучасними магнітомерами є мікроелектромеханічні системи, відомі як МЕМС [11]. Ці пристрої використовують чотири основні принципи для виявлення магнітного поля: ефект Холла, магніторезистивність, ефект флюкстейту і сила Лоренца. Вони застосовуються для виявлення транспортних засобів на місцях для заряджання, де вони забезпечують надійну і ефективну роботу системи виявлення.

Магнітометри, або магнітні датчики, працюють за допомогою виявлення змін у магнітному полі Землі, які виникають від диполів, що присутні на кузові транспортного засобу. Ці датчики менш схильні до впливу напруження, що виникає від транспортного руху, порівняно з іншими типами датчиків, наприклад, з індуктивними петлями [12]. Вони також вимагають менше втручання в дорожню поверхню для встановлення. Однак зазвичай звичайні магнітні датчики не можуть ефективно виявляти зупинений транспортний засіб і мають менші зони виявлення порівняно з альтернативами.

Обираючи магнітометр для використання у зарядці електромобілів, можна очікувати ефективну та надійну роботу системи виявлення транспортних засобів, зручність у встановленні та ефективність у використанні в різних умовах через такі фактори:

- низька чутливість до зовнішніх впливів: магнітні датчики менше схильні до впливу зовнішніх факторів, таких як вологість, пил, температурні зміни тощо. Це дозволяє їм працювати надійно в різних умовах середовища, які можуть зустрічатися на зовнішніх зарядних станціях [12];

- ефективність в обмеженому просторі: магнітні датчики можуть бути більш ефективними у виявленні транспортних засобів у вузьких або обмежених просторах, таких як стоянки для зарядки. Це особливо важливо в умовах міського середовища, де місце обмежено;

– менше потреба в технічному обслуговуванні: магнітні датчики зазвичай мають менше потреби в технічному обслуговуванні порівняно з іншими типами датчиків. Це може знизити витрати на утримання та забезпечити стабільну роботу системи протягом тривалого часу;

– широкий діапазон робочих умов: магнітометри можуть працювати в різних умовах, включаючи зовнішні та внутрішні середовища, і мають здатність притосовуватися до різних умов дорожнього руху.

1.3 Порівняльний аналіз існуючих систем для вирішення ICEing

Результати порівняльного аналізу існуючих систем виявлення паркування та прогнозування навантаження на зарядку електромобілів, що наведені у таблицях 1.3 і 1.4, демонструють переваги та недоліки різних підходів і технологій.

Таблиця 1.3 – Порівняльний аналіз існуючих систем для вирішення ICEing

Система/ Особливості	Принцип роботи	Спосіб візуального відображення	Переваги	Недоліки
SPS [13, 14]	Використання ультразвукових сенсорів для вимірювання відстані до автомобілів.	Статус парковочних місць відображається за допомогою LED.	Точне визначення вільних місць та статусу парковочних місць за допомогою LED-індикаторів.	Потенційна нестабільність через ефекти тіні, затемнення та зміни в освітленні.
Система з петлевим детектором [15]:	Використання електричних петель під поверхнею для виявлення присутності автомобіля.	Інформація про присутність автомобіля відображається через мікроконтролер.	Висока надійність виявлення автомобілів та автоматизованість процесів.	Обмеженість у визначенні статусу парковочних місць.

Продовження таблиці 1.3

Прогнозування навантаження на зарядку електромобілів [16, 17]:	Використання монтекарловських симуляцій та моделі SSA-RFR для прогнозування попиту на зарядку.	Прогнозоване навантаження на зарядку відображається графічно або числово.	Точне прогнозування просторового та часового розподілу навантаження.	Потреба у складних обчислювальних методах для прогнозування.
Система паркувального простору [18]:	Використання ультразвукового та візійного 3D датчиків для визначення розміру та розташування парковочного місця.	Інформація про парковочний простір показується візуально за допомогою ультразвукового та візійного датчиків.	Висока точність визначення розмірів та розташування парковочного простору.	Нестійкість обробки даних ультразвукового датчика та вразливість до помилкових виявлень.

Ультразвукові сенсори забезпечують високу точність виявлення вільних місць (понад 95%). Вони дозволяють точно визначати відсутність автомобіля на паркомісці і використовують LED-індикатори для ефективної візуалізації статусу місць. Основним недоліком є можливість нестабільності через ефекти тіні, затемнення та зміни в освітленні.

Система з петлевым детектором забезпечує високу надійність виявлення автомобілів та автоматизацію процесів завдяки використанню мікроконтролерів. Основним недоліком є обмеженість у визначенні статусу паркомісць, таких як зарезервовані або місця для інвалідів.

Прогнозування навантаження на зарядку електромобілів. Монтекарловські симуляції LED-індикатори [15] показали точність прогнозування попиту на зарядку понад 90%, що дозволяє моделювати різні сценарії попиту з високою достовірністю. Модель SSA-RFR [19] демонструє точність прогнозування навантаження близько 85%, ефективно враховуючи складні фактори та попит на зарядні станції. Основним недоліком є потреба у складних обчислювальних методах для прогнозування.

Система паркувального простору. Використання ультразвукового та візійного 3D датчиків забезпечує високу точність визначення розмірів та розташування паркомісця. Комбінований підхід дозволяє покращити точність визначення розмірів паркомісця на 10-50 см порівняно з системами, що використовують лише ультразвукові датчики. Недоліками є нестійкість обробки даних ультразвукових датчиків при наявності шуму та вразливість до помилкового виявлення через перешкоди.

Таблиця 1.4 – Результати аналізу існуючих систем до вирішення ICEing

Система/Метод	Результати Досліджень
SPS	
Ультразвукові сенсори	Точність виявлення вільних місць: більше 95%
	Відсутність автомобіля на паркомісці майже завжди визначається коректно
LED	Ефективна візуалізація статусу паркомісць за допомогою LED
	Кольорове кодування: вільні, зайняті, зарезервовані місця
Прогнозування навантаження на зарядку ЕМ	
Монтекарловські симуляції	Точність прогнозування попиту на зарядку: понад 90%
	Моделювання різних сценаріїв попиту з високою достовірністю
Модель SSA-RFR	Точність прогнозування навантаження: близько 85%
	Ефективність у врахуванні складних факторів та попиту на зарядні станції

Порівняльний аналіз існуючих систем виявлення паркування та прогнозу-

вання навантаження на зарядку електромобілів показав, що комбіноване використання різних сенсорів та методів машинного навчання дозволяє досягти високої точності та надійності. Однак впровадження таких систем потребує врахування специфічних умов експлуатації, можливих перешкод і обмежень. Подальші дослідження мають бути спрямовані на оптимізацію існуючих технологій та розробку нових підходів для підвищення ефективності та стабільності роботи систем в реальних умовах міського середовища. Це дозволить зробити міську інфраструктуру більш адаптивною, ефективною та зручною для користувачів.

Для контролю за паркувальними місцями використовуються різні методи машинного навчання. Було проаналізовано 10 досліджень систем на основі машинного навчання, результат яких подано у таблиці 1.5.

Таблиця 1.5 – Результати аналізу існуючих систем до вирішення проблеми ICEing на основі машинного навчання

№	Основна ідея	Методологія	Переваги
1	Використання глибокого навчання для візуального розпізнавання місць для паркування та обробки картинок [20]	Згорточна нейронна мережа на основі маскованої області (MR-CNN), механізм АОА	- Економічно ефективний алгоритм виявлення паркувальних місць. - Підвищена точність завдяки фокусуванню моделі на відповідних частинах зображення.
2	Використання методів обробки зображень та машинного навчання для підрахунку кількості доступних паркувальних місць у певному місці [21]	Машинне навчання, обробка зображень	- Реальний час підрахунку доступних паркувальних місць. - Можливість ідентифікації та виявлення доступних місць за допомогою машинного навчання. - Зручний інтерфейс для користувачів.

Продовження таблиці 1.5

№	Основна ідея	Методологія	Переваги
3	Вирішення проблеми паркування за допомогою системи паркування на основі Android з використанням недорогих методів обробки зображень [22]	Android, обробка зображень	<ul style="list-style-type: none"> - Виявлення вільних паркувальних зон з використанням технік обробки зображень. - Планування та впровадження системи паркування для торгових центрів.
4	Розробка інтелектуальної системи паркування на основі обробки зображень [23]	Комп'ютерний зір в реальному часі	<ul style="list-style-type: none"> - Розпізнавання вільних та зайнятих паркувальних місць. - Прогнозування доступних паркувальних місць та направляючі індикатори до різних типів паркувальних місць. - Інтеграція з системами відеоспостереження для захоплення та обробки зображень.
5	Оптимізація процесів контролю якості після виробництва продукту, використовуючи систему центрального процесора [24]	Промислова обробка зображень, кіберфізична хмарна система	<ul style="list-style-type: none"> - Визначення аномалій у виробничих лініях за допомогою обробки зображень. - Передача інформації системному адміністратору через кіберфізичну хмарну систему.

Продовження таблиці 1.5

№	Основна ідея	Методологія	Переваги
6	Система пошуку паркувальних місць з використанням обробки зображень і машинного навчання [25]	Машинне навчання, обробка зображень	<ul style="list-style-type: none"> - Реальний час підрахунку доступних паркувальних місць. - Зручний інтерфейс для користувачів. - Інтеграція з іншими системами міста для покращення загальної міської інфраструктури.
7	Інтелектуальна система управління паркуванням на основі обробки зображень [26]	Обробка зображень, камера	<ul style="list-style-type: none"> - Виявлення вільних паркувальних місць на паркувальному майданчику. - Використання камери як сенсора для зйомки фотографій. - Можливість легкої переміщення камери для виявлення різних паркувальних місць. - Зменшення часу пошуку паркувального місця для водіїв.
8	Система управління доступними паркувальними місцями на основі глибокого навчання [27]	Глибоке навчання, розпізнавання номерних знаків	<ul style="list-style-type: none"> - Автоматичне розпізнавання номерних знаків в реальному часі. - Зменшення витрат часу і зусиль на пошук паркувальних місць за рахунок автоматизованого управління. - Підвищення ефективності паркування та використання паркувальних майданчиків.

Продовження таблиці 1.5

№	Основна ідея	Методологія	Переваги
9	Інтелектуальна система паркування на основі обробки зображень [28]	Обробка зображень, комп'ютерний зір в реальному часі	- Розпізнавання вільних та зайнятих паркувальних місць. - Прогнозування доступних паркувальних місць та направляючі індикатори до різних типів паркувальних місць. - Використання існуючих систем відеоспостереження для захоплення та обробки зображень.
10	Система пошуку паркувальних місць за допомогою обробки зображень і машинного навчання [29]	Машинне навчання, обробка зображень	- Реальний час підрахунку доступних паркувальних місць. - Зручний інтерфейс для користувачів. - Можливість інтеграції з іншими системами міста для покращення міської інфраструктури.

Кожне дослідження пропонує свої унікальні рішення для вирішення проблеми нестачі паркувальних місць та оптимізації процесу паркування для водіїв.

Більшість систем виявлення та управління паркувальними місцями використовують методи машинного навчання та обробки зображень, що демонструють високу ефективність у визначенні доступних паркувальних місць. Багато з цих систем досягають точності понад 90%, а деякі навіть досягають точності понад 95% і навіть 96%. [20, 21, 22]. Ці результати свідчать про успішне використання сучасних методів машинного навчання, зокрема глибокого навчання, у розв'язанні проблем управління паркуванням і забезпеченні максимальної ефективності використання парковочних місць. Крім того, ці системи спрямовані на роботу в реальному часі, що є

важливим аспектом для забезпечення актуальної інформації про доступність парковочних місць у режимі онлайн.

1.4 Огляд методів обробки сигналів

Методи обробки сигналів є важливою складовою багатьох галузей, включаючи телекомунікації, обробку звуку та зображення. Ось деякі загальні методи аналізу та фільтрації сигналів:

- перетворення Фур'є - це математичний метод, який використовується для перетворення сигналу з області часу в область частот. Дискретне перетворення Фур'є використовується в аналізі сигналів для визначення домінуючих частот у заданому сигналі [30];

- фільтрація в обробці сигналів - це процес з метою приглушення деякого аспекту сигналу. Розповсюджені типи фільтрів включають у себе фільтри нижніх частот, фільтри верхніх частот, полосопропускні фільтри та фільтри зупинки смуги. Пасова смуга - це діапазон частот, які можуть проходити через фільтр без значної приглушеності [31];

- медіанна фільтрація – це метод, який включає заміну кожної точки даних в сигналі медіанним значенням невеликого вікна сусідніх точок даних. Він ефективно згладжує сигнал, вилучаючи викиди або пікові значення. Розмір вікна може бути налаштований для досягнення балансу між згладжуванням та збереженням деталей [31];

- фільтр Баттерворта - це тип нижньопропускного фільтра, який має пласку частотну відповідь у пасовій смузі та монотонний спад у зупинному діапазоні [32];

- фільтр Савіцького-Голея – метод, який є ефективним і популярним, оскільки дозволяє згладжувати шуми та вибивки в сигналі, а також виконувати диференціювання без значних втрат інформації. Основна ідея полягає в тому, щоб апроксимувати локальні відрізки сигналу поліномом заданого ступеня і потім використовувати цей поліном для обчислення значень згладженого сигналу [33];

Формула для обчислення значень сигналу після застосування фільтру Савіцького-Голея може бути подана таким чином [34]:

$$y_i = \frac{1}{N} \sum_{k=-w}^w c_k * x_{i+k}, \quad (1.1)$$

де y_i – нове значення сигналу на позиції;

x_{i+k} – значення сигналу в позиціях навколо i , де k є від $-w$ до w .

c_k – коефіцієнти фільтра, які визначаються поліномом Савіцького-Голея;

w – ширина вікна фільтра;

N – сума всіх коефіцієнтів c_k .

Завдяки своїй здатності ефективно згладжувати сигнали та видаляти шум та надлишкові вибивки, метод Савіцького-Голея забезпечує покращення якості сигналу, одночасно зберігаючи його структурні особливості. Цей метод є легким у впровадженні та відомий своєю ефективністю у обробці сигналів, при цьому він не потребує великої кількості обчислювальних ресурсів [34]. Ці характеристики роблять його привабливим для використання в машинному навчанні, особливо там, де важливою є швидкість обробки даних. Тому його було обрано як єдиний метод обробки вхідного сигналу магнітометра.

1.5 Огляд методів машинного навчання

Машинне навчання — це швидко розвиваюча галузь, що вивчає алгоритми для виявлення закономірностей у даних та прийняття прогнозів або рішень на їх основі. Існує багато різних методів машинного навчання, кожен з яких має свої сильні та слабкі сторони. У даному розділі розглянуто 4 популярні методи машинного навчання: Random Forest, SVC, MLP та CNN.

Random Forest є ансамблевим методом машинного навчання, який викорис-

товує декілька дерев прийняття рішень для класифікації або регресії. Кожне дерево випадкового лісу навчається на випадковій підвбірці даних і використовує голосування або усереднення результатів для прийняття рішення [34].

Індекс Джині (Gini index) [34] використовується для оцінки "чистоти" вузла в дереві рішень. Це міра того, наскільки часто випадково обраний елемент з вузла неправильно класифікується, якщо вибрати клас за його розподілом у вузлі. Нехай у вузлі t маємо n класів, і p_i - це ймовірність того, що випадково обраний елемент з цього вузла належить класу i . Тоді індекс Джині $I_G(t)$ обчислюється за формулою [34]:

$$I_G(t) = \sum_{i=0}^n p_i * (1 - p_i). \quad (1.2)]$$

Це значення близьке до нуля, якщо всі p_i дуже близькі до 0 або 1, що означає, що вузол добре розділяється за класами. Чим більше значення $I_G(t)$, тим менше чистота вузла [34].

Покращення індексу Джині (Gini impurity improvement) використовується при обчисленні вигоди розділення вузла на два дочірні вузли t_1 і t_2 . Це обчислюється як різниця між індексом Джині батьківського вузла $I_G(\text{parent})$ і зваженою сумою індексів Джині двох дочірніх вузлів $I_G(\text{child } 1)$ $I_G(\text{child } 2)$ [34]:

$$\text{Improvement} = I_G(\text{parent}) - (I_G(\text{child } 1) * w_1 + I_G(\text{child } 2) * w_2), \quad (1.3)$$

де w_1 і w_2 - це ваги, що відображають частку вибірки, яка потрапляє в кожний з дочірніх вузлів.

