Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

Галузь знань 12 – Інформаційні технології

Спеціальність 123 – Комп'ютерна інженерія

на тему «Кіберфізична система моніторингу цін на пальне на АЗС з використанням геолокаційних даних»

КвРКІ. 180225.22.01.17 ПЗ

Виконав: студент 2 курсу, група КІ2м-22-1 Бойко М.С.
Підпис Ініціали, прізвище

Керівник доктор техн. наук, професор Яцків В.В.
Науковий ступінь, вчене звання Підпис Ініціали, прізвище

До захисту допускаю:
Зав. кафедри КІІС, д.т.н., проф.
Т.О. Говорущенко
15 05 2024 р.

Хмельницький, 2024

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Факультет Інформаційних технологій
Кафедра Комп'ютерної інженерії та інформаційних систем
Освітній рівень магістр
Галузь знань 12 Інформаційні технології
Спеціальність 123 Комп'ютерна інженерія
Освітня програма освітньо-наукова програма «Комп'ютерна інженерія та програмування»

ЗАТВЕРДЖУЮ
Зав. кафедри Т.О.Говорущенко
"01" 09 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Бойко Микиті Сергійовичу
Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Кіберфізична система моніторингу цін на пальне на АЗС з використанням геолокаційних даних
Керівник проекту (роботи) Яцків В.В., д.т.н., професор
Прізвище, ім'я, по батькові, науковий ступінь, вчене звання
Затверджена наказом ректора університету від 01.01.2024 р. № 1
2. Строк подання студентом проекту (роботи) на кафедру 01.05.2024 р.
3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)
Аналіз предметної області по застосуванню кіберфізичних систем в сфері інтернету транспортних засобів
Моделювання кіберфізичної системи спостереження
Методи кіберфізичної системи спостереження
Кіберфізична система моніторингу цін на пальне на АЗС з використанням геолокаційних даних
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

6. Консультанти розділів кваліфікаційної роботи магістра

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., професор кафедри КІС		
Антиплагіат	Нічепорук А.О., доцент кафедри КІС		

7. Дата видачі завдання «01» 09 2023р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи магістра	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики КвРМ з керівником	01.09.2023	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.10.2023	виконано
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	01.11.2023	виконано
4	Робота над розділом 2 – розробка моделей для вирішення поставленої задачі	01.12.2023	виконано
5	Робота над науковою статтею	01.02.204	виконано
6	Робота над розділом 3 – розробка методів для вирішення поставленої задачі	15.02.2024	виконано
7	Робота над розділом 4 – проектування та розробка ПЗ для вирішення поставленої задачі, експериментальна частина	01.04.204	виконано
8	Оформлення пояснювальної записки згідно вимог	18.04.2024	виконано
9	Попередній захист ДРМ	25.04.2024	виконано
10	Захист ДРМ на засіданні ЕК	До 16.05.2024	

Студент _____ Бойко М.С.
Підпис Ініціали, прізвище

Керівник роботи _____ Яцків В.В.
Підпис Ініціали, прізвище

РЕФЕРАТ

Тема кваліфікаційної роботи магістра: Кіберфізична система моніторингу цін на пальне на АЗС з використанням геолокаційних даних.

Автор роботи: Бойко М.С., студент групи КІ2м-22-1

Керівник роботи: Яцків В.В., доктор технічних наук, професор, завідувач кафедри кібербезпеки.

Пояснювальна записка: 81 с., 26 рис., 13 табл., 3 дод., 84 джерел.

КІБЕРФІЗИЧНА СИСТЕМА, СИСТЕМА СПОСТЕРЕЖЕННЯ, ІНТЕРНЕТ ТРАНСПОРТНИХ ЗАСОБІВ, ОПТИЧНИЙ АНАЛІЗ, АНАЛІТИКА ВЕЛИКИХ ДАНИХ.

Об’єктом дослідження є процеси моніторингу цін на пальне на АЗС з використанням геолокаційних даних.

Предметом дослідження є методи та алгоритми моніторингу цін на пальне на АЗС з використанням геолокаційних даних.

Метою кваліфікаційної роботи магістра є розробка та дослідження кіберфізичної системи моніторингу цін на автозаправних станціях.

Для розв’язання поставлених задач використовувалися методи: комунікації між підсистемами, розпізнавання образів та тексту, геолокаційного позиціонування, аналізу великих обсягів даних та розроблено автоматичного прийняття рішення підсистемою кіберфізичної системи.

Наукова новизна отриманих результатів:

1) вперше запропоновано та розроблено кіберфізичну систему спостереження в середовищі з обмеженою швидкістю інтернету, основна ціль якого – це моніторинг цін на автозаправних станціях;

2) вперше розроблено метод автоматичного прийняття рішення периферійним вузлом, який забезпечує безвідмовну роботу підсистеми кіберфізичної системи спостереження.

На основі проведених досліджень описана технічна архітектура і компоненти програмного забезпечення двох підсистем: периферійного вузла та серверної підсистеми.

Практична значущість отриманих результатів полягає в отриманих практичних результатах дослідження підсистем кіберфізичної системи моніторингу цін на пальне на АЗС з використанням геолокаційних даних, які надають вичерпний аналіз результатів використання методів підсистем та результати їх роботи.

У першому розділі проведений огляд застосувань кіберфізичних систем в різних сферах діяльності та виокремлено кіберфізичні системи в сфері інтернету транспортного засобу як основний орієнтир в подальшій кваліфікаційній роботі. Проведено порівняльний аналіз переваг та недоліків існуючих рішень, в яких розглядалися наукові роботи з сфери інтернету транспортних засобів та кіберфізичних систем спостереження, на основі яких визначено методологічні підходи та здійснена постановка задачі.

У другому розділі розроблено модель кіберфізичної системи спостереження, в якій її було розділено на три підсистеми та розписано складові кожної з підсистем в окремо розроблених моделях. Також в даному розділі було розроблено модель основного процесу та часову модель КФСС.

У третьому розділі визначено необхідну методологію для реалізації кіберфізичної системи спостереження, а саме методи комунікації між підсистемами; методи розпізнавання образів і тексту; методи визначення геолокаційного позиціонування; методи аналізу великих даних. В даному розділі було сформовано новий метод автоматичного прийняття рішення підсистемою КФСС – периферійним вузлом, необхідність якого зумовлена технічними обмеженнями визначених методів комунікації між підсистемами.

У четвертому розділі розглянуто технічну архітектуру двох підсистем кіберфізичної системи спостереження, кожну з них реалізовано та досліджено на практиці, за результатами якої підтверджено дві гіпотези, спростовано дві гіпотези та висунуто одну нову гіпотезу.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ 6

ВСТУП 7

1 ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖУВАНОЇ ПРОБЛЕМИ 9

1.1 Аналіз кіберфізичних систем та їх застосування 9

1.1.1 Застосування КФС в індустрії четвертого покоління 10

1.1.2 Вимоги до структури КФС 11

1.1.3 Застосунки з використанням КФС 12

1.2 Виявлення наявних проблем і завдань кіберфізичних систем в IoV 13

1.3 Порівняльний аналіз переваг та недоліків існуючих рішень 14

1.4 Методологічні підходи до вирішення задачі за темою дослідження 17

1.5 Постановка задачі 18

2 МОДЕЛЬ КІБЕРФІЗИЧНОЇ СИСТЕМИ СПОСТЕРЕЖЕННЯ 20

2.1 Опис загальної моделі КФСС 21

2.2 Модель вузлової підсистеми КФСС 24

2.2.1 Позиціонування і математична модель апаратної частини 24

2.2.2 Програмна складова периферійного вузла 26

2.2.3 Взаємозв’язок апаратної та програмної частини 28

2.3 Модель клієнтської та серверної підсистеми КФСС 29

2.3.1 Структура підсистем 29

2.3.2 Функціональні вимоги 33

2.3.3 Математична модель підсистем 36

2.4 Модель основного процесу 37

2.5 Часова модель КФСС 38

2.6 Висновки 41
3 МЕТОДОЛОГІЯ ДОСЛІДЖЕННЯ КФСС 42
3.1 Методи комунікації між підсистемами КФСС 42
3.2 Методи розпізнавання образів та тексту 45
3.2.1 Алгоритми виявлення об'єктів на зображенні 46
3.2.2 Алгоритми оптичного розпізнавання символів 51
3.2.3 Аналіз сервісів оптичного розпізнавання символів 52
3.3 Методи геолокаційного позиціонування 56
3.4 Метод автоматичного прийняття рішення підсистемою КФСС – периферійним вузлом 57
3.5 Методи аналізу великих обсягів даних та виявлення аномальних значень у наборі даних 59
3.6 Висновки 64
4 КІБЕРФІЗИЧНА СИСТЕМА МОНІТОРИНГУ ЦІН НА ПАЛЬНЕ НА АЗС З ВИКОРИСТАННЯМ ГЕОЛОКАЦІЙНИХ ДАНИХ 66
4.1 Периферійний вузол 66
4.1.1 Технічна архітектура 66
4.1.2 Тренування нейронної мережі YOLO по розпізнаванню образів 72
4.1.3 Результати роботи підсистеми 74
4.2 Серверна підсистема 76
4.2.1 Технічна архітектура 76
4.2.2 Результати роботи підсистеми 78
4.3 Висновки 82
ВИСНОВКИ 84
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ 86

ДОДАТОК А Копії публікації 95

ДОДАТОК Б Лістинг коду..... 106

ДОДАТОК В Презентація до захисту кваліфікаційної роботи 112

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

IoT – Internet of Things

IoV – Internet of Vehicle

DSRC – Dedicated Short-Range Communications

DBSCAN – Density-Based Spatial Clustering of Applications with Noise

АЗС – Автозаправна станція

ГІС – Геоінформаційна система

КФС – Кіберфізична система

КФСС – Кіберфізична система спостереження

ВСТУП

В сучасному світі зростаюче значення набуває розвиток інноваційних технологій, спрямованих на оптимізацію та удосконалення різноманітних аспектів економічної діяльності. Однією з ключових сфер, що вимагає постійного вдосконалення, є ринок нафтопродуктів, особливо сегмент автозаправних станцій (АЗС). З огляду на постійні зміни цін на пальне, а також варіативність їх рівня в різних регіонах, виникає необхідність у впровадженні ефективних інструментів моніторингу цін, які б дозволили аналізувати та прогнозувати динаміку цінової політики на ринку нафтопродуктів.

У цьому контексті виникає актуальна проблема розробки та впровадження кіберфізичних систем моніторингу цін на пальне на АЗС з використанням геолокаційних даних. Кіберфізичні системи, поєднуючи в собі кібернетичні та фізичні компоненти, надають можливість в реальному часі збирати, аналізувати та використовувати дані для прийняття рішень у різних галузях. Використання геолокаційних даних дозволить не лише відстежувати ціни на пальне, а й аналізувати їхню динаміку в залежності від географічного розташування АЗС, транспортних маршрутів та інших факторів.

Важливо зазначити, що інтернет речей (IoT) в автомобільній промисловості, частіше називається інтернет транспортних засобів (IoV), стає все більш важливим елементом інфраструктури розумних міст. IoV являє собою можливість підключення автомобілів та їхніх складових до мережі Інтернет, що відкриває широкі можливості для збору, аналізу та обміну даними.

У зв'язку з цим, актуальність роботи ще більше підвищується, оскільки система моніторингу цін на пальне може ефективно інтегруватися з інфраструктурою IoV. Це відкриває можливості для збору додаткових даних, таких як статистика про подорожі, популярні маршрути та інші параметри, що можуть бути корисними для аналізу цінової динаміки та попиту на пальне. Такий підхід дозволяє більш точно прогнозувати зміни в ціновій політиці на ринку палива, що стає важливим фактором для управління бізнесом на АЗС та для споживачів.

Тому, розробка кіберфізичної системи моніторингу цін на пальне з використанням геолокаційних даних в контексті IoV має значення як для підприємств пального бізнесу, так і для автовласників, а також в цілому для економіки та сталого розвитку транспортної галузі.

Метою кваліфікаційної роботи магістра є дослідити можливість реалізації кіберфізичної системи спостереження в сфері інтернету транспортних засобів основною цілю якої є моніторинг цін на автозаправних станціях.

Для розв'язання поставлених задач використовувалися методи: комунікації між підсистемами, розпізнавання образів та тексту, геолокаційного позиціонування, аналізу великих обсягів даних та розроблено автоматичного прийняття рішення підсистемою кіберфізичної системи.

Наукова новизна отриманих результатів:

1) вперше запропоновано та розроблено кіберфізичну систему спостереження в середовищі з обмеженою швидкістю інтернету, основна ціль якого – це моніторинг корисної інформації;

2) вперше розроблено метод автоматичного прийняття рішення периферійним вузлом, який забезпечує безвідмовну роботу підсистеми кіберфізичної системи спостереження.

На основі проведених досліджень описана технічна архітектура і компоненти програмного забезпечення двох підсистем: периферійного вузла та серверної підсистеми.

Практична значущість отриманих результатів полягає в отриманих практичних результатах дослідження підсистем кіберфізичної системи моніторингу цін на пальне на АЗС з використанням геолокаційних даних, які надають вичерпний аналіз результатів використання методів підсистем та результати їх роботи.

1 ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖУВАНОЇ ПРОБЛЕМИ

1.1 Аналіз кіберфізичних систем та їх застосування

Кіберфізична система(КФС) характеризується як революційна технологія для керування пов'язаними системами між їх обчислювальними можливостями та фізичними активами. КФС – це мережа взаємопов'язаних компонентів інформаційних технологій, яка використовується для керування фізичними предметами, такими як механічні та електронні пристрої. Інтернет використовується як інформаційна інфраструктура для з'єднання. КФС складається з двох основних функціональних елементів:

1. Продвинуті шлюзи, які допомагають забезпечити отримання даних у реальному часі з фізичного світу або ж з кіберсередовища.

2. Інтелектуальні обробка даних, комп'ютерне моделювання та аналіз даних.

Декілька технологій нерозривно пов'язані з КФС, наприклад: мережі датчиків, Інтернет речей (IoT), бездротові та хмарні обчислення. Бездротові мережі вважаються важливою частиною КФС. Інтернет – є критичним компонентом для оптимізації продуктивності КФС. До параметрів технології інтернету включаються такі аспекти:

– інтернет речей [1], передбачає використання IP-адрес для спілкування з розумними системами. Це дозволяє призначати будь-якому фізичному пристрою свою IP-адресу;

– інтернет послуг [2], включає нові моделі комунікації, такі як ті, що пропонуються сервіс-орієнтованим дизайном (SOA) та технологією REST;

– інтернет даних [03], дозволяє ефективну передачу та збереження великих обсягів даних, а також створення інноваційних аналітичних методів для аналізу великих обсягів даних.

– інтернет ресурсів, який спрямований на забезпечення доступу до різноманітних онлайн-ресурсів та сервісів, таких як веб-сторінки, відео, аудіо та інші мультимедійні контенти;

– інтернет безпеки, що охоплює заходи та технології для захисту мережі, даних та пристроїв від шкідливих атак та несанкціонованого доступу;

– інтернет доступності, спрямований на забезпечення рівного доступу до інтернет-ресурсів для всіх користувачів незалежно від їхніх фізичних можливостей та місця проживання.

1.1.1 Застосування КФС в індустрії четвертого покоління

КФС являє собою високоінтегровану систему фізичних компонентів, що містять датчики, виконавчі механізми та різноманітне обладнання, разом з кібер-компонентами, які мають всюдисущу обробку та ефективний зв'язок, що є швидко розвиваючоюся темою дослідження [04]. Вирішення вимог, викликів та можливостей у різних промислових секторах може допомогти прискорити дослідження КФС. Основний акцент складається на створенні нових методологій системної науки та інженерії для проектування систем високої надійності, які є гнучкими, синергетичними та взаємопов'язаними на всіх рівнях. Витрати промислових потужностей на технології і дослідження КФС на сьогоднішній день є доволі значними, але вони зосереджені на короткострокових і швидко прибуткових рішеннях [05]. КФС, до яких входять: транспортні системи, електропостачальні системи, системи розподілу води/газу та автономні заводи, вважаються найбільш потужними промисловими системами з точки зору інженерії. Широкий спектр промислових роботів з інерційним навігаційним пристроєм або іншими датчиками програмується для руху по заздалегідь визначеному маршруту для виконання виробничих завдань разом [06, 07]. Тісна координація кіберфізичних аспектів в цих системах забезпечує вищу свободу, продуктивність, використовуваність, безпеку та гнучкість. Крім того, промислові КФС розглядаються як ключовий компонент Четвертої промислової революції, і викладені значні зусилля для продемонстрації їх важливості [08]. Промислові КФС є масштабними, глобально розподіленими, федеративними, спільними та життєво важливими системами з великою кількістю інтегрованих датчиків та виконавчих

механізмів, які з'єднані для надання обстеження в реальному часі та замкнутого керування. Крім того, мережі датчиків та розподілені мережі керування відіграють значну роль у виконанні промислових КФС. Іншими словами, в широко розгорнутій архітектурі КФС ці дві мережі часто є необхідними. Мережі датчиків зазвичай впроваджуються в середині або навколо заводів для збору великої кількості важливої інформації для отримання точних сприйняттів фізичних об'єктів. Використовуючи цю інформацію, виконавчі механізми можуть реагувати в реальному часі на зміни фізичних об'єктів [09]. Отже, при злитті кібер та фізичних світів виникає замкнуте зв'язування як між комунікаціями між датчиками та виконавчими механізмами, так і управлінням об'єктами за допомогою виконавчих механізмів. Багато промислових датчиків та виконавчих механізмів з деякими можливостями комунікації та обробки даних стали доступними в останні роки, завдяки зусиллям дослідницьких інститутів та підприємств у всьому світі.

1.1.2 Вимоги до структури КФС

Основною метою КФС є надання нових функцій фізичним системам шляхом поєднання обчислення та зв'язку з фізичними процесами. КФС повинна надавати в реальному часі, динамічне керування та інформаційні послуги для складних процесів через міцну співпрацю між трьома параметрами: обчисленням, зв'язком та керуванням [10]. КФС має покладати особливий акцент на великі можливості зв'язку та обчислень кіберсвіту, які також можуть покращити точність та ефективність фізичного світу [11]. Крім того, незалежно від того, чи йдеться про трьохрівневий, п'ятирівневий чи орієнтований на послуги дизайн, всі описані академічними дослідниками дизайни КФС сконцентровані на керуванні, а не на відображенні. Завдання КФС повинні здійснюватися за допомогою взаємного відображення, реального часу зв'язку та ефективної співпраці між кібер та фізичними світами [12]. Обчислювальна система повинна вплинути на більше одного фізичного об'єкта. Наприклад, структура може складатися з декількох

апаратних елементів. Тому зв'язок взаємодії між кібер та фізичними світами КФС є відношенням один-до-багатьох, а не один-до-одного.