SVC є одним із найпоширеніших методів машинного навчання для вирішення задач класифікації [35]. Основна ідея методу полягає в знаходженні оптимальної гіперплощини, яка найкраще розділяє дані на два класи. Ця гіперплощина обирається таким чином, щоб максимізувати відстань між нею та найближчими опорними векторами, які є точками даних, що лежать найближче до гіперплощини [35].

Формально, задача SVC полягає в знаходженні гіперплощини:

$$w^T x + b = 0, \quad (1.4)$$

де w — вектор нормалі до гіперплощини;

x — вектор ознак;

b — зміщення (bias), таким чином, щоб максимізувати відстань до найближчих опорних векторів.

Це можна формалізувати за допомогою оптимізаційної задачі з м'якою межею (soft margin) [36], де мінімізується вираз

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i, \quad (1.5)$$

з обмеженнями:

$$\begin{aligned} y_i(w^T x_i + b) &\geq 1 - \xi_i \\ \xi_i &\geq 0, \end{aligned}$$

де (x_i, y_i) — навчальні приклади;

y_i — мітка класу (+1 або -1);

w — вектор вагів;

b — зміщення;

ξ_i — змінна покарання для помилок класифікації;

C — параметр регуляризації, який контролює компроміс між максимізацією розділяючої гіперплощини і мінімізацією помилок класифікації.

Ця формула і її відповідна оптимізаційна задача дозволяють SVC визначати оптимальну гіперплощину для ефективної класифікації даних у просторі ознак [36].

MLP є типом штучної нейронної мережі, що складається з кількох шарів ней-

ронів, включаючи вхідний, прихований та вихідний шари [37]. MLP ефективно моделює складні залежності у даних і використовується для різноманітних завдань, включаючи класифікацію та регресію. Кожен нейрон у мережі з'єднаний з нейронами попереднього та наступного шарів, і має свої власні ваги та функцію активації. MLP може ефективно моделювати складні залежності у вхідних даних завдяки своїй структурі з багатьох шарів [37].

Формула, що описує вихід y_k нейрону k у вихідному шарі MLP, може бути записана як [32]:

$$y_k = f(\sum_{i=1}^{n_h} w_{kj(2)} * f(\sum_{i=1}^{n_x} w_{ji(1)}x_i + b_j(1)) + b_k(2)), \quad (1.6)$$

де x_i - вхідний сигнал i -го нейрону вхідного шару;

$w_{ji(1)}$ - вага з'єднання між i -м вхідним нейроном і j -м нейроном прихованого шару;

f - функція активації, яка застосовується до суми вагованих вхідних сигналів нейрону;

n_x - кількість нейронів у вхідному шарі;

n_h - кількість нейронів у прихованому шарі;

$w_{kj(2)}$ - вага з'єднання між j -м нейроном прихованого шару і k -м нейроном вихідного шару;

$b_j(1)$ і $b_k(1)$ - зсуви (bias) для прихованого і вихідного шарів відповідно.

Функція f може бути будь-якою нелинійною функцією активації, такою як сигмоїдальна функція (наприклад, логістична функція) або ReLU (Rectified Linear Unit) $\text{ReLU}(x)=\max(0,x)$.

Ця формула описує пропагацію сигналів через мультишаровий перцептрон і обчислення виходів нейронів у вихідному шарі з використанням ваг і функцій активації. Ваги w і зсуви b встановлюються під час тренування мережі за допомогою алгоритмів зворотнього поширення помилки (backpropagation), які мінімізують функцію втрат [37].

Згорткові нейронні мережі (CNN) [38] є ключовим інструментом у глибокому навчанні для обробки зображень. Вони відомі своєю здатністю автоматично виявляти просторові шаблони і важливі ознаки у вхідних зображеннях. Основні складові CNN включають згорткові шари для виявлення шаблонів, пулінгові шари для зменшення розмірності функцій і повнозв'язані шари для класифікації або регресії. Операція згортки використовує згорткові фільтри для витягнення ознак з зображень шляхом обчислення лінійних комбінацій пікселів. Пулінгові шари допомагають зменшити кількість параметрів і зробити модель більш устійчивою до малих зміщень у вхідних даних. Повнозв'язані шари об'єднують згорткові функції для здійснення остаточної класифікації або прогнозування [38].

Операція згортки у згортковому шарі описується такою формулою [38]:

$$Y[i, j] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X[i + m, j + n] * K[m, n] \quad (1.7)$$

де $Y[i, j]$ - значення пікселя у feature map;

$X[i + m, j + n]$ - вхідне значення пікселя;

$K[m, n]$ - вага згорткового ядра.

Пулінгові шари використовуються для зменшення розмірності функцій шляхом застосування пулінгової операції:

$$Y_{pool}[i, j] = pool(Y[i * s : i * s + f, j * s : j * s + f]) \quad (1.8)$$

де $Y_{pool}[i, j]$ - результат пулінгу;

s - крок пулінгу;

f - розмір вікна пулінгу.

Повнозв'язаний шар обчислює вихідний сигнал для кожного нейрона за допомогою вагової суми вхідних сигналів і функції активації за формулою [38]:

$$z_k = \sigma \left(\sum_{j=1}^N w_{kj} x_j + b_k \right) \quad (1.9)$$

де z_k - вихід нейрона;

$w_{kj} x_j$ - вага зв'язку;

x_j - вхідний сигнал;

b_k - зсув (bias);

σ - функція активації.

Ці операції спільно дозволяють CNN автоматично виявляти і інтерпретувати важливі зображення властивості, що робить їх ефективними для завдань класифікації та розпізнавання об'єктів у зображеннях [38].

1.6 Класифікаційні показники

У галузі машинного навчання використовуються різні показники для оцінки продуктивності класифікації.

У цьому розділі подано короткий огляд цих показників, включаючи матриці точності, прецизійності, повторного виклику, f1-score та плутанини. Ці показники важливі для оцінки ефективності моделей класифікації та визначення областей для вдосконалення.

Accuracy – це показник, який показує кількість правильних прогнозів порівняно із загальною кількістю прогнозів. Він обчислюється, як показано в рівнянні 1.9.

У деяких випадках точність може вводити в оману, наприклад, незбалансовані набори даних. Існують кращі показники та інструменти оцінки, такі як матриця точності та запам'ятовування або плутанини, які зосереджуються на продуктивності конкретних класів замість загальної продуктивності моделі [39].

$$\text{Accuracy} = \frac{\text{правильні прогнози}}{\text{правильні прогнози} + \text{неправильні прогнози}} \quad (1.10)$$

Precision показує, скільки з прогнозованих позитивних результатів є справжніми позитивними; іншими словами, скільки разів модель правильно визначає даний клас. Precision розраховується наступним чином у рівнянні 1.11 [39].

$$\text{Precision} = \frac{\text{справжні позитивні результати}}{\text{справжні позитивні} + \text{помилкові позитивні результати}} \quad (1.11)$$

Recall описує, який відсоток даного класу було правильно ідентифіковано. Відкликання є корисною метрикою, особливо для незбалансованих наборів даних, де вона забезпечує краще представлення справжньої передбачуваної сили моделі на відміну від точності. Recall розраховується наступним чином у рівнянні 1.12 [39].

$$\text{Recall} = \frac{\text{справжні позитивні результати}}{\text{справжні позитивні} + \text{помилкові негативні результати}} \quad (1.12)$$

F1-Score, також відомий як збалансований F-оцінка або F-показник, є обчисленою метрикою, як показано в рівнянні 25. Це значення від 0 до 1, обчислене на основі точності та запам'ятовування. Він усереднений для багатокласової задачі [39].

$$\text{F1 - score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (1.13)$$

1.7. Висновки до розділу 1

У цьому розділі було проведено дослідження з метою виявлення ефективних методів для систем виявлення паркування (SPS) та прогнозування навантаження на зарядку електромобілів. Результати дослідження підтверджують потенціал використання методів машинного навчання для підвищення ефективності та точності обох систем, зокрема в умовах затемнення та тіні. Розробка автоматичних алгоритмів уп-

равління на основі моделей машинного навчання дозволить покращити роботу систем в реальному часі та забезпечити більш точне передбачення попиту на парковочні місця та зарядні станції.

Було проведено дослідження щодо вибору найкращого датчика для виявлення руху. Після аналізу визначено, що використання магнітометра для цієї мети, зокрема у зарядних станціях для електромобілів, забезпечує ефективну та надійну роботу системи виявлення транспортних засобів. Магнітні датчики відзначаються низькою чутливістю до зовнішніх впливів, таких як вологість, пил або температурні зміни, що дозволяє їм працювати надійно в різних умовах середовища, зокрема на зовнішніх зарядних станціях. Вони також ефективні в обмеженому просторі, що особливо важливо в міському середовищі з обмеженими місцями для зарядки. Магнітні датчики потребують менше технічного обслуговування, що дозволяє знизити витрати на утримання системи і забезпечує її стабільну роботу протягом тривалого часу. Крім того, вони пристосовуються до різних умов дорожнього руху, що підвищує їх універсальність і зручність у використанні.

У розділі було описано обрані чотири популярні методи машинного навчання для вирішення задачі: випадковий ліс, опорні векторні машини (SVM), багаточаровий перцептрон (MLP) і згорткові нейронні мережі (CNN). Кожен з цих методів має свої унікальні характеристики і застосовується для різних завдань класифікації, регресії та обробки зображень. Випадковий ліс базується на ансамблі дерев прийняття рішень, SVM знаходить оптимальну гіперплощину для розділення даних, MLP моделює складні залежності у даних, а CNN використовується для обробки зображень та виявлення важливих ознак.

Це дослідження покликане допомогти вдосконалити технологічні аспекти системи виявлення паркування та управління зарядкою електромобілів через використання методів машинного навчання. Результати вказують на можливість оптимізації функціонування таких систем у реальному часі, зробивши їх більш адаптивними до змінних умов у середовищі, що є важливим кроком у напрямку підвищення їхньої продуктивності та корисності.

2 ТЕХНОЛОГІЯ ТА МЕТОДИ ОБРОБЛЕННЯ ДАНИХ ЩОДО ПАРКУВАННЯ ТРАНСПОРТНИХ ЗАСОБІВ

2.1 Постановка задачі та визначення вимог

Мета - створення та інтеграція системи для виявлення присутності автомобіля на паркувальному місці у STM32 на основі первинної дослідницької пропозиції, зробленої в Eaton European Innovation Center (EEIC). Отже, рішення, запропоноване в цій дипломній роботі, має відповідати наступним обмеженням і функціональним вимогам, викладеним у EEIC [40].

Обмеження та вимоги:

Рішення має бути простим в установці. Встановлення не вимагатиме модифікації місця для паркування, монтажу датчика над головою, складної проводки чи трудомісткого калібрування;

Рішення також має уникати будь-яких камер, щоб виключити ймовірність проблем щодо GDPR (загальний регламент про захист даних) [41];

Точність рішення машинного навчання. Має досягати не менше 80% для зробленого рішення;

Час виявлення змін статусу. Система повинна бути здатна виявляти зміни статусу зайнятості паркового місця протягом не більше 5 хвилин;

Адаптивність до різних умов паркування: Оцінка здатності системи працювати ефективно в різних умовах паркування;

Масштабованість системи до більшого обсягу місць для паркування: Оцінка можливості розширення функціоналу системи та її здатність ефективно працювати з більшою кількістю місць для паркування, забезпечуючи при цьому високу швидкість обробки та точність виявлення стану зайнятості.

Задачі, які потрібно вирішити:

- провести аналіз предметної області;
- проаналізувати існуючі підходи до вирішення проблеми ICEing;
- встановити тестовий стенд;

- провести збір даних, використовуючи 4 паркувальних місця, за допомогою 2 магнітометрів, 4 ультразвукових сенсорів та камери;
- розробити базу даних для зібраних даних;
- обробити сигнал за допомогою фільтра Савіцького-Голея;
- промаркувати зібрані дані для підготовки їх до подальшого аналізу та навчання моделі;
- навчити модель машинного навчання Random Forest;
- інтегрувати навчену модель у STM32 для реалізації практичного застосування.

2.2 Встановлення тестового стенду та збір даних

Дослідження починається з збору даних з 2 магнітних магнітометрів, які були розміщені на парковому місці на такому відстані, що кожен магнітометр охоплює 2 паркових місця. Для валідації даних було обрано використовувати ультразвукові сенсори, розміщені навпроти кожного паркового місця та камеру, як показано на рисунку 2.1.

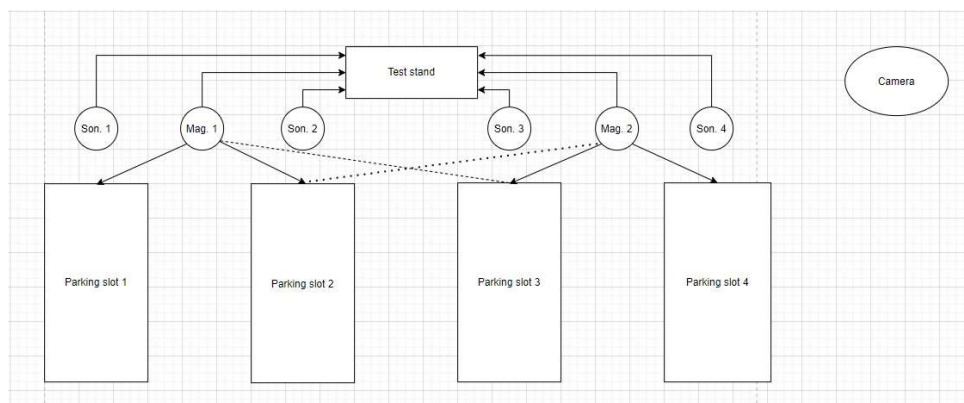


Рисунок 2.1 – Схема розташування магнітометрів та ультразвукових сенсорів на парковці

Розглянемо сценарії для визначення чутливості магнітометра та впливу автомобіля на сигнал:

1. виміряти початковий сигнал магнітометра - провести заміри сигналу магнітометра без наявності автомобіля, щоб визначити базовий рівень сигналу;
2. перевірити чутливість магнітометра через одне паркове місце:
 - розмістити магнітометр та провести заміри сигналу при різних відстанях до автомобіля, паркуючи його на парковому місці;
 - визначити критичну відстань, при якій спостерігається помітна зміна сигналу магнітометра.
3. перевірити паркувальне місце, яке знаходиться на відстані у 2 паркувальних місця від магнітометра та виміряти чутливість магнітометра;
4. перевірити 16 різних сценаріїв:
 - розглянути 16 різних сценаріїв розташування автомобілів на паркових місцях, враховуючи різні відстані та розташування;
 - провести заміри сигналу магнітометра для кожного сценарію.

Після проведення досліджень було встановлено, що для розпізнавання паркувальних місць нам не потрібні два магнітометри. Замість цього, рішення полягає в розміщенні його ліворуч від зарядної станції, що призводить до того, що паркувальне місце P1 отримує сильніший сигнал, як показано на рисунку 2.2.

Для наступного тестування треба врахувати такі фактори:

- збір даних на вулиці або в іншому відкритому просторі, де можуть бути різні умови паркування і перешкоди, такі як інші автомобілі, металеві перешкоди тощо;
- урахування габаритів автомобілів – різні розміри, типи автомобілів, їх маса та габарита, оскільки вони можуть впливати на сигнал магнітометра;
- провести експерименти з різними типами паркування, такими як паралельне, перпендикулярне, діагональне тощо, для збору різноманітних даних;
- врахування металевих перешкод - провести додаткові вимірювання для визначення впливу металевих перешкод, таких як забори, огорожі, на сигнал магнітометра;

– визначення базової відстані між парковальними місцями - встановити стандартну відстань між парковальними місцями для коректного збору даних і аналізу впливу автомобілів один на одного;

– урахування різних видів автомобілів - включити в дослідження різні типи автомобілів, включаючи легкові автомобілі, вантажівки, автобуси тощо.

У подальшому буде визначено оптимальну кількість сигналів для моделі - скільки вимірів сигналу буде потрібно для побудови моделі. Це можна здійснити, враховуючи як мінімальний, так і максимальний сигнали, або збираючи послідовні дані (тривалість, частота тощо) для моделі.