1.1.3 Застосунки з використанням КФС

Технології на основі КФС перебувають у початковому етапі [13]. Тому в промисловості 4.0 застосування CPS постійно змінюється та розширюється. Багато застосувань на основі CPS були розроблені або впроваджені в широкому спектрі галузей, таких як системи транспортування повітряних суден [14], військовий нагляд [15], хімічне виробництво [16], енергетика [17], сільське господарство (постачання продуктів харчування) [18], охорона здоров'я [19], освіта [20], промислова автоматизація [21], виробництво [22], мобільні пристрої [23], робототехніка [24], транспорт [25] та автотранспорт [26]. Дослідники та практики можуть змушені здійснити компроміс між своїми цілями для створення комбінації переваг і витрат, на основі бажаного CPS та промисловості. Всі вищезгадані галузі графічно відображені на рисунку1.1.

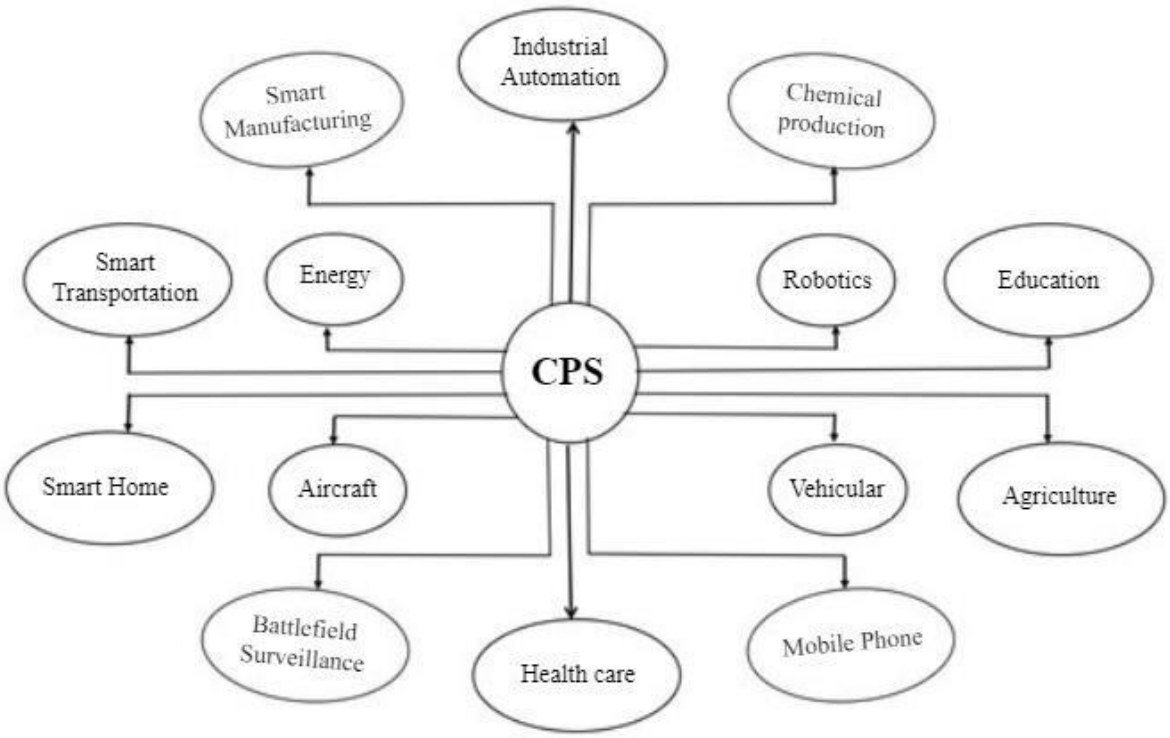


Рисунок 1.1 – Спектр галузей в яких застосовуються КФС [05]

1.2 Виявлення наявних проблем і завдань кіберфізичних систем в IoV

Проведений аналіз кіберфізичних систем та їх застосування в світі надає можливість виявлення наявних проблем і завдань у цілій сфері. Серед виявлених проблем і завдань можна виділити декілька ключових, серед яких: забезпечення безпеки, складність інтеграції, ефективність роботи, складність систем, енергоефективність та масштабованість.

Розглянемо ряд ключових аспектів, що потребують окремої уваги в сфері інтернету транспортних засобів, оскільки КФС яка пропонується даною кваліфікаційною роботою напряму зв’язана з транспортними засобами.

Проблема безпеки полягає в тому, що підключення через мережу пов’язане з ризиками, починаючи від проблем конфіденційності та закінчуючи вразливістю мережі. Усередині транспортних засобів багато датчиків і електронних блоків керування, які з’єднані через спеціальні мережеві протоколи, відомі як внутрішньоавтомобільний зв’язок, щоб допомогти водієві виконувати такі завдання, як автономне водіння(автопілот). Зв’язок за допомогою різних протоколів призводить до відкритості безпеки для зловмисників, щоб порушити нормальне функціонування датчиків і блоків управління, що іноді має шкідливі наслідки [27]. Крім того, в IoV існує активний зв’язок між транспортними засобами та дорожніми інфраструктурами (по типу світлофорів тощо). Мотивуючись проблемою безпеки, багато дослідників почали працювати в даній сфері і створили системи виявлення вторгнень (IDS) [28, 29, 30] для виявлення зловмисної поведінки, яка загрожує конфіденційності, цілісності, доступності та автентичності в інтернеті транспортних засобів.

Складність інтеграції полягає в тому, що транспортні засоби, такі як автомобілі, потребують спеціально адаптованих датчиків, актуаторів та інших пристроїв, які можуть працювати в умовах дорожнього руху. Також інтегровані КФС повинні взаємодіяти з існуючими системами автомобілів та інших

транспортних засобів, такими як системи безпеки, мережі зв'язку та системи управління двигуном [31].

Проблема масштабованості в контексті інтеграції КФС в IoV стосується здатності системи працювати ефективно при зростанні обсягів даних та збільшенні кількості підключених пристроїв. Зі збільшенням кількості підключених пристроїв та обсягів зібраних даних зростає навантаження на систему обробки даних. Це може вимагати розробки спеціалізованих алгоритмів обробки даних та використання потужних обчислювальних ресурсів[32]. Інтенсивний обмін даними між підключеними пристроями та центральними серверами може призвести до заторів в мережі. Для запобігання цьому потрібно розробляти ефективні алгоритми управління трафіком та ресурсами мережі[33, 34]. Архітектура системи повинна бути гнучкою та легко розширюваною, щоб забезпечити можливість додавання нових пристроїв та функціональностей без перерв у роботі системи.

Отже виявлення наявних проблем демонструє необхідність в проведені порівняльного аналізу переваг та недоліків існуючих рішень висвітлених в даному розділі оскільки для ефективного розв'язання цих проблем важливо виконати глибокий аналіз методологічних підходів для інтеграції КФС цілю якої є – моніторингу цін на пальне. Також виявлені проблеми потребують розробки нових стратегії збору, аналізу та використання геолокаційних даних.

1.3 Порівняльний аналіз переваг та недоліків існуючих рішень

Порівняльний аналіз переваг та недоліків існуючих рішень у сфері кіберфізичних систем моніторингу з використанням геолокаційних даних виявляє різноманітні підходи та технології, які застосовуються в цій галузі.

Одним із існуючих рішень є можливість відстежування цінової динаміки на різних АЗС у реальному часі. Це надає можливість оперативно реагувати на зміни цін та адаптувати свою стратегію ціноутворення для максимізації прибутку. Проте дані рішення присутні на глобальних або регіональних АЗС які загально відомі і при цьому не включають локальні АЗС.

Також деякі існуючі рішення можуть використовувати аналіз геолокаційних даних для виявлення тенденцій споживання пального [35] та прогнозування попиту у певних регіонах. Це дозволяє оптимізувати запаси пального та забезпечувати краще обслуговування клієнтів.

Проте існують і недоліки. Деякі системи можуть бути недостатньо точними або актуальними у зборі та аналізі даних, що може призвести до неправильних стратегій управління цінами. Також існує проблема захисту особистої інформації користувачів, оскільки деякі рішення можуть використовувати геолокаційні дані без належного забезпечення конфіденційності та безпеки.

Оскільки наша КФС для моніторингу буде встановлюватись в автомобіль, то вона буде тісно пов’язана з інтернетом транспортних засобів. Тому при аналізі існуючих рішень ми розглянемо ще переваги та недолікіи по реалізації мережі для моніторингу цін на АЗС в мережі IoV. Таким чином ми можемо визначити вже існуючі в даній сфері наукові роботи які висвітлюють позитивні та негативні аспекти.

З усіх робіт по IoV у [36] найкраще описано протоколи маршрутизації. В ній автори розглянули попередні дослідження у цій області включають класичні протоколи маршрутизації, які походять з досліджень мереж бездротового спілкування, таких як Destination-Sequenced Distance-Vector (DSDV) [37], Dynamic Source Routing (DSR) [38], і Ad-hoc On Demand Distance Vector (AODV) [39]. У зв'язку з унікальними властивостями транспортних засобів, деякі дослідники пропонували використовувати географічно-орієнтовані протоколи маршрутизації, такі як Greedy Perimeter Coordinator Routing (GPCR) [40] і Greedy Perimeter Stateless Routing (GPSR) [41]. Ці протоколи враховують географічне розташування транспортних засобів для прийняття рішень щодо маршрутизації даних.