Коф. для М1	100		100	50		10	
Коф. для М2		10		50	100		100
		М1			М2		
	П1		П2	П3		П4	
1	1	260	1	1	260	1	
2	1	250	1	1	160	0	
3	1	210	1	0	160	1	
4	1	200	1	0	60	0	
5	1	160	0	1	210	1	
6	1	150	0	1	110	0	
7	1	110	0	0	110	1	
8	1	100	0	0	10	0	
9	0	160	1	1	250	1	
10	0	150	1	1	150	0	
11	0	110	1	0	150	1	
12	0	100	1	0	50	0	
13	0	60	0	1	200	1	
14	0	50	0	1	100	0	
15	0	10	0	0	100	1	
16	0	0	0	0	0	0	

Рисунок 2.2 – Сценарії для збору даних

2.3 Апаратне забезпечення

Апаратне забезпечення складалося з одноплатного комп'ютера та кількох підключених сенсорних модулів. Beaglebone black [42] був обраний як головний процесор, який збирав дані з усіх датчиків. Апаратне забезпечення було встановлено на опорній платі, щоб забезпечити жорстку конструкцію та забезпечити узгодженість даних. Основним засобом зв'язку в усьому тестовому обладнанні була шина ІС або аналоговий рівень напруг [43], як показано на рисунку 2.3.

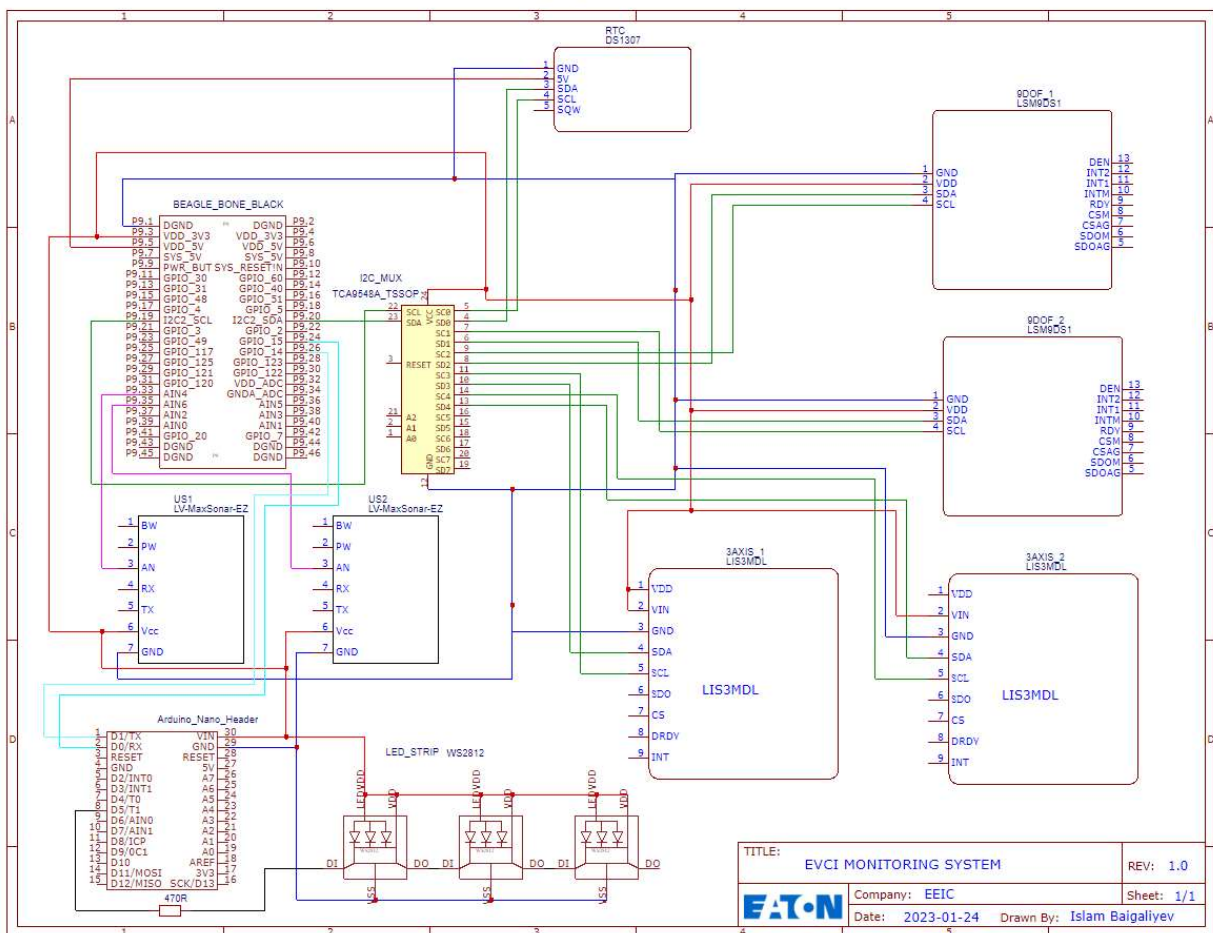


Рисунок 2.3 – Схема HW тестового стенду

Основним завданням було вибрати найбільш підходящі сенсори для цього проекту. Було обрано кілька типів сенсорів: 9-DOF (акселерометр + магнітоскоп +

гіроскоп), 3-осевий магнітоскоп, ширококутний та дуже кутовий ультразвукові сенсори. Ці сенсори були підключені до однієї системи, керованої одноплатним комп'ютером (SBC). Додатково була підключена IP-камера як додатковий засіб моніторингу. Сенсори підключені до SBC за допомогою інтерфейсів I2C [44] та аналогових сигналів. Підключення реалізоване за допомогою Python 3 та CircuitPython [45]. Додатково система включає локальну мережу для віддаленого підключення до мікроконтролера. Для візуалізації даних з сенсорів у реальному часі використовуються програмовані світлодіоди, як показано на рисунку 2.4.

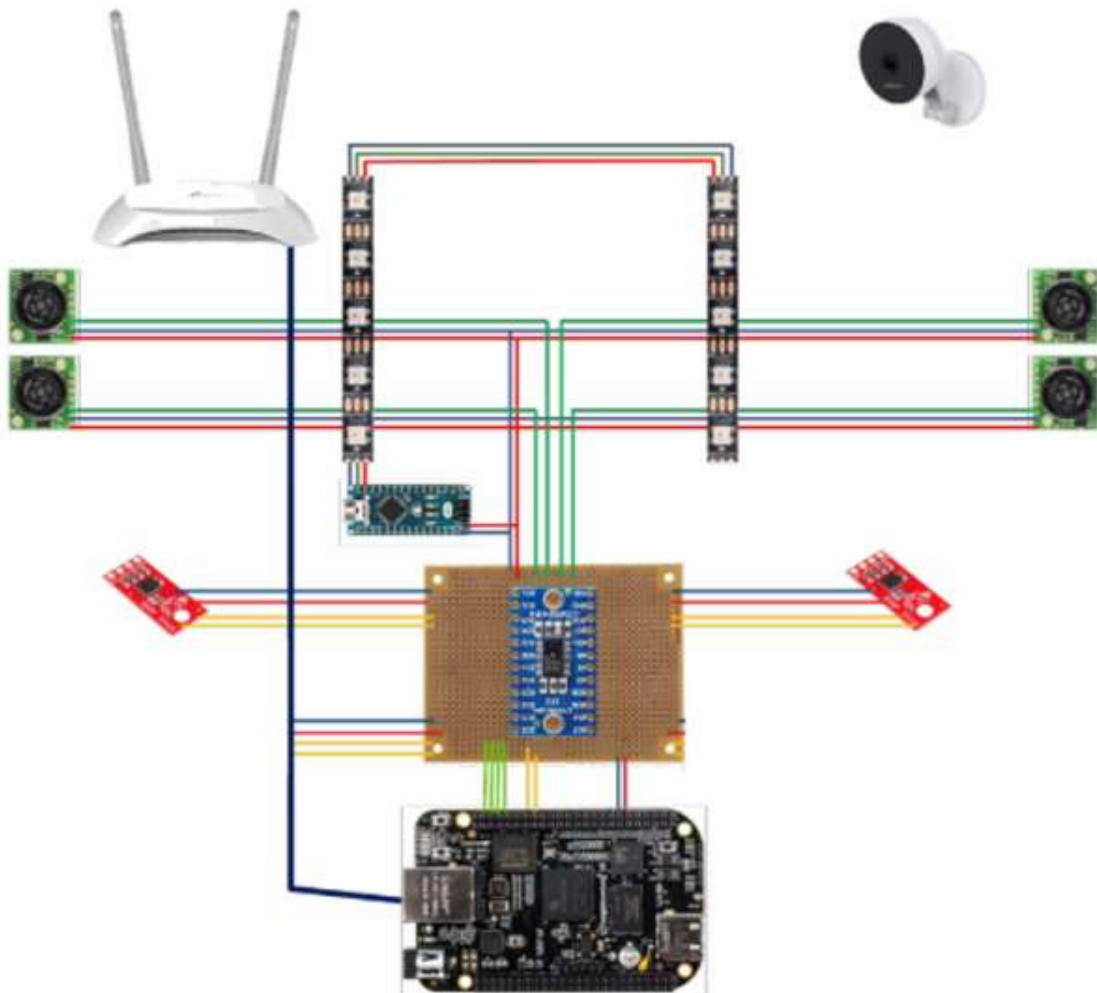


Рисунок 2.4 – Місце знаходження компонентів

У якості магнітометра було обрано Сенсор SparkFun LSM9DS1 Sensor Stick IMU 9Dof, який містить 3 осі гіроскопа, 3 акселерометра та 3 магнітометра. Мікросхема обмінюється даними через інтерфейс I²C, що підтримує швидкість до 400 КГц [46], як показано у таблиці 2.1.

У якості ультразвукових сенсорів було обрано Модулі MaxSonar-EZ4 і MaxSonar-EZO - це ультразвукові далекоміри, які забезпечують виявлення об'єктів малої потужності за допомогою ультразвукових хвиль. Пристрій поєднує в собі перетворювач і детектор і обчислює відстань на основі часу польоту випромінюваної акустичної хвилі [47], як показано у таблиці 2.2.

Таблиця 2.1 – Характеристики плати SparkFun 9DoF Sensor Stick

Робоча напруга	3,3 V
Комунікація	I2C
Шкала лінійного прискорення	±2, ±4, ±8, ±16
Магнітна шкала	±4, ±8, ±12, ±16
Кутова швидкість	±245, ±500, ±2000

Таблиця 2.2 – Характеристики комутаційної плати LV-MaxSonar - EZ

Робоча напруга	2,5 – 5,5 V
Комунікація	RS232, тривалість вихідних імпульсів (PWM), аналоговий рівень напруги
Діапазон (см)	15,2-645
Максимальна частота дискретизації (Гц)	20

2.4 Створення та управління базою даних

Дизайн бази даних був спрощений для полегшення використання і використовувався як сховище даних, тому база даних складеться з однієї таблиці, яка містить дані з випробувального стенду разом з часовими мітками, а саме: 9 значень DoF IMU

1, 9 значень DoF IMU 2 та дані з ультразвукових сенсорів. База даних SQLite використовувалася у випробувальному стенді компанії Eaton.

Дані завантажувалися до цієї бази через імпорт CSV-файлів безпосередньо в базу даних за допомогою SQL-скриптів, які виконувалися у середовищі PyCharm. Це було необхідною процедурою через обмеження доступу до локальної мережі на випробувальному стенді.

SQLite був обраний з кількох причин:

- Простота використання. SQLite є вбудованою базою даних, що не потребує налаштування сервера. Це значно спрощує процес налаштування і використання бази даних. SQLite легко інтегрується з Python за допомогою стандартної бібліотеки `sqlite3`, що дозволяє зручно працювати з базою даних без додаткових залежностей;
- портативність: SQLite зберігає всі дані у одному файлі, що робить її легкою для переноса і резервного копіювання. Це особливо зручно для задач машинного навчання на платформах на кшталт Kaggle, де дані можна легко імпортувати і експортувати для аналізу;
- швидкість та ефективність. Для багатьох застосувань SQLite може бути досить швидкою і ефективною, особливо для читання даних;
- відсутність необхідності в адмініструванні. SQLite не потребує налаштування користувачів, паролів чи прав доступу, що спрощує її використання у невеликих проектах.

Незважаючи на численні переваги, SQLite має свої обмеження, особливо щодо масштабованості [48]:

- обмеження по розміру бази даних. Максимальний розмір бази даних SQLite складає 281 терабайт, але для більшості застосувань розмір файлу може бути суттєво меншим;
- обмеження на кількість одночасних записів. SQLite обмежує кількість одночасних транзакцій на запис. Вона підтримує одночасне читання багатьма користувачами, але тільки один користувач може виконувати запис в базу даних в будь-який момент часу. Це не є критичним для нашої задачі;

– обмеження по продуктивності при великих навантаженнях. Для великих обсягів даних і високих навантажень, таких як обробка великих кількостей запитів або даних в реальному часі, SQLite може не бути найкращим вибором. У таких випадках доцільніше використовувати серверні СУБД, такі як PostgreSQL або MySQL.

СУБД SQLite є ефективним вибором для невеликих проектів, простих застосувань та для розробки і тестування. Її простота у використанні, портативність і інтеграція з Python роблять її ідеальним інструментом для багатьох задач. Однак, для великих проектів або систем, що вимагають високої продуктивності і масштабованості, варто розглянути інші СУБД. У випадку з нашою базою даних, що містить 663 805 записів, SQLite може ефективно виконувати завдання зберігання даних, але слід враховувати її обмеження при зростанні обсягу даних або навантаження. Структура бази даних показана на рисунку 2.4.

Сутність «data» – дані з усіх датчиків:

- id – ідентифікатор сутності;
- timestamp – дата, час;
- 9dof1_mag_x – x-вісь магнітометра у IMU 1;
- 9dof1_mag_y – y-вісь магнітометра у IMU 1;
- 9dof1_mag_z – z-вісь магнітометра у IMU 1;
- 9dof1_gyr_x – x-вісь магнітометра у IMU 2;
- 9dof1_gyr_y – y-вісь магнітометра у IMU 2;
- 9dof1_gyr_z – z-вісь магнітометра у IMU 2;
- 9dof1_acc_x – x-вісь акселерометра у IMU 1;
- 9dof1_acc_y – y-вісь акселерометра у IMU 1;
- 9dof1_acc_z – z-вісь акселерометра у IMU 1;
- 9dof2_mag_x – x-вісь магнітометра у IMU 1;
- 9dof2_mag_y – y-вісь магнітометра у IMU 1;
- 9dof2_mag_z – z-вісь магнітометра у IMU 1;
- 9dof2_gyr_x – x-вісь магнітометра у IMU 2;
- 9dof2_gyr_y – y-вісь магнітометра у IMU 2;

- 9dof2_gyr_z – z-вісь магнітометра у IMU 2;
- 9dof2_acc_x – x-вісь акселерометра у IMU 1;
- 9dof2_acc_y – y-вісь акселерометра у IMU 1;
- 9dof2_acc_z – z-вісь акселерометра у IMU 1;
- us1 – ultrasonic sensor 1;
- us2 – ultrasonic sensor 2;
- us3 – ultrasonic sensor 3;
- us4 – ultrasonic sensor 4.

data	
id	INTEGER
timestamp	DATETIME
9dof1_mag_x	INTEGER
9dof1_mag_y	INTEGER
9dof1_mag_z	INTEGER
9dof1_gyr_x	INTEGER
9dof1_gyr_y	INTEGER
9dof1_gyr_z	INTEGER
9dof1_acc_x	INTEGER
9dof1_acc_y	INTEGER
9dof1_acc_z	INTEGER
9dof2_mag_x	INTEGER
9dof2_mag_y	INTEGER
9dof2_mag_z	INTEGER
9dof2_gyr_x	INTEGER
9dof2_gyr_y	INTEGER
9dof2_gyr_z	INTEGER
9dof2_acc_x	INTEGER
9dof2_acc_y	INTEGER
9dof2_acc_z	INTEGER
us1	INTEGER
us2	INTEGER
us3	INTEGER
us4	INTEGER

Рисунок 2.5 – Фізична модель даних

2.5 Обробка та аналіз сигналів

Процес фільтрації полягає у виборі вікна (або області) уздовж сигналу, в якому буде застосований поліноміальний підхід. Далі, за допомогою методу найменших

квадратів, обчислюються коефіцієнти полінома, що найкраще відповідають цим даним. Після цього значення сигналу у кожній точці вікна розраховуються як значення полінома у відповідній точці. Цей процес повторюється для кожної точки сигналу, що дає згладжений сигнал (див. розділ 1.4).

Важливими параметрами фільтра Савіцького-Голя є розмір вікна і ступінь полінома, формула наведена у розділі 1.4. Величина вікна визначає рівень згладжування: більше вікно забезпечує більше згладжування, але може призвести до втрати деталей у сигналі. Ступінь полінома впливає на рівень апроксимації: вищий ступінь дозволяє більш точно апроксимувати складніші залежності в даних, але може призвести до перенавчання, особливо у випадку наявності шумів.

Використання фільтра Савіцького-Голя з параметрами `polyorder = 3` та `width = 7` за допомогою бібліотеки `scipy.signal` у Python, допомогло зробити сигнал більш гладким, що можна побачити на рисунках 2.6, 2.7.

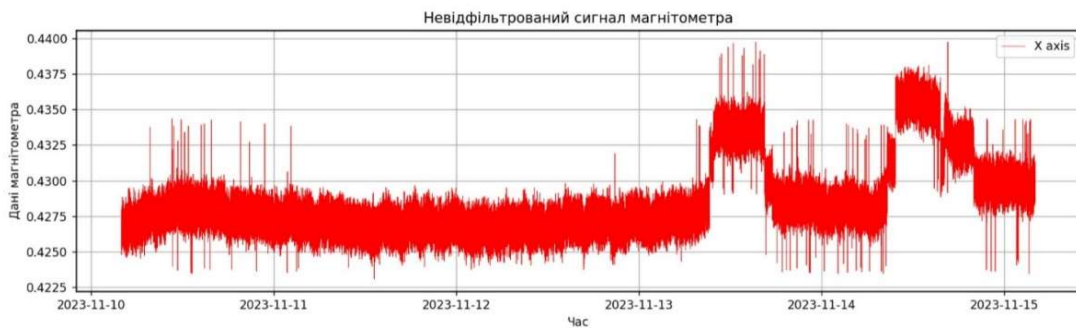


Рисунок 2.6 – Невідфільтрований сигнал

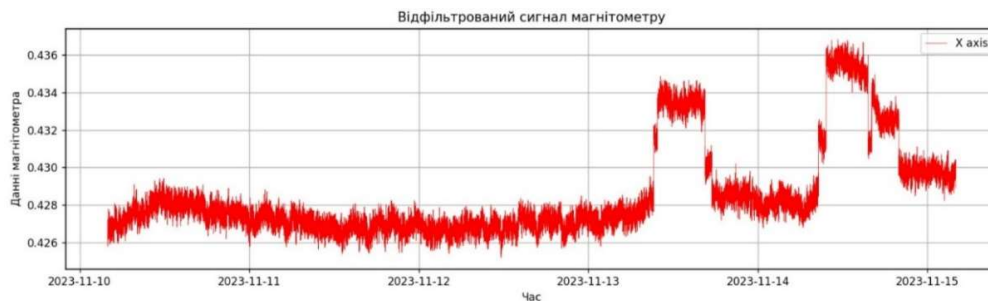


Рисунок 2.7 – Відфільтрований сигнал за допомогою Фільтра Савіцького-Голя

Параметр `polyorder` визначає ступінь полінома, який використовується для апроксимації сигналу. У вашому випадку, `polyorder = 3`, що означає використання кубічного полінома для згладжування сигналу. Це дозволяє ухвалити деяку кривизну та видалити великі варіації у сигналі. Щодо параметра `width`, він вказує на ширину вікна, яке використовується для обчислення коефіцієнтів фільтру. У вашому випадку, `width = 7`, що означає використання вікна завширшки 7 точок для обчислення коефіцієнтів фільтру. Це дозволяє згладжувати сигнал, зберігаючи при цьому деяку деталізацію. Загалом, такий підхід допомагає зменшити шум та видалити випадкові або незначні коливання у сигналі, забезпечуючи більш чітку та плавну візуалізацію. Ви можете експериментувати зі значеннями цих параметрів, щоб підлаштувати фільтр під ваші конкретні потреби та отримати найкращі результати.

2.6 Маркування та підготовка даних для моделювання

Після розгляду даних, вони були класифіковані на 2 стани присутності автомобіля: 1 - "Зайнято", 0 - "Вільно". Хоча спочатку ці події автоматично ідентифікувалися, їх переглянуто вручну для уникнення отримання некоректних даних, оскільки сигнал може коливатися через різні зовнішні фактори. Після проведення тестування ультразвукового сенсора і даних з камери біло прийнято рішення обирати події автоматично, через те що дослідження показало, що ультразвуковий відображає коректно присутність машини, щоб відібрати найкращі дані для моделі машинного навчання. Сирі дані сигналу з ультразвукового сенсору та магнітометра показано на рисунку 2.8.

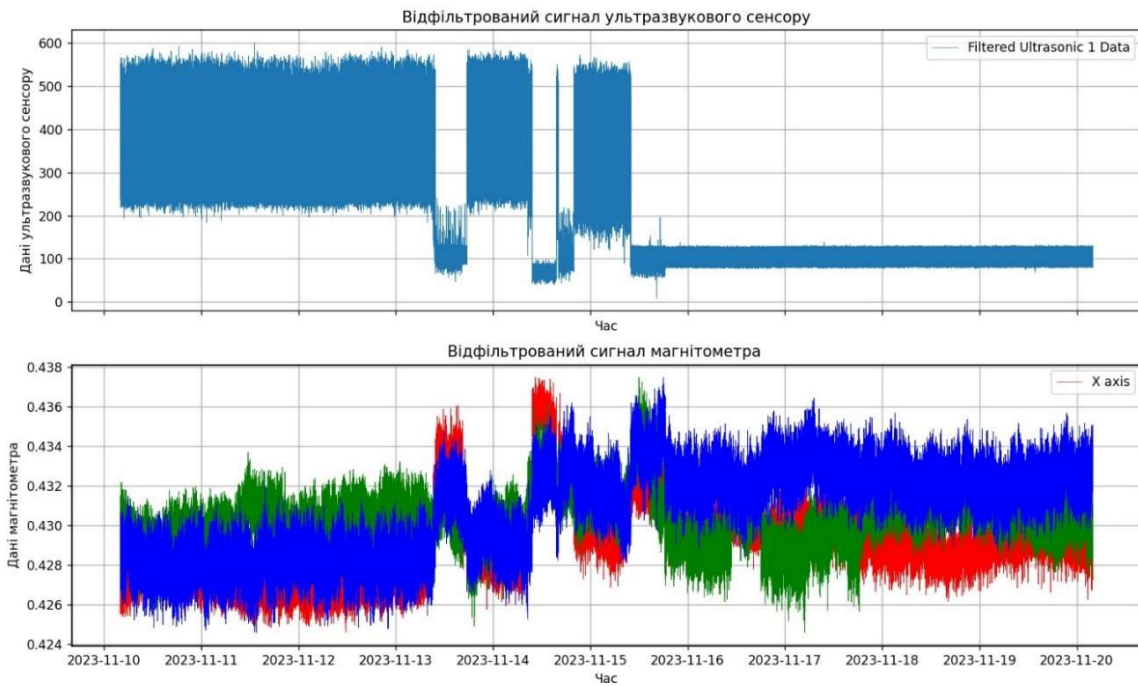


Рисунок 2.8 – Сирий сигнал з ультразвукового сенсору та сигнал магнітометра

Ультразвуковий сенсор був відфільтрований для того щоб з його допомогою виявити наявність або відсутність автомобіля на парковочному місці. Після виявлення не нульового стабільної значення сигналу ультразвукового сенсору алгоритм автоматично обирав проміжки сигналу магнітометра та присвоював стан, як показано на рисунку 2.9.



Рисунок 2.9 – Алгоритм маркування даних магнітометра на основі даних ультразвукового сенсору

Попередньо було розглянуто можливість автоматизації маркування даних для 4 станів паркувального місця “Зайнято”, "Вільно", “Відїжає», «Підїжає», але від цієї ідеї відмовились через неможливість чітко визначати стабільний спад ультразвукового сенсора, як можна побачити на рисунку 2.8 сигнал у ультразвукового сенсора нестабільний, а тому автоматизувати це неможливо. А задача стояла у автоматизації маркування для подальшого перенавчання моделі на нових даних. Запропонований алгоритм маркування для 4 станів показано на рисунку 2.10.

Результат показано на першій діаграммі на рисунках 2.11, 2.12. Дані, отримані в результаті маркування, додавалися до файлу формату CSV у вигляді, наприклад “10, [0.6238239833286828, 0.6280804588681174, 0.40785616920107887]”, де для першого магнітометра, відсутність автомобіля позначалася як подія "10", а присутність - "1". Для другого магнітометра відсутність автомобіля позначалася як "20", а присутність - "2". Усього записів у CSV було 27 588.

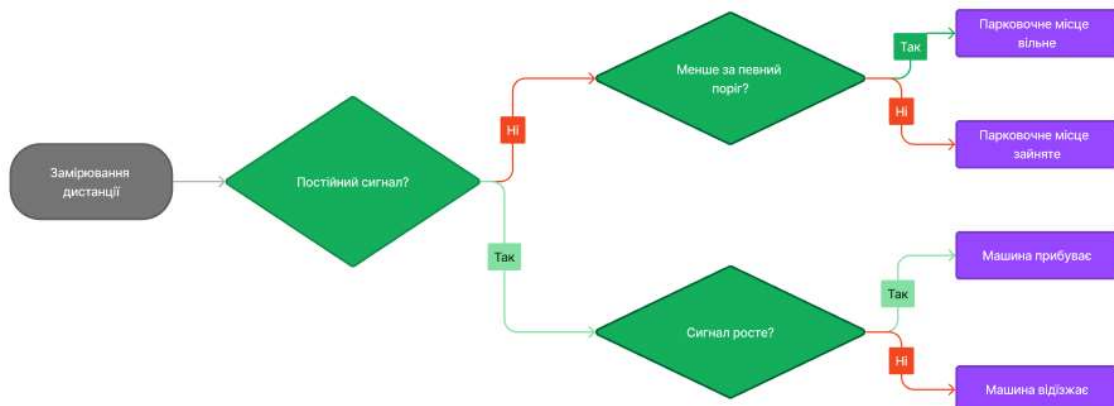


Рисунок 2.10 – Алгоритм маркування даних магнітометра на основі даних ультразвукового сенсору для 4 станів

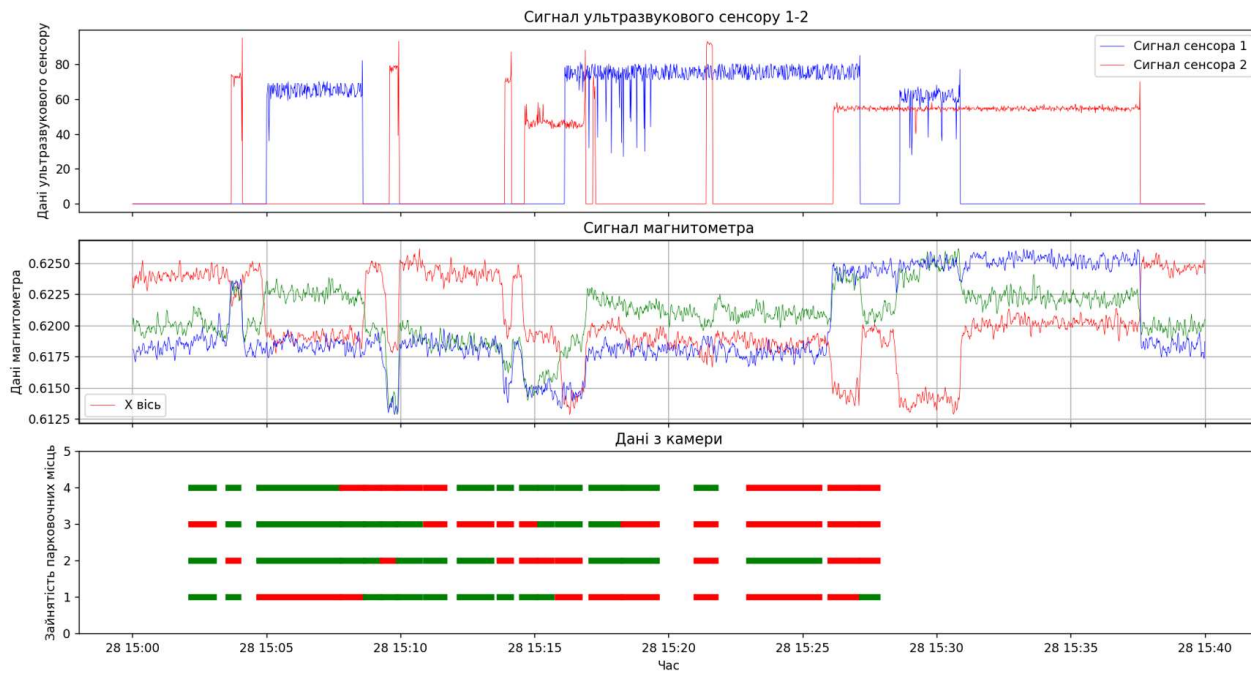


Рисунок 2.11 – Сигнал з ультразвукового сенсору для першого і другого паркувального місця, першого магнітометра та камери

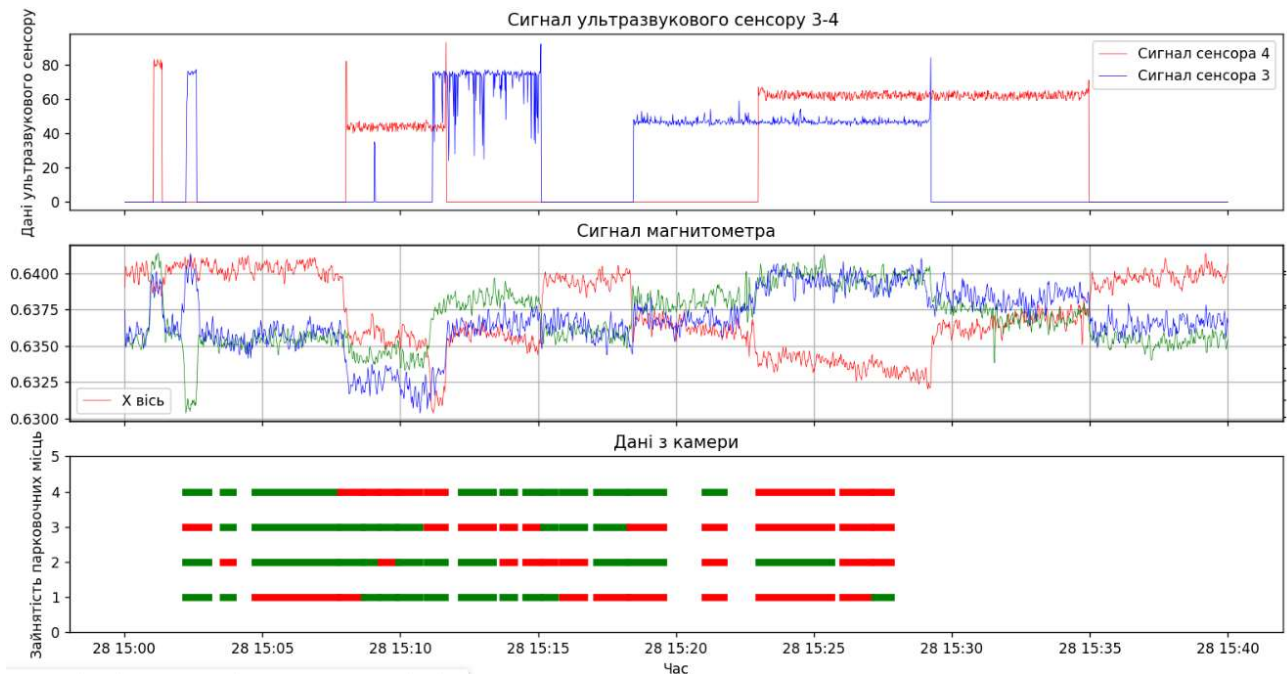


Рисунок 2.12 – Сигнал з ультразвукового сенсору для третього і четвертого паркувального місця, другого магнітометра та камери

2.7 Висновки до розділу 2

У розділі проведений аналіз сенсорів та їх характеристик дозволив зробити обґрунтований вибір у вигляді SparkFun LSM9DS1 Sensor Stick IMU 9Dof для вимірювання руху та магнітних полів, а також ультразвукових сенсорів MaxSonar-EZ4 і MaxSonar-EZO для вимірювання відстані. Представлені характеристики цих сенсорів, такі як робоча напруга, тип комунікації та діапазони вимірювань, що дозволили зробити обґрунтований вибір для вирішення поставленої задачі.