Одним з відомих рішень по захисту особистої інформації користувачів запропоновано в [42], де автори описують новий кіберфізичний механізм який базується на натовпі розумний людей (SPC) та на натовпі розумних транспортних засобів (SVC), в статті описано систему кіберфізичного спостереження та вказано на її обмеження. Автори також описали концепцію безпечного IoV яку можна

побачити на рисунку 1.2. Мінусом даної роботи можна відзначити, що в ній розглядається концепт, а не реалізація системи.

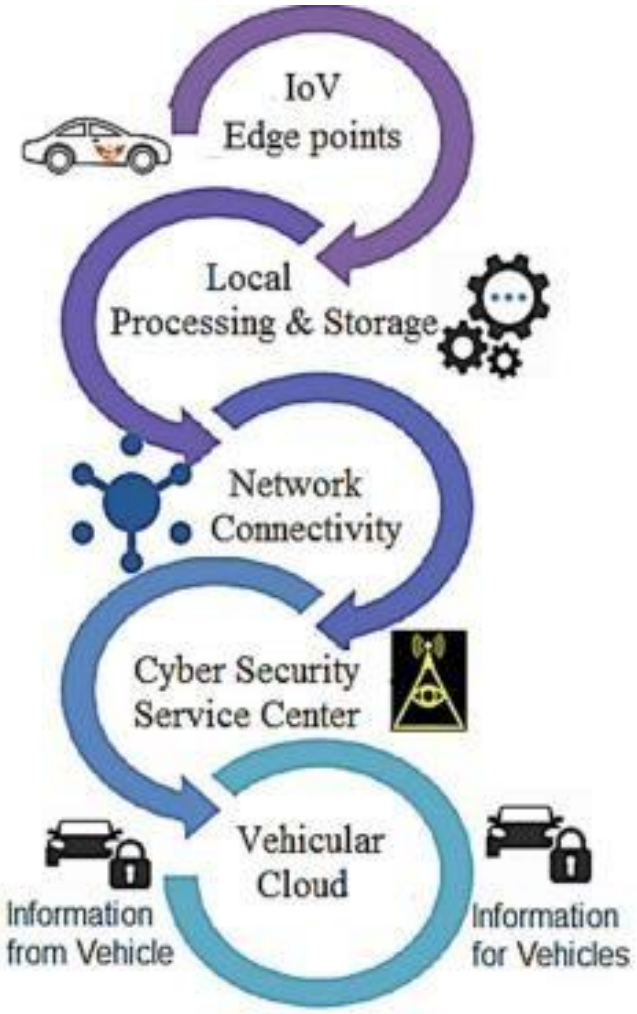


Рисунок 1.2 – Концепція безпечного Інтернету транспортних засобів. [42]

Також однією зі схожих робіт є [43] в якій був запропонований і застосований метод вивантаження завдань для відеоспостереження в розподіленому обчислювальному середовищі з використанням IoV. Автори сфокусувалися на скороченні часових витрат, збережені балансу навантаження вузлів розподіленого обчислювання та на підвищені захисту приватності. Проте мінусом даної роботи є те, що в ній розглядається стаціонарна система стеження, а не динамічна.

Також варто відзначити роботу [44] в якій розглянуто інтеграцію КФС в сферу військової безпеки а саме наглядом за полем бою. Автор спрямував основний

фокус роботи на напрямок оптимізації на рівні компілятора, проте такий підхід більш зосереджений на роботі одного вузла а не всій мережі спостереження.

Отже, порівняльний аналіз існуючих рішень показує, що необхідно розвивати нові технології та методи для вдосконалення моніторингу, які б забезпечували точність, актуальність та захищеність геолокаційних даних. У нашій досліджуваній області інтернету речей, важливо враховувати не лише існуючі позитивні та негативні аспекти, а й перспективи застосування. Аналізуючи роботи, що висвітлюють переваги та недоліки, а також покликані розв'язати проблеми у сфері моніторингу цін на АЗС в мережі IoV, ми можемо визначити перспективні напрямки подальших досліджень та розробок. Особливу увагу слід звернути на захист особистої інформації користувачів та вдосконалення точності та актуальності збору даних для ефективного управління ціноутворенням на АЗС.

1.4 Методологічні підходи до вирішення задачі за темою дослідження

Методологічні підходи до вирішення задачі за темою дослідження "Кіберфізична система моніторингу цін на пальне на АЗС з використанням геолокаційних даних" можуть включати наступні аспекти:

Застосування технологій інтернету речей. Використання датчиків та пристроїв, які забезпечують збір геолокаційних даних про розташування АЗС та ціни на пальне в реальному часі. В цю категорію також входять технології IoV по типу протоколів зв’язку і систем відеореєстрації.

Аналіз даних і штучний інтелект [45-49]. Використання алгоритмів машинного навчання для аналізу великих обсягів геолокаційних даних та прогнозування цінової динаміки на ринку пального.

Геоінформаційні системи (ГІС). Використання ГІС для візуалізації та аналізу геолокаційних даних, створення карт та географічних аналітичних звітів [50, 51].

Методи оптимізації та прийняття рішень. Використання математичних моделей оптимізації для вирішення проблем планування маршрутів доставки пального та оптимального розміщення АЗС [52-55].

Методи периферійних обчислень передбачають використання обчислювальних ресурсів, що знаходяться на периферійних пристроях або в інших вузлах мережі [56-58]. Застосування таких методів може допомогти розподілити обчислювальне навантаження та оптимізувати використання ресурсів для аналізу геолокаційних даних та моніторингу цінової динаміки на різних АЗС. Може включати в себе використання вбудованих обчислювальних можливостей автомобільних систем, смартфонів або інших периферійних пристроїв для збору та обробки даних у реальному часі.

Також, методи вивантаження відео [59-61] можуть бути важливим аспектом у розробці кіберфізичних систем моніторингу на АЗС. Ці методи передбачають ефективне управління потоками відеоданих, включаючи їх збір, трансляцію, збереження та аналіз. Вони можуть включати в себе різноманітні технології стиснення відео, стратегії передачі даних та алгоритми оптимізації для забезпечення найкращої якості та ефективності відеомоніторингу на АЗС. Такі методи можуть бути спрямовані на зменшення обсягу передачі даних, збільшення швидкості передачі та покращення точності та достовірності аналізу відеоінформації.

Ці методологічні підходи в сукупності можуть забезпечити комплексне розуміння та ефективне розв'язання завдань, пов'язаних з розробкою кіберфізичної системи моніторингу цін на пальне на АЗС з використанням геолокаційних даних.

1.5 Постановка задачі

Постановка задачі дослідження "Кіберфізична система моніторингу цін на пальне на АЗС з використанням геолокаційних даних" включає наступні основні завдання:

1. Згідно виявлених проблем та аналізу існуючих рішень необхідно розробити модель кіберфізичної системи спостереження (КФСС), яка буде враховувати існуючі проблеми та функціональні вимоги до її підсистем. Розробити модель основного процесу КФС та її часову модель.

2. Вивчення та детальний аналіз методів необхідних для функціонування КФСС. Допускається розгляд відомих алгоритмів та програмного забезпечення які нададуть можливість впровадити розроблену КФС для моніторингу в реальне середовище.

3. Розробка основних функціональних можливостей КФСС, які надають можливість моніторингу цін на АЗС та комбінація всіх складових в одну систему.

4. Тестування та дослідження отриманих результатів роботи КФСС та її підсистем. Спростування, підтвердження та висунення гіпотез згідно результатів.

5. Формулювання висновків з отриманих результатів, створення рекомендації щодо подальшого вдосконалення та розвитку КФСС, включаючи можливі шляхи оптимізації та розширення її функціональності, а також визначення потенційних перешкод та способів їх подолання для ефективної реалізації подібних системи на практиці.

2 МОДЕЛЬ КІБЕРФІЗИЧНОЇ СИСТЕМИ СПОСТЕРЕЖЕННЯ

Метою кіберфізичної системи моніторингу цін на пальне з використанням геолокаційних даних є створення ефективного і точного інструменту для аналізу та прогнозування цін на пальне на ринках з урахуванням географічного положення користувачів. Система має за мету забезпечити:

1. Аналіз цін на пальне. Забезпечення можливості користувачам системи отримувати актуальну інформацію про ціни на пальне в різних географічних регіонах.

2. Прогнозування цін. Надання користувачам інструментів для прогнозування та аналізу динаміки цін на пальне з використанням історичних даних та аналізу ринкових тенденцій.

3. Персоналізований підхід. Забезпечення можливості персоналізації інформації про ціни на пальне залежно від географічного положення користувача та їхніх індивідуальних потреб.

4. Підтримка прийняття рішень. Надання користувачам інструментів для прийняття обґрунтованих рішень щодо закупівлі пального на основі доступної інформації про ринок.

5. Швидкий доступ до інформації. Забезпечення швидкого та безперервного доступу до актуальної інформації про ціни на пальне для користувачів системи.

6. Безпека даних. Забезпечення певного рівня захисту конфіденційності та безпеки даних користувачів системи моніторингу цін на пальне.