Описано процес створення та управління базою даних, в якій зберігаються дані з різних сенсорів разом з часовими мітками. Використання бази даних SQLite спростило процес зберігання та імпорту даних, що було важливою процедурою для ефективного управління даними з випробувального стенду.

Окремий акцент було зроблено на обробці та аналізі сигналів, зокрема на застосуванні фільтра Савіцького-Голея. Цей метод дозволяє згладжувати сигнали та видаляти шум, що поліпшує якість даних та робить їх більш придатними для подальшого аналізу та моделювання.

Додатковий розділ розглядає процес маркування та підготовки даних для моделювання, що є ключовим етапом у впровадженні системи виявлення автомобілів на парковальних місцях. Під час цього процесу була проведена класифікація сигналів на 2 стани присутності автомобіля ("Зайнято" та "Вільно") для 2 типів паркувального місця (ліве та праве). Це дозволило врахувати різноманітні ситуації та уникнути неправильної інтерпретації даних у реальному часі.

Окрім того, сигнали ультразвукового сенсору та магнітометра були відображені для аналізу та детального вивчення їх характеристик. Це дозволило зрозуміти, як сигнали пов'язані між собою та як вони змінюються залежно від різних умов та вхідних факторів. Отримані дані готові для подальшого аналізу та моделювання для розробки ефективної системи виявлення автомобілів на парковальних місцях.

3 ЗАСТОСУВАННЯ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ ДЛЯ ІНТЕГРАЦІЇ З МІКРОКОНТРОЛЕРОМ У ЗАРЯДНІЙ СТАНЦІЇ ДЛЯ ЕЛЕКТРОМОБІЛІВ

3.1 Метод підготовки даних та застосування алгоритмів машинного навчання

Для аналізу даних та прогнозування подій було використано методи машинного навчання. Для оцінки ефективності було випробувано кілька алгоритмів машинного навчання: SVC, Random Forest, MLP та CNN. Навчання моделі проводилося у середовищі Kaggle [49], що забезпечило оптимальні умови для роботи алгоритмів та підготувало їх для ефективного використання в реальних умовах.

Спочатку з необроблених даних було підготовлено остаточний набір даних, який було розділено на навчальну та тестову частини. Розподіл даних виглядав наступним чином: 60% - для навчання, 20% - для валідації та 20% - для тестування. Набір даних було збалансовано через значний дисбаланс між стаціонарними класами та прибуттями й відльотами. Для цього було знайдено найменш представлений клас, а потім випадковим чином вибрано таку ж кількість зразків з інших класів. Вихідні дані обробки набору даних використовувалися всіма частинами кінцевого конвеєра машинного навчання, тому набори даних було збережено в каталозі `kaggle/working`, щоб зробити їх доступними. Оновлена версія згенерованих наборів даних завжди доступна через посилання на цей блокнот, що забезпечує узгодженість даних під час оцінки моделі.

Моделі запускалися двічі для навчання на лівому та правому паркувальних місцях. Як було визначено в розділі 2.6, для першого магнітометра відсутність автомобіля позначалася як подія "10", а присутність - "1". Для другого магнітометра відсутність автомобіля позначалася як "20", а присутність - "2".

3.2 Тренування та валідація моделі Random Forest

Для аналізу та навчання моделі Random Forest реалізовано такі кроки:

1. визначити параметри сітки для RandomForestClassifier, такі як кількість дерев (`n_estimators`), максимальну глибину дерев (`max_depth`), критерій розбиття (`criterion`), та максимальну кількість ознак для розгляду при розбитті (`max_features`);
2. створити екземпляр RandomForestClassifier;
3. створити екземпляр GridSearchCV для пошуку найкращих параметрів моделі;
4. провести навчання моделі за допомогою методу GridSearchCV.fit() з використанням навчальних даних [50].
5. вивести найкращі параметри та найкращий показник точності, отриманий в процесі навчання. 10 з найкращих можна побачити на рисунку 3.1;

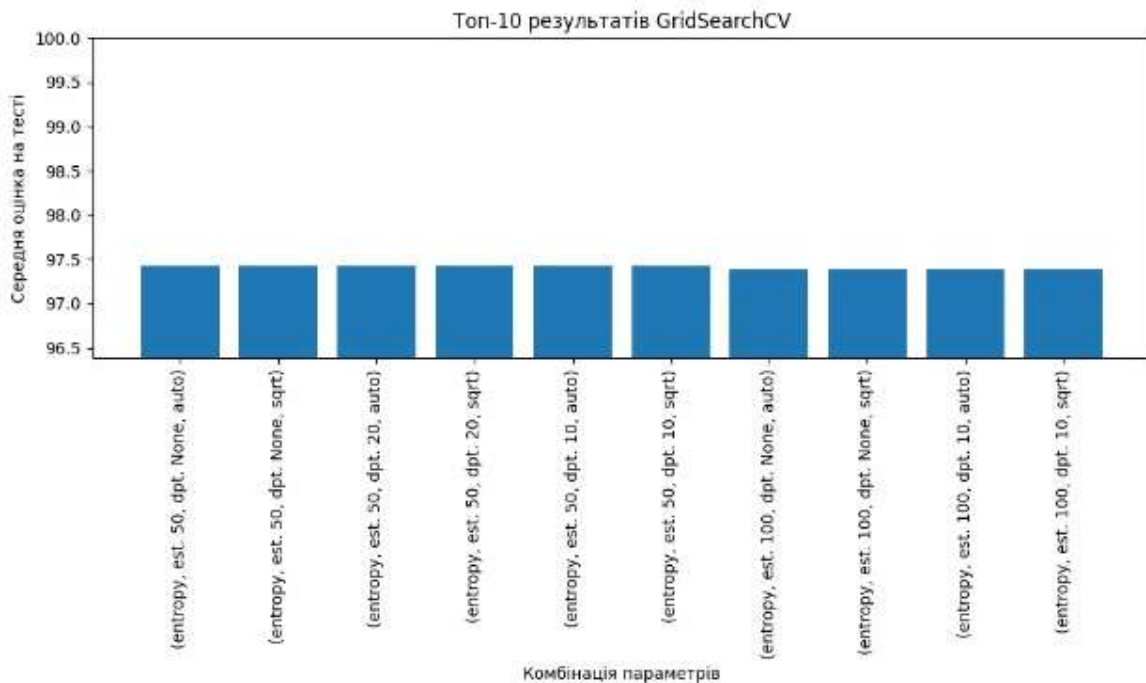


Рисунок 3.1 – Топ- 10 результатів GridSearchCV

Найкращі параметри:

- критерій: ентропія (criterion: entropy);
- максимальна глибина: без обмежень (max_depth: None);
- максимальні характеристики: автоматично (max_features: auto);
- кількість дерев: 50 (n_estimators: 50);
- найкращий показник точності: 0.9743.

6. використати найкращу модель для передбачення на тестовому наборі даних.

7. оцінити результати моделі на тестовому наборі даних за допомогою метрик точності та звіту про класифікацію, що показані на рисунках 3.2, 3.3 та 3.4.

```

Classification Report:

```

	precision	recall	f1-score	support
2	0.97	0.97	0.97	358
28	0.98	0.98	0.98	563
accuracy			0.98	921
macro avg	0.98	0.98	0.98	921
weighted avg	0.98	0.98	0.98	921

Рисунок 3.2 – Звіт з класифікації для правого парковочного місця Random Forest

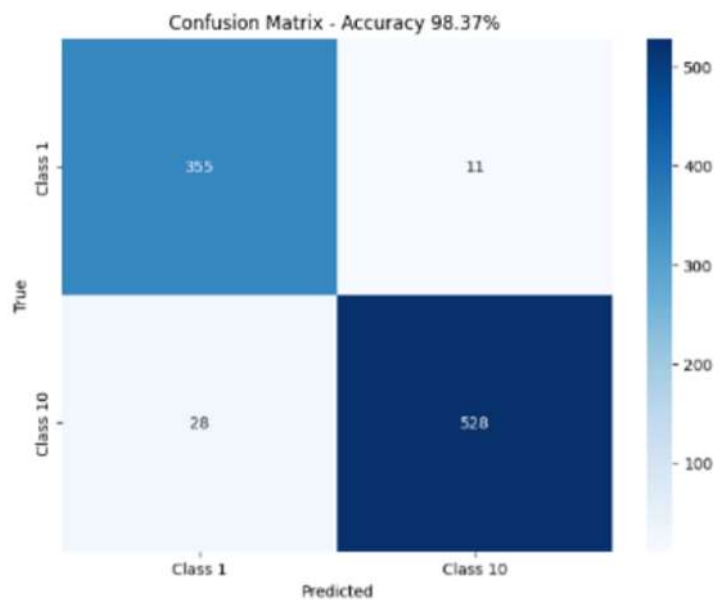


Рисунок 3.3– Матриця невідповідностей (Confusion matrix) Random Forest для лівого паркувального місця

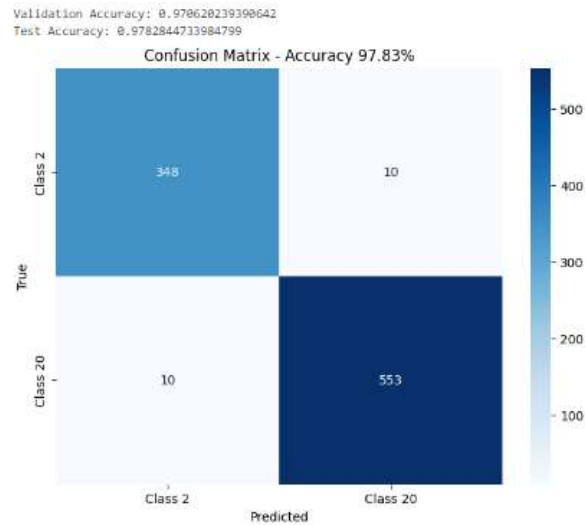


Рисунок 3.4— Матриця невідповідностей (Confusion matrix) Random Forest для правого паркувального місця

8. перевірити модель на нових даних. Для перевірки моделі було обрано схожі дані сигналу магнітометра та викликано модель як показано на рисунку 3.5.

```

new_data = np.array([[0.6335, 0.53355, 0.5663793654]])
new_predictions = grid.predict(new_data)
print("Передбачення для нових даних:", new_predictions)

```

Передбачення для нових даних: [2]

DB Browser

17674	2, [0.6331227620442712, 0.5354890369233634, 0.5657261439732146]
17675	2, [0.6331787109375003, 0.5348830450148813, 0.5661308651878724]
17676	2, [0.6332353864397324, 0.534884963262652, 0.5664665585472473]
17677	2, [0.6334591820126492, 0.5362463557012652, 0.5662747337706617]

Рисунок 3.4 – Перевірка правильності моделі Random Forest

Таблиця 3.1 – Звіт з результатами навчання Random Forest

Метрика	Звіт для лівого паркувального місця	Звіт для правого паркувального місця
Точність валідації	0.9818702490859834	0.970620239390642
Точність тесту	0.9837310195227765	0.9771986970684039
Звіт класифікації		
Клас 1/2		
Точність (precision)	0.97	0.97
Повнота (recall)	0.99	0.97
F1-показник	0.98	0.97

Продовження таблиці 3.1

Метрика	Звіт для лівого паркувального місця	Звіт для правого паркувального місця
Підтримка (support)	366	358
Клас 10/20		
Точність (precision)	0.99	0.98
Повнота (recall)	0.98	0.98
F1-показник	0.99	0.98
Підтримка (support)	556	563
Середнє по макро		
Точність (precision)	0.98	0.98
Повнота (recall)	0.98	0.98
F1-показник	0.98	0.98
Підтримка (support)	922	921
Зважене середнє		
Точність (precision)	0.98	0.98
Повнота (recall)	0.98	0.98
F1-показник	0.98	0.98
Підтримка (support)	922	921

Аналіз результатів навчання моделі Random Forest показав, що точність валідації та тесту для лівого паркувального місця (0.9819 і 0.9837 відповідно) перевищують відповідні показники для правого паркувального місця (0.9706 і 0.9772). Точність (precision), повнота (recall) та F1-показник для класу 1/2 майже однакові в обох випадках (0.97-0.99 для лівого та 0.97 для правого місця). Аналогічно, для класу 10/20 метрики також дуже близькі (0.99/0.98 для точності, 0.98 для повноти, та 0.99/0.98 для F1-показника). Середні по макро та зважені середні показники для всіх метрик також однакові (0.98) для обох паркувальних місць, що свідчить про високу і стабільну ефективність моделі Random Forest у обох випадках (див. табл. 3.1).

3.3 Тренування та валідація моделі SVC

Для проведення аналізу та навчання моделі SVC виконано наступні кроки:

1. використати метод GridSearchCV для пошуку найкращих гіперпараметрів моделі SVC з використанням 5-кратної крос-валідації;
2. встановити найкращі значення гіперпараметрів моделі SVC, такі як $C=100$, $kernel='rbf'$, $gamma='scale'$ та $class_weight=None$. 10 з найкращих можна побачити на рисунку 3.5.

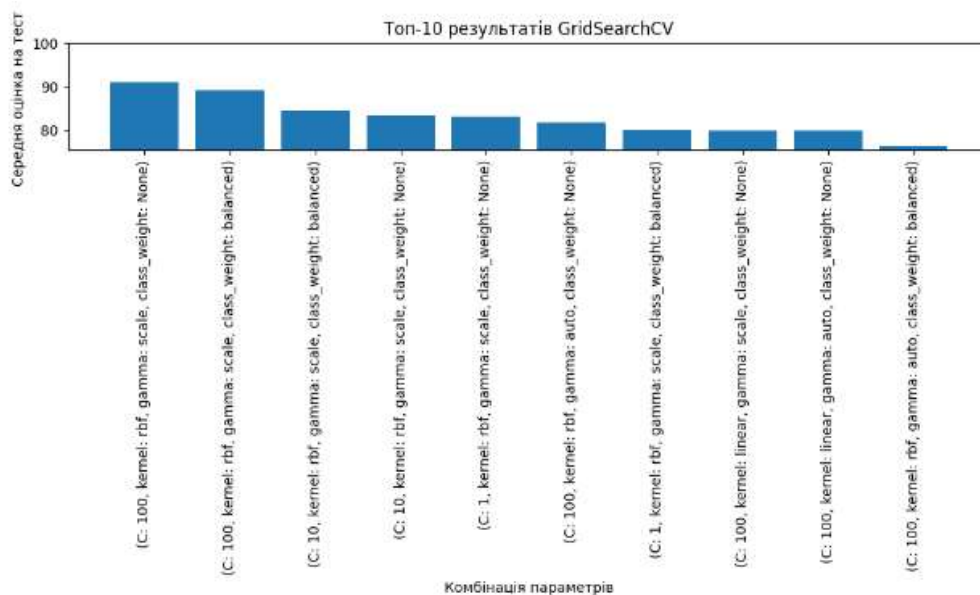


Рисунок 3.5 – Топ- 10 результатів GridSearchCV для SVC

3. навчити модель на наборі даних, використовуючи найкращі параметри;
 4. оцінити точність моделі на валідаційному наборі даних;
 5. використати навчену модель для прогнозування класів на тестовому наборі даних;
 6. оцінити точність класифікації моделі на тестовому наборі даних;
- Модель SVC показала дуже високу точність на тестовому наборі даних, дося-

гнувши значення близько 92.62%. Матриця плутанини та звіт про класифікацію показують, що модель має високу точність для обох класів (Class 2 та Class 20), хоча відмічається дещо краща точність для Class 20, що має більшу підтримку. Також, точність моделі на валідаційному наборі даних склала приблизно 90.32%, що підтверджує стабільність моделі на нових даних.

7. згенерувати звіт про класифікацію, що містить метрики precision, recall та f1-score для кожного класу, а також загальну точність моделі, що показані на рисунках 3.6, 3.7 та 3.8.

```
Fitting 5 folds for each of 32 candidates, totalling 160 fits
Best parameters: {'C': 100, 'class_weight': None, 'gamma': 'scale', 'kernel': 'rbf'}
Best accuracy score: 0.9097155369683578
Test Accuracy: 0.9261672095548317
Classification Report:

```

	precision	recall	f1-score	support
2	0.89	0.92	0.91	358
20	0.95	0.93	0.94	563
accuracy			0.93	921
macro avg	0.92	0.93	0.92	921
weighted avg	0.93	0.93	0.93	921

Рисунок 3.6 – Звіт з класифікації для правого парковочного місця SVC

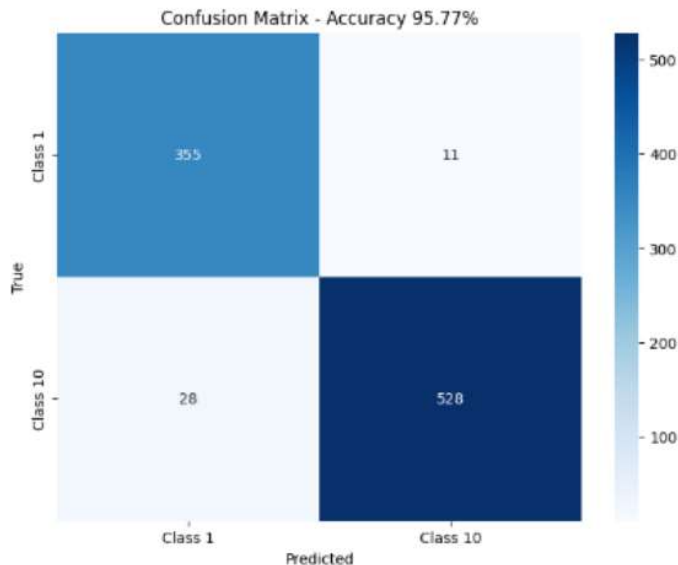


Рисунок 3.7 – Матриця невідповідностей (Confusion matrix) SVC для лівого паркувального місця

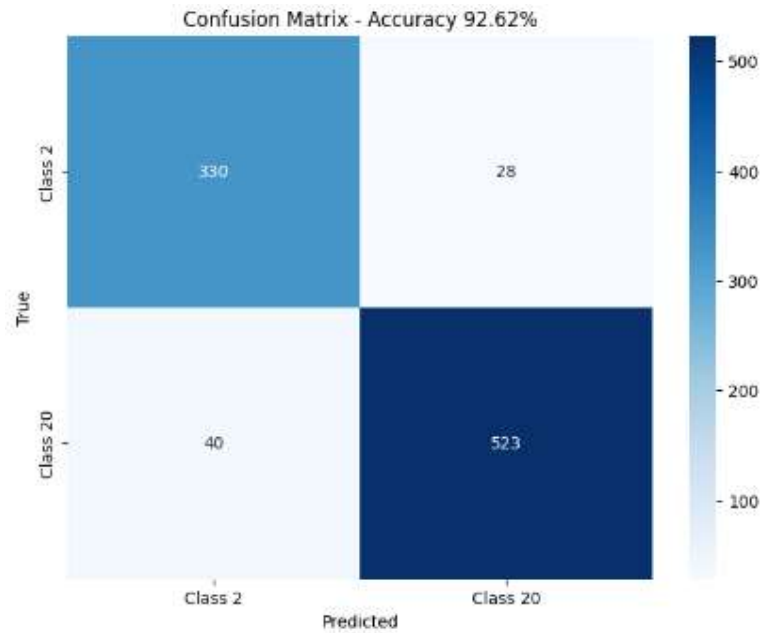


Рисунок 3.8 – Матриця невідповідностей (Confusion matrix) SVC для правого паркувального місця

8. перевірити модель на нових даних. Для перевірки моделі було обрано схожі дані сигналу магнітометра та викликано модель як показано на рисунку 3.9.

Validation Accuracy: 0.9031556039173014
 Test Accuracy: 0.9261671095548317
 SVC Предсказания для новых данных: [2]

Рисунок 3.9 – Перевірка правильності моделі SVC

Таблиця 3.2 – Звіт з результатами навчання SVC

Метрика	Перший звіт	Другий звіт
Точність валідації	0.9506884715536967	0.9031556039173014
Точність тесту	0.9577006507592191	0.9261671095548317
Звіт класифікації		
Клас 1/2		
Точність (precision)	0.93	0.89
Повнота (recall)	0.97	0.92

Продовження таблиці 3.2

Метрика	Звіт для лівого паркуваль-ного місця	Звіт для правого парку-вального місця
F1-показник	0.95	0.91
Підтримка (support)	366	358
Клас 10/20		
Точність (precision)	0.98	0.95
Повнота (recall)	0.95	0.93
F1-показник	0.96	0.94
Підтримка (support)	556	563
Середнє по макро		
Точність (precision)	0.95	0.92
Повнота (recall)	0.96	0.93
F1-показник	0.96	0.92
Підтримка (support)	922	921
Зважене середнє		
Точність (precision)	0.96	0.93
Повнота (recall)	0.96	0.93
F1-показник	0.96	0.93
Підтримка (support)	922	921

Аналіз результатів навчання моделі SVC показав, що перший звіт демонструє вищі показники точності валідації (0.9507) та тесту (0.9577) порівняно з другим звітом (0.9032 і 0.9262). Для класу 1/2 точність (precision), повнота (recall) та F1-показник у першому звіті також перевищують показники другого звіту, що вказує на кращу ефективність моделі. Аналогічно, для класу 10/20 всі метрики вищі в першому звіті. Середні по макро та зважені середні показники для всіх метрик підтверджують, що модель у першому звіті має загально вищу точність і надійність, що підкреслює необхідність ретельного підбору та налаштування моделей машинного навчання (див. табл. 3.2).

3.4 Тренування та валідація моделі MLP

Для проведення аналізу та навчання моделі MLPClassifier реалізовано такі кроки:

1. визначити параметри моделі, включаючи функцію активації, параметр регуляризації, конфігурацію скритих шарів, розклад швидкості навчання, максимальну кількість ітерацій та метод оптимізації;
2. створити класифікатор MLPClassifier та вказали в ньому параметр `random_state=42` для відтворюваності результатів;
3. використати GridSearchCV для перебору гіперпараметрів моделі за допомогою перехресної перевірки на 5 фолдах;
4. навчити модель на тренувальному наборі даних (`X_train`, `y_train`) за допомогою методу `fit()`;
5. вивести найкращі параметри та найкращу точність, які були отримані під час пошуку гіперпараметрів;
6. використати найкращу модель для передбачення класів на тестовому наборі даних (`X_test`), використовуючи метод `predict()`;
7. оцінити результати класифікації на тестовому наборі даних, вивівши звіт про класифікацію, який включав `precision`, `recall` та `f1-score` для кожного класу, а також загальну точність моделі;
8. зробити висновок про роботу та результати навчання моделі MLPClassifier.

```
Best accuracy score: 0.646820668613062
Accuracy on test set: 0.8219326818675353
Classification Report:

```

	precision	recall	f1-score	support
2	0.81	0.71	0.76	358
20	0.83	0.89	0.86	563
accuracy			0.82	921
macro avg	0.82	0.80	0.81	921
weighted avg	0.82	0.82	0.82	921

Рисунок 3.10 – Звіт з класифікації для правого парковочного місця MLP

```
Validation Accuracy: 0.7899891186071817
Test Accuracy: 0.8219326818675353
MLPClassifier Предсказания для новых данных: [2]
```

Рисунок 3.11 – Перевірка правильності моделі MLP

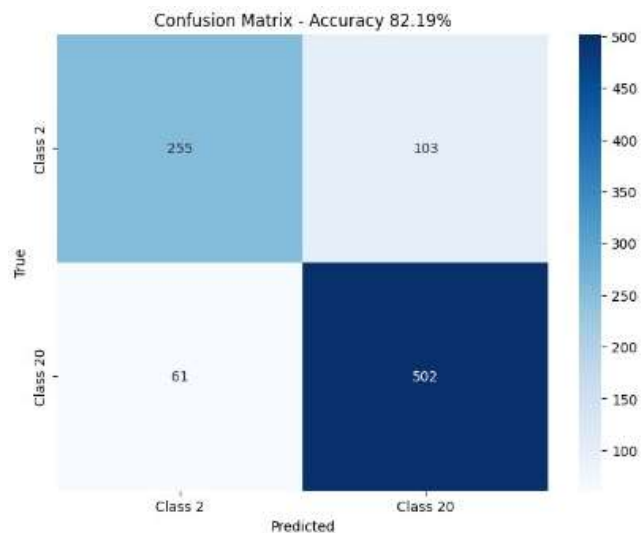


Рисунок 3.12– Матриця невідповідностей (Confusion matrix) MLP для правого паркувального місця

Результати використання моделі MLPClassifier показали наступне:

Найкращі параметри моделі були визначені як:

Функція активації (activation): 'tanh'

- параметр регуляризації (alpha): 0.0001;
- конфігурація скритих шарів (hidden_layer_sizes): (50, 50);
- розклад швидкості навчання (learning_rate): 'constant';
- максимальна кількість ітерацій (max_iter): 300;
- метод оптимізації (solver): 'adam'.

Модель була навчена на тренувальному наборі даних та показала найкращий результат на валідаційному наборі з точністю близько 64.68%.

Після навчання модель була протестована на тестовому наборі даних, де була досягнута точність приблизно 82.19%.

Аналіз результатів класифікації показав, що модель має добру здатність до

класифікації обох класів "2" і "20" з високими значеннями precision, recall та f1-score. Також, загальна точність моделі на тестовому наборі є досить високою (близько 82%).

Отже, можна зробити висновок, що модель MLPClassifier показала дуже непогані результати.

3.5 Тренування та валідація моделі CNN

Для проведення аналізу та навчання моделі CNN ми виконали наступні кроки:

1. визначити архітектуру моделі, яка включає в себе послідовність згорткових шарів, шарів пулінгу та повнозв'язних шарів;
2. побудувати модель згорткової нейронної мережі (CNN) за допомогою бібліотеки TensorFlow;
3. навчити модель на тренувальному наборі даних (X_{train} , y_{train}) за допомогою методу `fit()`;
4. вивести звіт про навчання моделі, який містить метрики точності (accuracy) та втрат (loss) на тренувальному та валідаційному наборах даних на кожній епосі;
5. оцінити результати класифікації на тестовому наборі даних, використовуючи метод `evaluate()` або `predict()`;
6. вивести звіт про класифікацію, який включає метрики precision, recall та f1-score для кожного класу, а також загальну точність моделі;
7. зробити висновок про роботу та результати навчання моделі CNN.

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 2, 32)	96
max_pooling1d (MaxPooling1D)	(None, 2, 32)	0
conv1d_1 (Conv1D)	(None, 2, 64)	2,112
max_pooling1d_1 (MaxPooling1D)	(None, 2, 64)	0
flatten (Flatten)	(None, 128)	0
dense (Dense)	(None, 64)	8,256
dense_1 (Dense)	(None, 1)	65

Total params: 10,529 (41.13 KB)

Trainable params: 10,529 (41.13 KB)

Non-trainable params: 0 (0.00 B)

Epoch 1/10

241/241 — 3s 4ms/step - accuracy: 0.6971 - loss: 0.6147 - val_accuracy: 0.7225 - val_loss: 0.5908

Epoch 2/10

241/241 — 1s 3ms/step - accuracy: 0.7204 - loss: 0.5931 - val_accuracy: 0.7225 - val_loss: 0.5939

Epoch 3/10

241/241 — 1s 3ms/step - accuracy: 0.7234 - loss: 0.5901 - val_accuracy: 0.7225 - val_loss: 0.5906

Epoch 4/10

241/241 — 1s 3ms/step - accuracy: 0.7294 - loss: 0.5843 - val_accuracy: 0.7225 - val_loss: 0.5903

Epoch 5/10

241/241 — 1s 3ms/step - accuracy: 0.7189 - loss: 0.5947 - val_accuracy: 0.7225 - val_loss: 0.5926

Epoch 6/10

241/241 — 1s 3ms/step - accuracy: 0.7179 - loss: 0.5975 - val_accuracy: 0.7225 - val_loss: 0.5904

Epoch 7/10

241/241 — 1s 3ms/step - accuracy: 0.7219 - loss: 0.5931 - val_accuracy: 0.7225 - val_loss: 0.5947

Epoch 8/10

241/241 — 1s 3ms/step - accuracy: 0.7075 - loss: 0.6055 - val_accuracy: 0.7225 - val_loss: 0.5903

Epoch 9/10

241/241 — 1s 3ms/step - accuracy: 0.7356 - loss: 0.5782 - val_accuracy: 0.7225 - val_loss: 0.5909

Epoch 10/10

241/241 — 1s 3ms/step - accuracy: 0.7294 - loss: 0.5828 - val_accuracy: 0.7225 - val_loss: 0.5905

41/41 — 0s 2ms/step - accuracy: 0.7510 - loss: 0.5660

Test Accuracy: 0.7218337059020996

41/41 — 0s 4ms/step

Classification Report:

	precision	recall	f1-score	support
Class 1	0.72	1.00	0.84	929
Class 0	0.00	0.00	0.00	358
accuracy			0.72	1287
macro avg	0.36	0.50	0.42	1287
weighted avg	0.52	0.72	0.61	1287

Test Accuracy: 0.7218337218337219

Рисунок 3.13 – Звіт з класифікації для правого паркувального місця CNN

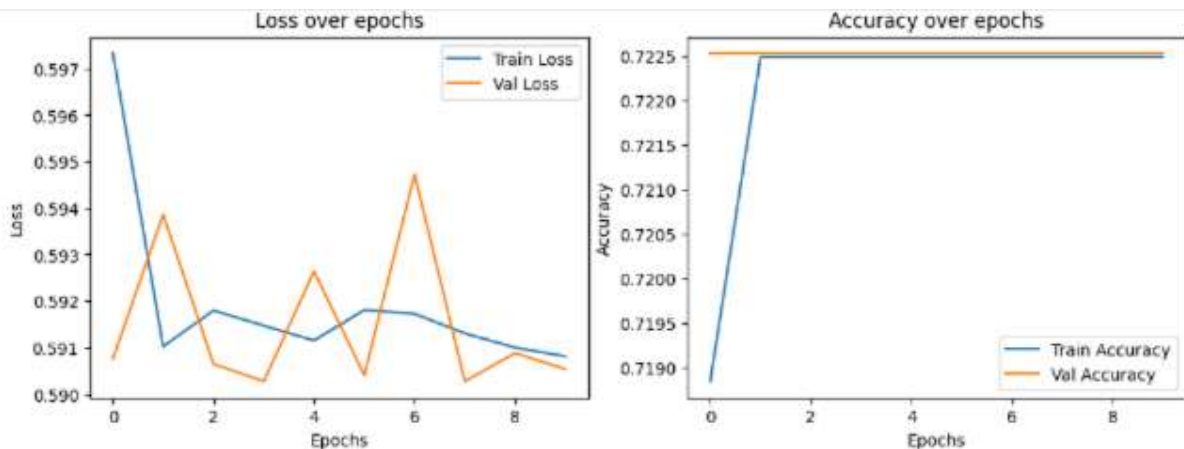


Рисунок 3.14 – Діаграма Loss over epochs and Accuracy over epochs

Втрати протягом епох (Loss over epochs) – це графік, який показує, як змінюються втрати на тренувальній та валідаційній вибірках протягом кожної епохи. Ідеально, втрати повинні зменшуватися протягом навчання, що вказує на те, що модель краще підлаштовується під дані як показано на рисунку 3.14.

Точність протягом епох (Accuracy over epochs) – це графік, який показує, як змінюється точність на тренувальній та валідаційній вибірках протягом кожної епохи. Ідеально, точність повинна збільшуватися протягом навчання, що вказує на покращення продуктивності моделі. Графіки точності протягом епох та витрат протягом епох навчання CNN показано на рисунку 3.14.

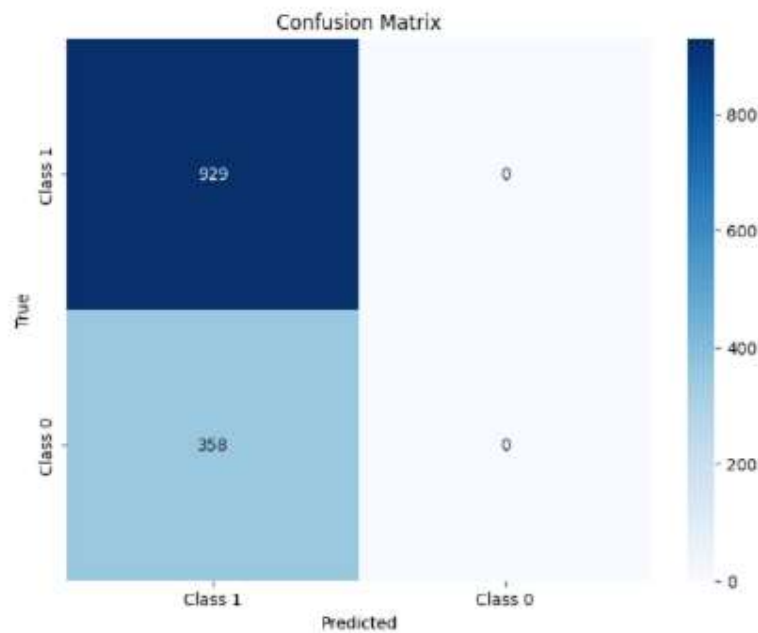


Рисунок 3.15– Матриця невідповідностей (Confusion matrix) CNN для правого паркувального місця

Результати використання моделі CNN показали:

Модель CNN має таку архітектуру:

- загальна кількість параметрів: 10,529;
- кількість параметрів, що можна навчати: 10,529;
- кількість ненавчальних параметрів: 0.