Враховуючи описані цілі КФСС, можемо розглянути прогнозовану класифікацію за різними категоріями. До цих категорій відносяться:

– тип даних для моніторингу цін на пальне, що включає ринкові ціни на пальне та геолокаційні дані, такі як місцезнаходження станцій пального;

– технології та методи моніторингу, які передбачають використання сучасних технологій та алгоритмів для аналізу цін та геолокаційних даних, а також застосування машинного навчання для прогнозування та аналізу змін цін на пальне;

– інтеграція з іншими джерелами даних, що включає взаємодію з додатками та сервісами для збору додаткових даних про ціни на пальне та геолокаційні дані;

– функціональні можливості, які охоплюють моніторинг цін на пальне в реальному часі, аналіз їх динаміки та прогнозування на основі історичних даних та геолокаційних відомостей, а також генерацію звітів та аналітичних даних про зміни в цінах на пальне в різних регіонах;

– рівень доступності та підтримка, що охоплюють доступність сервісу для користувачів в різних країнах та регіонах, а також підтримку мобільних додатків та веб-інтерфейсу для зручного доступу користувачів до інформації про ціни на пальне;

– забезпечення конфіденційності та безпеки даних, що передбачає застосування заходів безпеки для захисту особистих та фінансових даних користувачів, а також шифрування та захист збережених даних про ціни на пальне та геолокаційні відомості.

2.1 Опис загальної моделі КФСС

Розробку кіберфізичної системи спостереження(КФСС) почнемо з опису основних компонентів та функціональних блоків даної системи. Дана система як і будь-яка з подібних буде включати в себе різноманітні сенсори, засоби збору та обробки даних, модулі зв’язку, аналітичні інструменти та інші компоненти. Ось базовий перелік необхідних складових нашої системи:

1) сенсори – пристрої для збору даних з реального світу. В нашому випадку це будуть камери (відеореєстратори) прикріплені в транспортному засобі;

2) засоби збору та накопичення даних – компоненти призначені для отримання, обробки та збереження даних, які надходять від сенсорів;

3) модулі зв'язку – включають в себе засоби передачі даних між різними частинами системи, в нашому випадку вони можуть бути тільки бездротовими (Wi-Fi, Bluetooth);

4) обчислювальні ресурси – компоненти, які виконуватимуть обчислення та обробку даних. Це може бути хмара, локальні сервери, вбудовані системи, які виконуватимуть аналіз та виявлення паттернів у даних;

5) аналітичні інструменти – інструменти використовуються для аналізу та інтерпретації даних, щоб виділити корисну інформацію та виявити аномалії або інші цікаві події;

6) інтерфейс користувача – це засоби взаємодії з системою для користувача. Це може бути веб-інтерфейс, мобільний додаток або інше програмне забезпечення;

7) система управління – це компоненти, що відповідають за керування різними аспектами системи, такі як збір та передача даних, керування енергоефективністю тощо.

Таким чином дані складові можна розділити на три групи підсистем з яких скрадатиметься наша КФСС: вузли, сервери, клієнти. Загальний абстрактний виглядом КФСС можна побачити на рисунку 2.1.

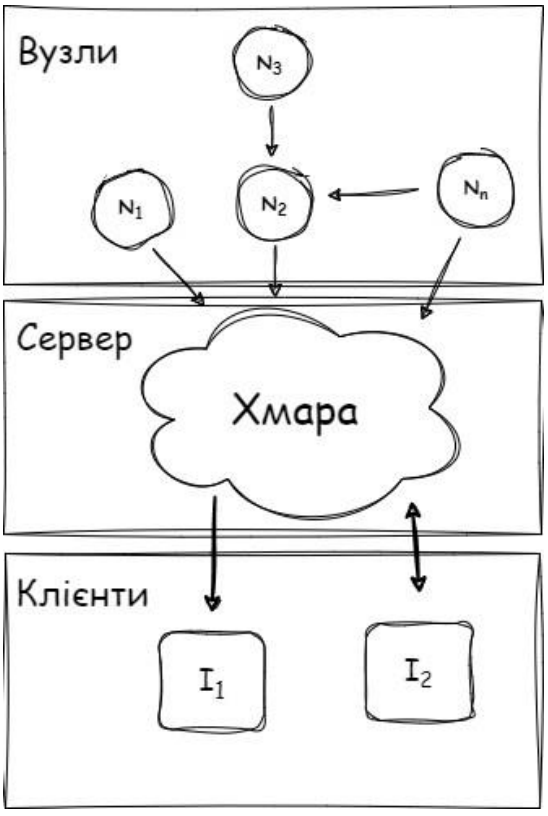


Рисунок 2.1 – Загальна абстрактна модель КФСС

Розподілимо попередньо описані складові до кожної групи підсистем.

Вузол або крайовий пристрій повинен місити в собі:

– сенсори, а саме камери для збору зображень;

– засоби збору даних, а саме: модуль збору даних з камери та передачі їх на сервер для подальшої обробки;

– обчислювальні ресурси: модуль обробки зображення з використанням нейронної мережі.

Сервер або кластер серверів міститиме в собі:

– засоби збору даних: приймає дані від вузла та зберігає їх;

– обчислювальні ресурси: виконує більш складні обчислення, такі як обробка зображень, аналіз даних, прогнозування тощо;

– аналітичні інструменти: виявлення паттернів, аномалій або інших цікавих подій у зібраних даних.

Клієнти або інтерфейси для користувачів та аналітиків або адміністраторів міститиме в собі:

– інтерфейс користувача: веб-інтерфейс або мобільний додаток, який дозволяє користувачам взаємодіяти з системою, переглядати результати аналізу даних, отримувати сповіщення тощо;

– система управління: керування різними аспектами системи, такими як збір та передача даних, керування енергоефективністю тощо, здійснюватиметься через інтерфейс користувача.

Згідно описаних груп КФСС виділимо можливі підсистеми зі структури КФС і опишемо їх в якості математичної моделі. Таку систему можна описати у вигляді наступної множини:

$$SS = \{ss_1, \dots, ss_n\}, \quad (2.1)$$

де ss_i, – підсистема, а $i = \overline{1,n}$, n – є невизначеним числом, оскільки КФС може складатися з довільної кількості периферійних вузлів і при цьому мати можливість на подальше розширення.

В наступних розділах ми розглянемо детальніше структуру кожної підсистеми описавши їх за допомогою математичних моделей, а також розписавши їхню взаємодію за допомогою діаграм.

2.2 Модель вузлової підсистеми КФСС

При розробці і проектуванні вузлової підсистеми було визначено, що дана підсистема повинна складатися з двох частин, а саме: апаратної частини та програмної. В даному розділі буде розглянуто окремо будову та інтеграцію апаратної частини в транспортний засіб, визначено програмну складову системи та способи взаємодії між апаратною та програмною складовою. Також при детальному описі розробимо математичну модель для апаратної та програмної складової.

2.2.1 Позиціонування і математична модель апаратної частини

Перед розробко вузла КФСС потрібно визначити до якої частини автомобільної мережі можна віднести нашу систему. Згідно [62] топологію автомобіля можна розділити на чотири частини які мають власні шлюзи для діагностики, а саме: телематику, силовий агрегат, корпус/комфорт, шасі. Розташування вище згаданих шлюзів в автомобілі можна також побачити на рисунку 2.2. В даній топології телематика найкраще підходить для інтеграції нашого вузла КФСС, оскільки вона поєднує в собі телекомунікації, транспортні технології, електротехніку та інформатику автомобіля, а також в ній використовується протокол зв'язку Media Oriented Systems Transport (MOST).

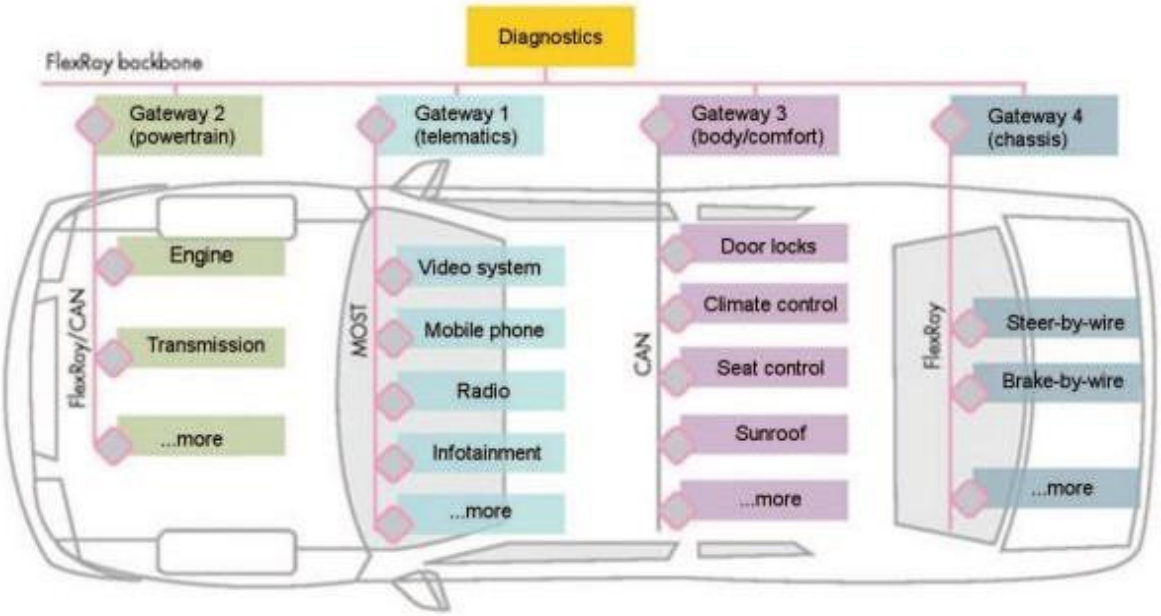


Рисунок 2.2 – Топологія автомобільної мережі [63]

Визначимо апаратні частини даної підсистеми і розпишемо їх у вигляді математичної моделі:

1. Модуль зв’язку (*CM*). На початкових етапах можна використовувати смартфон в якості точки доступу яка забезпечує зв'язок між системою та віддаленим сервером через мережу Інтернет.

Параметри: стан зв’язку (CM_{state}), швидкість передачі даних (CM_{speed}), час доступу до мережі (CM_{access_time}).

2. GPS-модуль (*G*). Визначатиме місцезнаходження нашої системи (транспортого засобу) і надаватиме координати, які потім можна використовувати для різних цілей, таких як навігація або визначення місця знаходження.

Параметри: координати(*x, y*), точність визначення місцезнаходження ($G_{accuracy}$), час оновлення (G_{update_time}).

3. Камера стеження(*C*). Призначена для збору інформації про події на дорозі.

Параметри: роздільна здатність ($C_{resolution}$), частота кадрів (C_{frame_rate}), час затримки обробки ($C_{processing_time}$).

4. Джерело живлення (*P*). Забезпечує електропостачання для всієї периферійної системи.

Параметри: Напруга живлення ($P_{voltage}$), Потужність (P_{power}).

5. Одноплатний комп'ютер(R). Відповідальний за обробку зображень, взаємодію з GPS-модулем і передачу даних на віддалений сервер.

Параметри: величина обсягу пам’яті (R_{memory}), частота процесора ($R_{processor_speed}$), рівень завантаження (R_{load}).

Таким чином, підсистему вузла можна описати у вигляді наступної множини:

$$HSS_i = \{CM, G, C, P, R\}. \quad (2.2)$$

Підмножина (2.2) є складовою наступної множини:

$$HSS = \{hss_1, \dots, hss_m\}, \quad (2.3)$$

де $HSS \in SS$ – підмножина систем які відповідають за спостереження та периферійні обчислення, а m – кількість можливих вузлів.

Визначені апаратні частини вузлової підсистеми представляють ключові компоненти КФСС в IoV для спостереження, кожна з яких відповідає за визначену функціональність та має свої власні параметри, які визначають їхню ефективність та продуктивність. Розробка математичної моделі підсистеми дозволила докладніше розуміти її роботу та взаємодію компонентів, що є ключовим для успішної імплементації та оптимізації функціонування системи в реальних умовах експлуатації.

2.2.2 Програмна складова периферійного вузла

До програмної складової вузла можна віднести наступне програмне забезпечення(ПЗ): операційну систему(ОС), машинний зір, нейронну мережу для розпізнавання образів, мова програмування Python.

Розглянемо обов’язки кожного ПЗ: ОС – забезпечує основні функції операційної системи на міні-комп'ютері(однопалатному комп’ютері), такі як

керування ресурсами, взаємодія зі зовнішніми пристроями тощо; машинний зір – обробка зображень з камери для виявлення об'єктів, визначення їхнього положення тощо; нейронна мережа для розпізнавання образів – аналіз та розпізнавання образів, наприклад, розпізнавання дорожніх знаків або інших об'єктів на дорозі; мова програмування Python – реалізація алгоритмів обробки зображень, навчання нейронної мережі та взаємодія з іншими частинами системи.

Програмна складова вузла може бути розглянута як математична модель, що використовується для аналізу зображень та розпізнавання образів. Основними елементами такої моделі будуть алгоритми обробки зображень та нейронна мережа для розпізнавання образів.

Обробку зображення можна визначити в якості наступної моделі, де нехай I - це вхідне зображення, що отримане з камери, тоді процес обробки зображень можна описати наступним чином:

$$I_{processed} = f_{preprocessing}(I), \tag{2.4}$$

де $f_{preprocessing}$ – це функція попередньої обробки, яка може включати в себе такі операції, як зменшення шуму, збільшення контрастності, підвищення чіткості тощо.

Нейронна мережа для розпізнавання образів буде вже взаємодіяти з отриманим з (2.4) $I_{processed}$ – обробленим зображенням, згідно якого нейронна мережа може визначити клас об’єкта C на зображенні. Модель по отриманні C можна побачити в формулі (2.5), яка описана нижче:

$$C = f_{NN}(I_{processed}), \tag{2.5}$$

де f_{NN} – це функція нейронної мережі, яка приймає оброблене зображення та повертає ймовірності (або вектор оцінок) для різних класів об'єктів на зображенні.

Ці дві частини можуть бути об'єднані в одну функцію для розпізнавання об'єктів на зображенні:

$$C = f_{recognition}(I) = f_{NN}(f_{preprocessing}(I)). \quad (2.6)$$

Модель (2.6) – лише загальна математична модель, яка описує процес обробки зображень та розпізнавання образів у програмній складовій. У подальшій роботі, а саме в третьому розділі будуть розглянуті більш детальні методи машинного навчання та різних алгоритми по вирішенню поставленої задачі, таких як згорткові нейронні мережі, алгоритми попередньої обробки, активаційні функції тощо.

2.2.3 Взаємозв’язок апаратної та програмної частини

Щоб найкраще зрозуміти взаємозв’язок компонентів даної підсистеми розглянемо модель станів, яка відображає стани підсистеми під час взаємодії її апаратної частини та програмної частини з навколишнім середовищем на іншими підсистемами. На рисунку 2.3 зображено модель станів та умови згідно яких дана підсистема вузла переходить з одного стану в інший.

Як видно на рисунку 2.3 дана підсистема має п’ять станів та дві умови за якої вона працює. Розглянемо їх детальніше та розпишемо компоненти які задіяні в кожному стані:

1. Рух транспортного засобу, стандартний стан системи в якому вона буде знаходитись більшість свого часу. В даному стані відпрацьовують всі апаратні та програмні складові підсистеми вузла, але не приймається ніяких рішень підсистемою.
2. Умова “АЗС помічено?” – стан в якому в об’єктив відеореєстратор потрапляє табло АЗС та нейронна мережа виділяє дошку з цінами. На даному кроці основну функцію виконують: відеореєстратор, одноплатний комп'ютер та нейронна мережа по розпізнаванню образів.
3. Одноплатний комп'ютер отримує дані з GPS-модуля – стан в якому підсистема отримує дані про місцезнаходження АЗС та форматує координати під власний тип. Це проміжний стан в якому знаходиться система і його функціонал

можна виділити в окремий паралельний процес разом з наступним кроком. На даному кроці основну функцію виконують: GPS-модуль, одноплатний комп'ютер та програмне забезпечення яке обробляє дані GPS-модуля в потрібний формат.

4. Комп'ютер підсистеми обробляє зображення – стан в якому програмне забезпечення відіграє основну роль по форматуванню зображення, а саме: вибір найякіснішого зображення з діапазону кадрів які записала камера, а потім зменшення розміру обраного зображення до виключно необхідного. Як і попередній крок даний функціонал можна виділити в окремий паралельний процес.

5. Умова “Система має доступ до інтернету?” – стан в якому за допомогою програмного забезпечення перевіряється наявність доступу до інтернету і якщо його немає, то процес по відправленню даних відтерміновується і підсистема або процес переходить в стан 6. В інакшому випадку якщо є доступ до мережі, то підсистема переходить до стану 7.

6. Стан в якому процес по надсиланню даних про ціни на АЗС відтермінований. В цьому стані даний процес лише очікує доступу до інтернету, а отже задіяна тільки ПЗ.

7. Стан в якому система надсилає дані до віддаленого сервера через мережу інтернет. Іншими словами передає зроблене зображення, координати та дату та час коли було зроблено знімок зображення.

Після виконання стану 7 підсистема зациклює своє виконання і переходить до стану 1. На даному етапі можна завершити опис взаємозв’язків апаратної та програмної даної підсистеми, оскільки за допомогою діаграми станів було висвітлено та описало взаємодію всіх її компонентів.

2.3 Модель клієнтської та серверної підсистеми КФСС

2.3.1 Структура підсистем

Клієнтська та серверна підсистеми є ключовими компонентами в розробці та функціонуванні КФСС. Клієнтська підсистема відповідає за взаємодію з користувачем та збір та передачу даних з датчиків та вузлів спостереження. З

іншого боку, серверна підсистема забезпечує обробку, аналіз та зберігання цих даних, а також надає різноманітні сервіси для користувачів, такі як моніторинг, аналітика та генерація звітів. В цьому розділі буде надано опис структури кожної з цих підсистем, а також описано їх взаємозв'язок та співпрацю в рамках кіберфізичної системи спостереження.

Детально описану структуру підсистем проілюстровано на рисунках 2.4-2.6, які відображають модель підсистем що надають послуги згідно функціональних вимог, також на рисунку відображено стрілками використання підсистемами послуг інших підсистем.