Модель була навчена на протязі 10 епох, під час яких вдалося досягти найкращої точності на валідаційному наборі приблизно 72.25%. Проте точність на тестовому наборі даних складає приблизно 72.18%, що свідчить про певну загальну здатність моделі до узагальнення, але можливо є місце для поліпшення (див. 3.13).

Результати класифікації:

1. Точність (Precision): для "Class 1" вона становить близько 72%, що означає, що більшість елементів, які модель визначила як "Class 1", насправді належать до цього класу. Проте для "Class 0" точність дорівнює 0%, що свідчить про те, що модель навіть не розпізнає цей клас.

2. Повнота (Recall): для "Class 1" повнота також становить близько 100%, що означає, що модель виявляє більшість елементів цього класу. Проте для "Class 0" повнота також дорівнює 0%, що свідчить про те, що модель не виявляє елементи цього класу.

3. F1-оцінка (F1-score): вагована середня точності та повноти для "Class 1" складає близько 84%, що є рівнем, який можна вважати прийнятним. Проте для "Class 0" F1-оцінка дорівнює 0%, що свідчить про слабкий класифікатор для цього класу.

Отже, можна зробити висновок, що модель CNN має певний потенціал у класифікації, але потребує подальшої оптимізації та налаштування для досягнення кращих результатів. Це може включати зміну архітектури, налаштування гіперпараметрів, збільшення обсягу даних або використання додаткових методів регуляризації та оптимізації.

3.6 Інтеграція моделі машинного навчання у мікроконтролер

У цьому підрозділі розглянуто процес інтеграції розробленої моделі машинного навчання у мікроконтролер. Описано технології та інструменти, використані для підключення моделі до мікроконтролера, а також процес передачі та обробки даних.

Наступним кроком була інтеграція моделі Random Forest на мікроконтролер

STM32mp157f-dk2, що включала отримання та обробку даних сигналу магнітометрів з подальшим тестуванням. Для цього була використана технологія з'єднання чіпу з магнітометрами через Modbus та використання RS-485 протоколу. Запропонована система була спроектована для ефективного виявлення наявності або відсутності автомобілів на парковому місці. Оскільки планується реалізувати цю функцію на десятках тисяч некаліброваних пристроїв, які знаходяться у різних умовах, включаючи безперервно змінні умови, система повинна бути досить стійкою до різноманітних сценаріїв та змінних умов – зокрема, змінності шуму та електромагнітних перешкод, що можуть виникнути у реальному середовищі (див. 3.16). Для досягнення цієї мети система моніторингу має базуватися на найкращому, після тестування, алгоритмі машинного навчання Random Forest, який може ефективно адаптуватися до різних умов і забезпечувати точні результати [51].

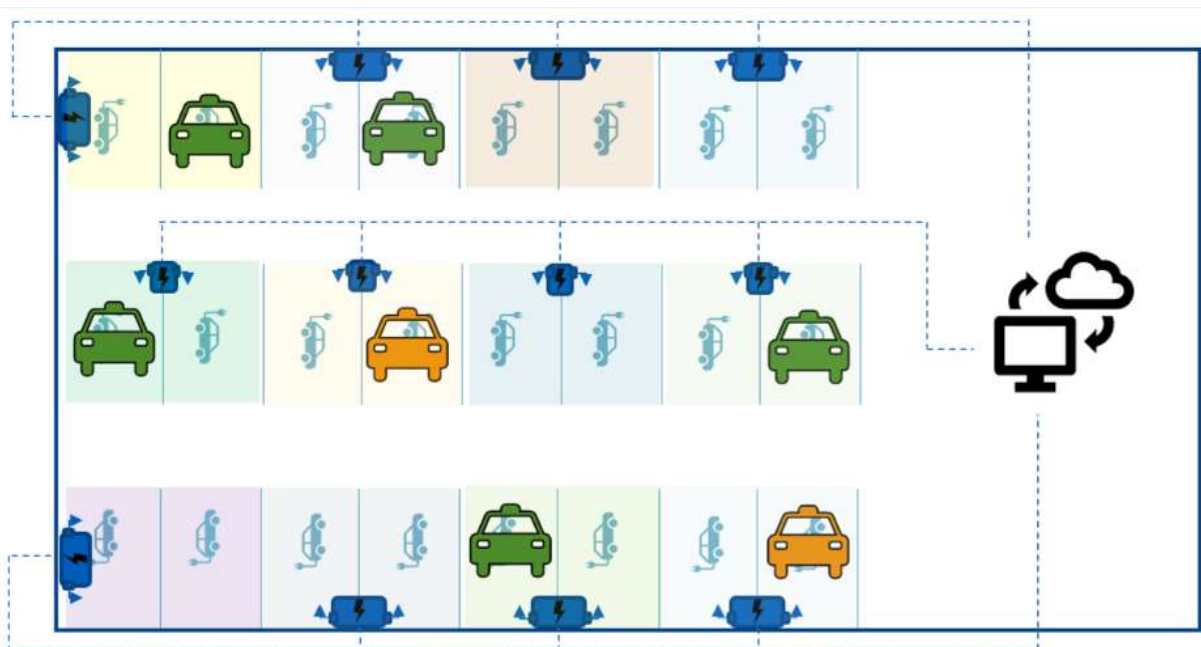


Рисунок 3.16 – Схема використання моделі у реальному середовищі

Основні кроки, які треба виконати для досягнення цієї мети:

- навчання моделі на Kaggle: довести, що модель, яка генерується, може бути використана на мікроконтролері з обмеженими ресурсами, такими як STM32, що було виконано у розділі 3;

- експорт моделі: Після навчання моделі експортувати її у формата, який підтримується мікроконтролером STM32, що було виконано у розділі 3;
- перенесення моделі на STM32: Розробити програму на Python або C, яка взаємодіє з мікроконтролером STM32 і завантажує модель машинного навчання на нього. Використати бібліотеки, такі як TensorFlow Lite Micro, які дозволяють виконувати нейронні мережі на мікроконтролерах з обмеженими ресурсами;
- налаштування середовища виконання: Мікроконтролер STM32 повинен мати належне середовище виконання, необхідне для виконання моделі машинного навчання. Це може включати в себе драйвери, бібліотеки та інші компоненти програмного забезпечення, необхідні для роботи з нейронними мережами;
- тестування та налагодження: Після завантаження моделі на мікроконтролер здійснить тестування та налагодження її роботи. Впевніться, що модель працює правильно і даватиме очікувані результати на вашому мікроконтролері.

Процес інтеграції включав такі основні кроки:

1. прошивка мікроконтролера. Початковим кроком було прошивання мікроконтролера STM32mp157f-dk2 відповідною прошивкою, яка забезпечує базове функціонування системи та підключення периферійних пристроїв;
2. встановлення Python та додаткових бібліотек. Для виконання моделі Random Forest на мікроконтролері, було необхідно встановити Python та відповідні бібліотеки, що підтримують обробку даних та виконання алгоритмів машинного навчання;
3. запакування моделі у формат ONNX [52].

Модель Random Forest була запакована у формат ONNX (Open Neural Network Exchange), що дозволяє зручно переносити та розгортати модель на різних платформах;

4. розгортання моделі на STM та запуск. Після підготовки моделі у форматі ONNX, вона була розгорнута на мікроконтролері STM32mp157f-dk2 та запущена для тестування та подальшого використання;

5. перевірка результатів. Для забезпечення коректності роботи моделі на мікроконтролері, результати виконання були порівняні з результатами на Kaggle, де модель була попередньо протестована.

Ефективне функціонування цієї системи вимагає точного дотримання протоколу I2C, а також правильної інтерпретації команд і відповідей між мікроконтролером та магнітометром. Це дозволяє забезпечити надійну передачу даних та стабільну роботу системи в умовах реального середовища, враховуючи можливі перешкоди та змінні умови.

Інтерфейс I2C (Inter-Integrated Circuit) використовується для зв'язку між мікроконтролером та магнітометричними датчиками. Це двопровідний інтерфейс, що складається з тактової лінії (SCL) і лінії даних (SDA). Контролер (майстер) генерує годинник та ініціює зв'язок з цільовими (підлеглими) вузлами, які отримують команди та передають дані у відповідь [53].

У конкретному сценарії взаємодії, Майстер, на якому запущена модель машинного навчання, ініціює взаємодію з магнітометром шляхом відправлення запитів до певних реєстрів, які містять необхідну інформацію про магнітні поля, їх напрямки, чутливість тощо як показано на рисунку 3.17. Це досягається за допомогою передачі відповідних команд та параметрів через шину I2C. Після отримання запиту від Майстра, магнітометр реагує на ці команди, зчитуючи вміст відповідних реєстрів і передаючи отримані дані назад Майстру через ту саму шину I2C. Цей процес включає зворотну передачу даних про магнітні поля від магнітометра до Майстра.

Ефективне здійснення цього процесу вимагає точного дотримання протоколу спілкування між Майстром та магнітометром, а також встановлення спільних правил і форматів обміну даними. Крім того, обидва пристрої повинні правильно інтерпретувати отримані команди та відповіді для забезпечення коректної передачі та розуміння інформації, що обмінюється через інтерфейс I2C.

Таким чином, інтеграція моделі машинного навчання у мікроконтролер забезпечує точне і ефективне виконання завдань виявлення автомобілів на парковому місці, демонструючи стійкість та адаптивність до різноманітних умов експлуатації.

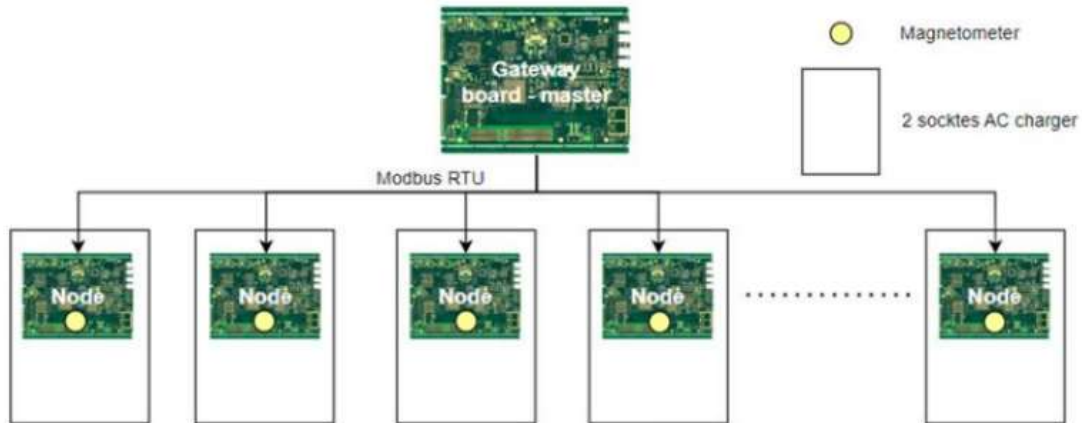


Рисунок 3.17 – Схема зв'язку моделі у STM і магнітометрів

3.7 Перспективи та рекомендації для майбутніх досліджень

Після проведення дослідження було вибрано модель для виявлення транспортних засобів на парковочних місцях з метою запобігання незаконному зайняттю місць для зарядки електромобілів. Однак, для досягнення більшої точності виявлення, необхідно зібрати додаткові дані та провести додаткові експерименти.

Перш за все, необхідно збільшити обсяг доступних даних для навчання моделі, щоб покращити її точність виявлення транспортних засобів на парковочних місцях. Для цього буде важливо провести збір даних на різних парковках з різним рівнем шуму та інших перешкод. Це включає тестування системи в різних середовищах, таких як міські та сільські райони, підземні паркінги та відкриті стоянки.

Автоматичне виявлення активної зарядки: У майбутньому можливо виникне необхідність врахувати стан зарядки автомобіля. Якщо автомобіль фактично заряджається, модель має бути неактивною, оскільки очевидно, що автомобіль знаходиться на місці з метою заряджання. Такий підхід може зменшити непотрібне використання ресурсів для перевірки стану парковочних місць.

Для забезпечення надійності та стабільності роботи системи важливо розг-

лянути можливість введення колибрування моделі, яке дозволить коригувати результати в реальному часі з урахуванням змінних умов. Одним з можливих підходів є використання мінімального перенавчання моделі, що дозволить адаптувати її до нових умов без необхідності повного навчання. Але перенавчання моделі безпосередньо на мікроконтролері, такому як STM32, є складним завданням через обмеженість ресурсів цього пристрою. Тому є рішення мати декілька моделей для різних паркувальних місць на основі схожого статичного магнітного шуму. Статичний магнітний шум може бути спричинений різними факторами, такими як електромагнітні поля, вплив металевих конструкцій, електронне обладнання тощо, і може створювати перешкоди для точного виявлення транспортних засобів за допомогою сенсорів.

3.8 Висновки до розділу 3

В розділі аналізу та навчання моделей машинного навчання для виявлення транспортних засобів на парковочних місцях було проведено дослідження з використанням чотирьох різних алгоритмів: Random Forest, SVM, MLP та CNN.

Результати експерименту показали, що метод Random Forest виявився найефективнішим серед усіх алгоритмів, досягаючи до 98% точності при використанні найефективніших комбінацій параметрів. Точність моделі Random Forest на тестовому наборі даних для лівого паркувального місця склала 98.37%, а для правого - 97.72%. Модель також відзначається високою стійкістю до перенавчання та швидкістю обробки великих обсягів даних.

Модель SVM також показала гарні результати, з точністю на тестовому наборі даних приблизно 92.62%. Для лівого паркового місця точність валідації склала 95.07%, а для другого - 90.32%. Однак, модель Random Forest все ж продемонструвала кращі показники ефективності порівняно з SVM.

Моделі MLP та CNN, хоча можуть бути потенційно корисними для складніших завдань аналізу зображень, у цьому випадку не показали таких високих результатів, як Random Forest і SVM. Точність моделі MLP склала 82.19%, а CNN - 72.25%.

Отже, аналіз результатів показав, що метод Random Forest є найефективнішим для виявлення транспортних засобів на парковочних місцях. Однак, модель SVM також виявилася досить ефективною. Моделі MLP та CNN потребують подальшої оптимізації для досягнення вищих результатів точності.

Інтеграція моделі машинного навчання на мікроконтролер STM32mp157f-dk2 для виявлення автомобілів на парковому місці є критичним етапом у розробці системи моніторингу. У процесі інтеграції була використана модель Random Forest, яка була навчена на платформі Kaggle та експортована у формат ONNX для подальшого використання на мікроконтролері.

Основні кроки процесу включали навчання моделі на платформі Kaggle, експорт моделі у формат ONNX, перенесення моделі на мікроконтролер STM32, налаштування середовища виконання та тестування та налагодження роботи моделі.

Важливою частиною процесу було налагодження взаємодії між мікроконтролером та магнітометром за допомогою інтерфейсу I2C. Цей інтерфейс дозволяв передавати дані про магнітні поля від магнітометра до мікроконтролера, що забезпечило ефективне виявлення наявності або відсутності автомобілів на парковому місці.

У результаті було досягнуто ефективної роботи системи моніторингу, забезпечивши стійкість та адаптивність до різних умов експлуатації. Інтеграція моделі Random Forest на мікроконтролер STM32 дозволила системі ефективно виявляти автомобілі на парковому місці з високою точністю та надійністю.

Таблиця 3.3 – Звіт результатів навчання Random Forest, SVC, MLP та CNN

Метрика/ Модель	Random Forest	SVC	MLP	CNN
Точність валідації	0.982	0.951	0.647	-
Точність тесту	0.984	0.958	0.822	0.722
Час навчання	25.4 сек.	11.3 сек	2.4 хв.	22.21 сек.