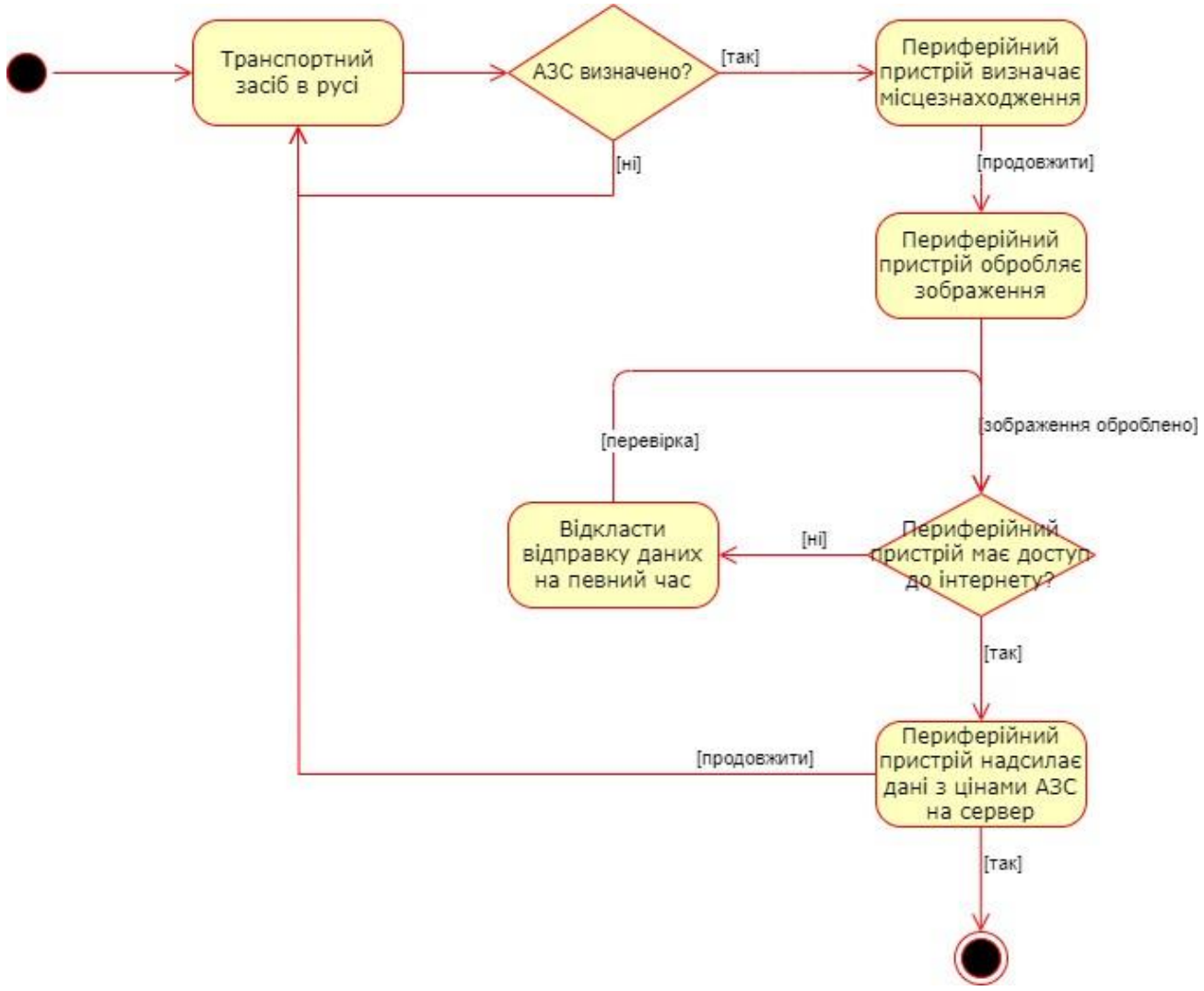


Рисунок 2.3 – Модель станів підсистеми вузла

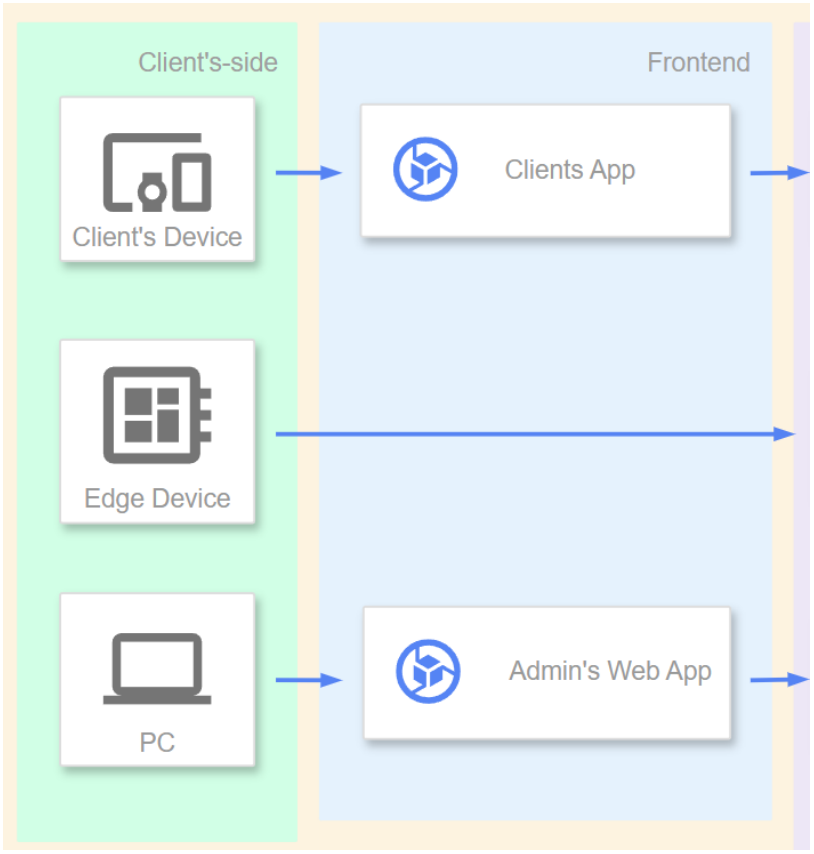


Рисунок 2.4 – Структура клієнтської підсистеми

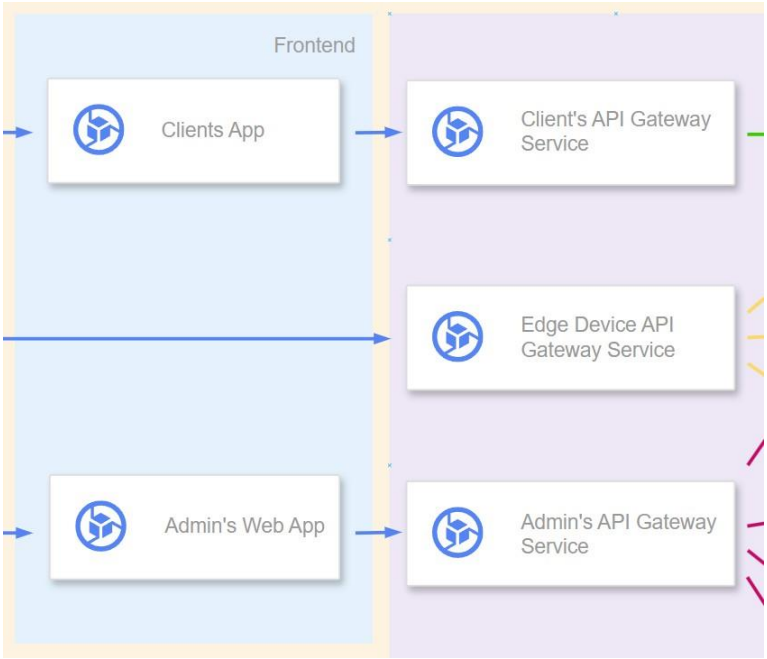


Рисунок 2.5 – Структура клієнтської та серверної підсистеми

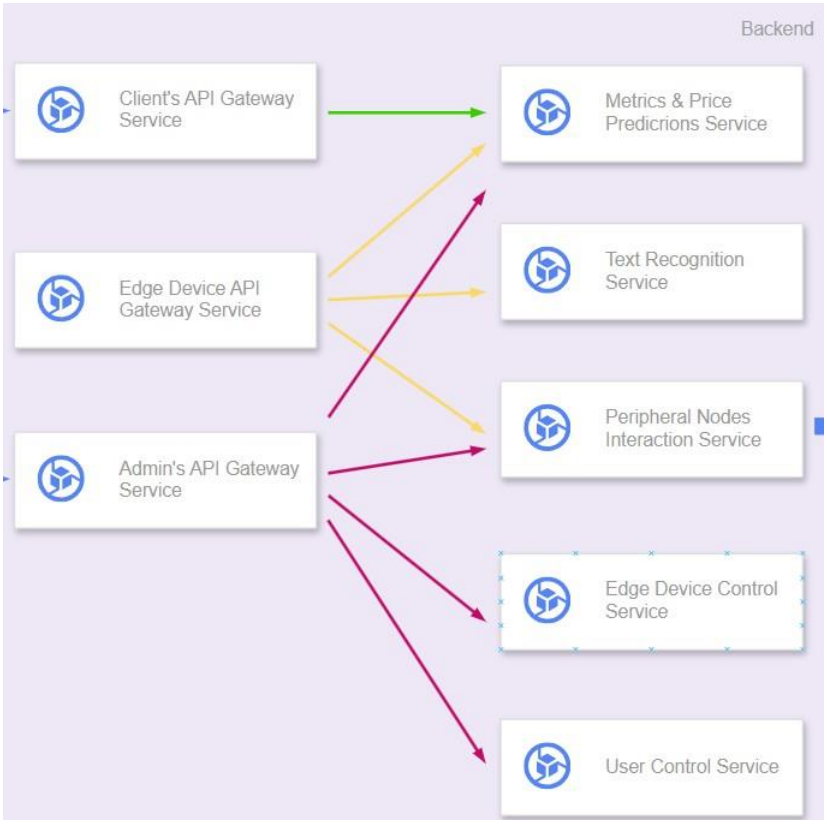


Рисунок 2.6 – Структура серверної підсистеми

Згідно рисунку 2.5 можна побачити, що клієнтська та серверна підсистема тісно пов’язані та є життєво необхідними для функціонування всієї КФСС.