Продовження таблиці 3.3

Клас 2				
Точність (precision)	0.97	0.93	0.81	0.72
Повнота (recall)	0.99	0.97	0.71	1.0
F1-показник	0.98	0.95	0.76	0.84
Підтримка (support)	366	366	358	929
Клас 20				
Точність (precision)	0.99	0.98	0.83	0
Повнота (recall)	0.98	0.95	0.89	0
F1-показник	0.99	0.96	0.86	0
Підтримка (support)	556	556	563	358
Середнє по макро				
Точність (precision)	0.98	0.95	0.82	0.36
Повнота (recall)	0.98	0.96	0.80	0.5
F1-показник	0.98	0.96	0.81	0.42
Підтримка (support)	922	922	921	1287
Зважене середнє				
Точність (precision)	0.98	0.96	0.82	0.52
Повнота (recall)	0.98	0.96	0.82	0.72
F1-показник	0.98	0.96	0.82	0.61
Підтримка (support)	922	922	921	1287

ВИСНОВКИ

У роботі було проведено огляд і аналіз підходів до вирішення проблеми ICEing. Під час аналізу досліджувалися різні аспекти, включаючи сучасні датчики для виявлення руху, різні методи обробки сигналів та алгоритми для вирішення проблеми.

Було сформульовано конкретні задачі та розроблено вимоги до системи, яка має вирішувати проблему ICEing. Це включало в себе не лише технічні аспекти, а й вимоги до ефективності, стійкості та зручності використання.

Після постановки задачі та визначення вимог було проведено підготовку даних для моделі машинного навчання. Цей процес включав встановлення тестового стенду для збору даних, розробку бази даних для їх зберігання, обробку сигналу для підготовки даних для моделі, а також маркування даних для навчання моделі з урахуванням вимог та особливостей проблеми ICEing.

У дослідженні запропоновано підхід до ефективного виявлення наявності або відсутності автомобілів на парковому місці. Оскільки планується реалізувати цю функцію на десятках тисяч некаліброваних пристроїв, які знаходяться у різних умовах, включаючи безперервно змінні умови, система має бути досить стійкою до різноманітних сценаріїв та змінних умов – зокрема, що становить змінність шуму та електромагнітних перешкод, що можуть виникати у реальному середовищі. Для цього система моніторингу має використовувати після аналізу результатів тестування алгоритм машинного навчання, який може ефективно адаптуватися до різних умов і забезпечувати точні результати.

Під час тестування було порівняно ефективність різних алгоритмів машинного навчання, де метод Random Forest показав найкращі результати, досягаючи високої точності виявлення транспортних засобів, досягаючи до 98% точності та показуючи 98.37% для лівого і 97.72% для правого паркувальних місць. Також було виявлено, що модель SVM є досить ефективною. Інші методи, такі як MLP та CNN, вимагають подальшої оптимізації для досягнення більш високих результатів. Модель SVM продемонструвала точність 92.62%, з 95.07% для лівого та 90.32% для

правого паркувальних місць, але поступається Random Forest. Моделі MLP і CNN показали точність 82.19% та 72.25% відповідно, що значно нижче за результати Random Forest і SVM.

Таким чином, результати дослідження підтвердили, що метод Random Forest є найефективнішим для вирішення проблеми ICEing, забезпечуючи високу точність виявлення транспортних засобів на парковці. Однак інші алгоритми, такі як SVM, також можуть бути використані з успіхом, хоча вони потребують додаткової оптимізації.

Інтеграція моделі машинного навчання на мікроконтролері STM32 була успішною, що дозволило створити ефективну систему моніторингу на парковочних місцях з високою точністю та надійністю 98% для обох паркувальних місць. Процес інтеграції включав налаштування взаємодії між мікроконтролером та сенсорами, що забезпечило оптимальну роботу системи в реальному часі.

У цілому, це дослідження відкриває нові можливості для вдосконалення систем виявлення паркування та управління зарядкою електромобілів за допомогою передових методів машинного навчання, що може сприяти подальшому розвитку та впровадженню інновацій у цій галузі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Gas Cars Parked at EV Charging Spots: What Are the Rules? Motor and Wheels : web site. URL: <https://motorandwheels.com/gas-cars-park-ev-charging-spots-rules/> (дата звернення 02.03.2024)
2. Gentle Reminder: Stop Parking Your Cars in Electric Charging Stalls. The Drive : web site. URL: <https://www.thedrive.com/article/17222/gentle-reminder-stop-parking-your-cars-in-electric-charging-stalls> (дата звернення 02.03.2024)
3. Public EV Charging Etiquette. Electrify News : web site. URL: <https://electrifynews.com/news/ev-chargers/public-ev-charging-etiquette/> (дата звернення 02.03.2024)
4. Infrastructure for charging electric vehicles: more charging stations but uneven deployment makes travel across the EU complicated. European Court of Auditors. 2021. URL: <https://op.europa.eu/webpub/eca/special-reports/electrical-recharging-5-2021/en/> (дата звернення 02.03.2024)
5. Lu Y., Li Y., Xie D. The application of improved random forest algorithm on the prediction of electric vehicle charging load. Energies. 2018. Vol. 11, No. 11, P. 1-16.
6. Wang D., Ge Y., Cao J., Lin Q., Chen R. Charging load forecasting of electric vehicles based on sparrow search algorithm-improved random forest regression model. The Journal of Engineering. 2023. Vol. 6, No. 6, P. 1-11. URL: <https://ietresearch.onlinelibrary.wiley.com/doi/pdfdirect/10.1049/tje2.12280> (дата звернення: 03.03.2024).
7. Kutila M., Pyykonen P., Ritter W., Sawade O., Schäufele B. Automotive LIDAR sensor development scenarios for harsh weather conditions. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). 2016. Vol. 1–4, No.11, P. 265–270.
8. Abhishek Kumar, Kavita Agrawal. Parking Space Finder Using Image Processing and Machine Learning. International Journal for Multidisciplinary Research (IJFMR). 2023. Vol. 5, No. 4, P. 1-15. URL: https://www.researchgate.net/publication/370059211_Parking_Space_Finder_Using_Image_Processing_and_Machine_Learning (дата звернення: 03.03.2024)

9. Vehicle Detection Sensors. URL: <https://www.bannerengineering.com/us/en/solutions/vehicle-detection/vehicle-detection.html#all> (дата звернення: 04.03.2024).
10. Microwave: web site. URL: <https://en.wikipedia.org/wiki/Microwave#Radar>
11. MEMS : web site. URL: <https://en.wikipedia.org/wiki/MEMS>
12. Sarcevic P., Pletl S., Odry A. Real-Time Vehicle Classification System Using a Single Magnetometer. MDPI. 2022. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9739454/> (дата звернення: 04.03.2024).
13. Amin Kianpishah, Norlia Mustaffa, Pakapan Limtrairut, Pantea Keikhosrokiani. Smart Parking System (SPS) Architecture Using Ultrasonic Detector. International Journal of Software Engineering and Its Applications. 2012. Vol. 6, No. 3, P. 51-58 URL: https://www.researchgate.net/publication/230701092_Smart_Parking_System_SPS_Architecture_Using_Ultrasonic_Detector (дата звернення: 04.03.2024).
14. Kaur H., Malhotra J. A Review of Smart Parking System based on Internet of Things. International Journal of Intelligent Systems and Applications in Engineering. 2018. Vol. 6, No. 4, P. 248–250. URL: <https://doi.org/10.18201/ijisae.2018448450> (дата звернення: 05.03.2024).
15. Sai Pavan V. N. V., Nishanth N. V., Kathirvelan J. Automation of Vehicular Parking Using Loop Detector with single lane traffic: A Design Approach. International Journal of Engineering and Technology (IJET). 2013. Vol. 3, No. 5, P. 2471-2475 URL: https://www.researchgate.net/publication/283141443_Automation_of_Vehicular_Parking_Using_Loop_Detector_with_single_lane_traffic_A_Design_Approach (дата звернення: 05.03.2024).
16. Savari G.F., Krishnasamy V., Sathik J. Internet of things based real-time electric vehicle load forecasting and charging station recommendation. ISA Trans. 2020. Vol. 97, P. 431-447
17. Luo Z.W., Hu Z.C., Song Y.H. Study on plug-in EVs charging load calculating. Automation of Electric Power Systems. 2011. Vol. 35, No. 14, P. 36-42.
18. Abad F., Bendahan R., Wybo S., Bougnoux S., Vestri C., Kakinami T. Parking space detection. In Proceedings of the 14th World Congress on Intelligent Transport

Systems. Beijing, 2007. Vol. 2, P. 9-13. URL: https://www.researchgate.net/publication/228716676_Parking_space_detection (дата звернення: 06.03.2024).

19. Wang D., Ge Y., Cao J., Lin Q., Chen R. Charging load forecasting of electric vehicles based on sparrow search algorithm-improved random forest regression model. *The Journal of Engineering*. 2023. Vol. 6, P. 1–11. URL: https://www.researchgate.net/publication/371640478_Charging_load_forecasting_of_electric_vehicles_based_on_sparrow_search_algorithm-improved_random_forest_regression_model (дата звернення: 07.03.2024)

20. Quintero M. B. R., Carvajal M. C.M., Zuluaga J. F. Real-time parking space occupancy detection system based on image processing and machine learning. In *Proceedings of the International Conference on Computer Science and Software Engineering*. 2019. P. 1-5.

21. Kim D. W., Lee D. K., Kang J.W. A machine learning-based parking occupancy detection system using images captured from a UAV. *Sensors*. 2019. Vol. 19, No. 11, P. 2429.

22. Jung K. T., Kim J. K. Intelligent parking lot management system using image processing and machine learning. *IEEE Access*. 2019. Vol. 7, P. 60017-60028.

23. Raza A. M., Rathore S. B., Kumar S. A smart parking system using machine learning and image processing for IoT-enabled smart cities. *Journal of AmbientIntelligence and HumanizedComputing*. 2019. Vol. 10, No. 2, P. 661-674.

24. Jeong H. B., Park H. K. A deep learning-based vehicle detection and parking space occupancy system for smart parking management. *Sensors*. 2020. Vol. 20, No. 6, P. 1664.

25. Li Q., Liu Y., Li Z. A novel parking space counting system using image processing and machine learning with edge computing. *IEEE Transactions on Industrial Informatics*. 2020. Vol. 16, No. 10, P. 6356-6365.

26. Park S. H., Kim C. S., Ko Y.B. A hybrid approach to parking space occupancy detection using machine learning and image processing. *Sensors*. 2020. Vol. 20, No. 23, P. 6866.

27. Gupta A., Kumar A. An automated parking space counting system based on image processing and machine learning. *International Journal of Computer Science and Network Security*. 2018. Vol. 18, No. 4, P. 192-198.

28. Martins P. H. P., Pinto L. M., Mendonça P. R. S. Parking space occupancy detection using machine learning and image processing: A comparative study. *IEEE Transactions on Intelligent Transportation Systems*. 2021. Vol. 22, No. 7, P. 4522-4533.
29. He X., Wang Y. A review of smart parking systems. *IEEE Transactions on Intelligent Transportation Systems*. 2017. Vol. 18, No. 2, P. 332-347.
30. Перетворення Фур'є. Збірник задач: навч. посіб. для здобувачів ступеня бакалавра за освітніми програмами «Системи технічного захисту інформації», «Системи, технології та математичні методи кібербезпеки» спеціальності 125 Кібербезпека / уклад.: Г. О. Южакова, О. В. Кіреєнко. Київ : КПІ ім. Ігоря Сікорського, 2022. 28 с. URL: <https://ela.kpi.ua/bitstream/123456789/52394/1/Peretvorennia.pdf> (дата звернення: 10.03.2024).
31. Oppenheim A. V., Schafer R. W. *Discrete-time signal processing*. Prentice-Hall. Upper Saddle River, 1975. 893 p.
32. Haykin S., Moher M. *Communication Systems*. Wiley. Hoboken, NJ, 2007. 537 p.
33. Savitzky A., Golay M.J.E. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical chemistry*. 1964. Vol. 36, No. 8, P. 1627-1639.
34. Yang B.-B., Gao W., Li M. On the Robust Splitting Criterion of Random Forest. In *2019 IEEE International Conference on Data Mining (ICDM)*. Beijing, 2019. P. 1-10.
35. Vapnik V. N., Chervonenkis A. Y. *Support-Vector Networks*. Boston: Kluwer Academic Publishers, 1995. 25 p. URL: http://image.diku.dk/imagecanon/material/cortes_vapnik95.pdf (дата звернення: 12.03.2024).
36. Hastie T., Tibshirani R., Friedman J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer, 2009. 764 p.
37. LeCun Y., Bengio Y., Hinton G. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*. 1998. Vol. 86, No. 11, P. 2278-2324. URL: http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf (дата звернення: 13.03.2024).

38. Levenberg A., Divol V., Stein S. Convolutional Neural Networks for Image Processing in Deep Learning. Artificial Intelligence Journal. 2020. Vol. 7, No. 3, P. 45-58. URL: https://www.researchgate.net/publication/220934784_Convolutional_Neural_Networks_for_Image_Processing_An_Application_in_Robot_Vision (дата звернення: 14.03.2024).

39. Scikit-learn Documentation. "Model evaluation: quantifying the quality of predictions." Official Scikit-learn Documentation : web site. URL: https://scikit-learn.org/stable/modules/model_evaluation.html (дата звернення: 15.03.2024).

40. Eaton European Innovation Center (EEIC) : web site. URL: <https://www.eaton.com/cz/cs-cz/company/careers/life-at-eaton/life-in-roztoky-en-gb.html>

41. Video-surveillance. European Data Protection Supervisor : web site. URL: https://www.edps.europa.eu/data-protection/data-protection/reference-library/video-surveillance_en

42. Bc. Jakub Tomeš. Detection of charging space occupancy state / University of Chemistry and Technology Prague, Faculty of Chemical Engineering. Prague, 2023. URL: <https://theses.cz/id/bhns67/?lang=en> (дата звернення: 16.03.2024).

43. BeagleBone Black : web site. URL: <https://www.beagleboard.org/boards/beaglebone-black>

44. A basic guide to I2C : web site. URL: https://www.ti.com/lit/an/sbaa565/sbaa565.pdf?ts=1717930582813&ref_url=https%253A%252F%252Fwww.google.com%252F (дата звернення: 16.03.2024).

45. Circuit python : web site. URL: <https://circuitpython.org/> (дата звернення: 16.03.2024).

46. SparkFun 9DoF IMU Breakout - LSM9DS1 : web site. URL: <https://www.sparkfun.com/products/retired/13284> (дата звернення: 17.03.2024).

47. LV-MaxSonar-EZ. MaxBotix Incorporated. 2015. URL: <https://docs.rs-online.com/26f9/0900766b81534388.pdf> (дата звернення: 17.03.2024).

48. Limits In SQLite : web site. URL: <https://sqlite.org/limits.html> (дата звернення: 19.03.2024).

49. Level up with the largest AI & ML community : web site. URL: <https://www.kaggle.com/> (дата звернення: 11.04.2024).

50. Scikit-learn : web site. URL: https://scikit-learn.org/stable/modules/grid_search.html (дата звернення: 11.04.2024)

51. Єрошенко С.О. Розроблення системи моніторингу для зарядної системи електромобілів з застосуванням методів машинного навчання. Радіоелектроніка та молодь у XXI столітті: Зб. матеріалів 28-го Міжнародного молодіжного форуму. Харків: ХНУРЕ, 2024. Т. 6, С. 317-319. URL: <https://openarchive.nure.ua/handle/document/26345> (дата звернення: 20.04.2024).

52. GitHub : web site. URL: <https://github.com/onnx/onnx> (дата звернення: 20.04.2024)

53. Nicholas Zambetti, Karl Söderby, Jacob Hylén. Inter-Integrated Circuit (I2C) Protocol. 2024. URL: <https://docs.arduino.cc/learn/communication/wire/> (дата звернення: 25.04.2024).

54. Методичні вказівки до організації виконання та захисту атестаційної роботи на здобуття другого (магістерського) рівня вищої освіти спеціальності 122 Комп'ютерні науки освітньо-наукова програма «Системне проектування» / Упорядники: І.В. Гребеннік, Н.І. Калита, І.М. Рябченко, З.А. Імангулова, О.Б. Колесник, М.Ю. Вишняк. Харків: ХНУРЕ, 2021. 54 с.