Клієнтська частина складається з трьох типів пристроїв: периферійного пристрою(безпосередньо вузлова підсистема), пристрою клієнтів і персонального комп’ютера адміністратора. Для двох пристроїв з цього списку, а саме: пристрою клієнта та персонального комп’ютера адміністратора відведено спеціальні інтерфейси для взаємодії системою. На рисунку 2.4 це додаток для клієнтів та вебзастосунок для адміністраторів. Як видно з того самого рисунку периферійні пристрої не мають необхідності в такому інтерфейсі, що свідчить про створення автоматизованого способу комунікації між вузлом та серверною підсистемою. На цьому опис клієнтської частина закінчується та починається серверна підсистема.

Серверна підсистема складається з восьми служб які можна розділити за функціональними можливостями які повинні реалізовувати функціональні вимоги описані в наступному розділі. Служби мають наступні функції: шлюзи доступу для кожного типу клієнта, будування метриків та прогнозування, розпізнавання тексту

та класифікація, управління клієнтами, управління периферійними пристроями, взаємодії периферійних пристроїв. Варто відзначити, що на рисунку служби програмних інтерфейсів виконують роль не тільки шлюзів доступу до основного функціоналу системи, а й роль служб безпеки які відповідають за аутентифікацію та авторизацію користувачів та периферійних пристроїв. Кожна служба серверної підсистеми має можливість доступу до баз даних та сховищ які необхідні для їх ключової роботи. Сховище даних має зберігати безпосередньо зображення різних табло АЗС над якими проводиться аналіз, а бази даних зберігають в собі корисну інформацію по типу: оброблених даних з зображень, користувачів системи, журнали подій, локації(координати) АЗС та різні дані для аналітики та метрики.

Далі розглянемо функціональні вимоги які необхідні для створення клієнтської та серверної частини КФСС адже описаної структури недостатньо для створення інженерного рішення.

2.3.2 Функціональні вимоги

У нашому випадку, розробка клієнтської та серверної частини КФСС передбачає реалізацію різноманітних функцій. Ось функціональні вимоги, які є важливими для даних підсистем:

– обробка вхідних даних: підсистема повинна мати здатність обробити вхідні дані, що надходять від вузлів, включаючи зображення та відеопотоки;

– розпізнавання тексту: підсистема повинна володіти здатністю розпізнавати текст на вхідних зображеннях і класифікувати його згідно типу пального та цін на нього. Це може включати виявлення та класифікацію об'єктів за допомогою нейронних мереж;

– збереження результатів: підсистема повинна забезпечувати можливість збереження результатів розпізнавання об'єктів для подальшого аналізу та використання;

– відправлення даних на віддалений сервер: система повинна мати можливість відправляти оброблені дані на віддалений сервер для подальшої обробки та аналізу;

– аутентифікація та авторизація: підсистема повинна забезпечувати можливість аутентифікації користувачів та контролю доступу до функціональності системи;

– моніторинг та аналіз роботи системи: підсистема повинна здати забезпечити можливість моніторингу та аналізу роботи системи, включаючи перевірку її стану, виявлення помилок та відновлення роботи.

Визначені вимоги потребують доведення згідно наступних атрибутів: необхідність та доцільність, верифікованість, досяжність, однозначність та узгодженість. Далі, з урахуванням функціональних вимог, розглянемо доведення наявних атрибутів для кожної вимоги.

У вимозі "Обробка вхідних даних", перш за все, необхідно відзначити, що обробка вхідних даних є необхідною для виконання функцій системи спостереження, таких як розпізнавання об'єктів та аналіз ситуації. Вона є ключовою частиною системи, оскільки від цього залежить її здатність виконувати свої функції. По-друге, процес обробки вхідних даних може бути верифікованим через тестування, що забезпечує надійність та коректність роботи цієї функції. Крім того, обробка вхідних даних доступна за допомогою різних технологій та інструментів, що забезпечує досяжність цієї функціональної вимоги. Для забезпечення однозначності та узгодженості, функціональні вимоги повинні бути чітко визначені у специфікації системи та узгоджені з загальною архітектурою. Нарешті, обробка вхідних даних повинна відбуватися відповідно до визначеної послідовності у специфікації системи, щоб забезпечити коректну роботу та взаємодію всіх компонентів системи.

У вимозі "Розпізнавання об'єктів" також важливо врахувати, що ця функція є необхідною для даної підсистеми спостереження, оскільки дозволяє автоматично виявляти об'єкти на зображеннях або відеопотоках. Її ефективність та точність можна перевірити через тестування системи з використанням різних тестових

наборів даних. Також важливо, щоб функціональні вимоги були однозначно визначені та узгоджені з загальними вимогами до системи. Вони повинні бути реалізовані відповідно до визначеної послідовності, щоб забезпечити коректну роботу та взаємодію всіх компонентів системи.

Розглянемо доведення наявних атрибутів для функціональної вимоги "Збереження результатів". Збереження результатів є важливою функцією для системи спостереження, оскільки це дозволяє зберегти важливу інформацію про розпізнані об'єкти та події для подальшого аналізу або використання. Ця функція необхідна для забезпечення повноцінної роботи системи. Ефективність та точність збереження результатів можна перевірити через тестування системи зі збереженням та відновленням даних. Досяжність цієї функції досяжна за допомогою доступних технологій та механізмів збереження даних, таких як бази даних, файлові системи тощо. Використання таких інструментів дозволяє системі ефективно зберігати результати розпізнавання. Функціональні вимоги до збереження результатів повинні бути чітко визначені та узгоджені з загальними вимогами до системи спостереження та її архітектурою, щоб забезпечити синхронізацію роботи всіх компонентів системи. Збереження результатів повинно відбуватися відповідно до визначеної послідовності у специфікації системи, що забезпечить коректну роботу та взаємодію всіх компонентів системи відповідно до її призначення.

Вимога щодо відправлення даних на віддалений сервер є критичною для системи спостереження. Це не лише дозволяє передавати результати аналізу на сервер для подальшої обробки чи збереження, а й є ключовим елементом взаємодії системи з іншими системами або користувачами. Ефективність та коректність цієї функції перевіряються через тестування системи, включаючи передачу, прийом та обробку даних на сервері. Вона досяжна завдяки мережевим протоколам та інтернет-з'єднанню. Функціональні вимоги до відправлення даних повинні бути чітко визначені, узгоджені з загальними вимогами до системи та відбуватися відповідно до визначеної послідовності.

Моніторинг та аналізу роботи системи – ця функція є важливою для забезпечення ефективності та надійності системи. Вона дозволяє виявляти можливі проблеми та збирати статистику для подальшого вдосконалення. Ефективність та правильність моніторингу та аналізу перевіряються через тестування системи, включаючи спостереження за роботою системи та аналіз отриманих результатів. Ці функції доступні завдяки різним інструментам, таким як системи моніторингу та візуалізації даних. Функціональні вимоги до цих процесів повинні бути чітко визначені, узгоджені та відбуватися відповідно до визначеної послідовності.

2.3.3 Математична модель підсистем

Згідно функціональних вимог можемо визначити необхідні компоненти для клієнтської та серверної підсистеми. Відповідно до вже описаних множини (2.1) та підмножини (2.3), можна створити подібну підмножину для даних підсистем:

$$SSS = \{sss_{m+1}, \dots, sss_{m+9}\}, \tag{2.7}$$

де $SSS \in SS$ – підмножина систем які відповідають за визначений раніше набір функціональних вимог. Також варто відзначити, що дана підмножина має визначену кількість підсистем, і її можна розділити на ще дві підмножини, а саме:

$$FSS = \{fss_{m+1},\ fss_{m+2}\}, \tag{2.8}$$

де $FSS \in SSS$ – підмножина систем які відповідають за репрезентацію інтерфейсу користувачу, де fss_{m+1} – додаток для всіх користувачів, fss_{m+2} – додаток для адміністраторів системи.

$$BSS = \{bss_{m+3}, \dots, bss_{m+9}\}, \tag{2.9}$$

де $BSS \in SSS$ – підмножина систем які відповідають за реалізацію функціональних можливостей серверної підсистеми, де bss_{m+3} – програмний інтерфейс який надає можливість взаємодіяти підсистемі fss_{m+1} з іншими компонентами системи, bss_{m+4} – програмний інтерфейс який надає можливість взаємодіяти підсистемі fss_{m+2} з іншими компонентами системи, bss_{m+5} – підсистема яка відповідає обробку вхідних даних та взаємодію з периферійними пристроями, bss_{m+6} – підсистема яка відповідає за розпізнавання тексту на зображені, bss_{m+7} – підсистема аутентифікації та авторизації для адміністраторів, bss_{m+8} – підсистема прогнозування та аналітики, bss_{m+9} – підсистема взаємодії з периферійними вузлами.

2.4 Модель основного процесу

Модель основного процесу системи моніторингу цін на пальне може бути побудована наступним чином:

Об’єкт *O*: цінові дані за пальне на різних ринках, геолокаційні дані про заправки та їх ціни, аналіз тенденцій цін та попиту.

Суб’єкт *A*: користувачі, які використовують систему для моніторингу цін на пальне та прийняття торговельних рішень.

Задача *Z*: забезпечення користувачам доступу до актуальної цінової інформації, аналіз та прогнозування змін цін на пальне, надання рекомендацій щодо оптимальних моментів для заправки.

Ресурси *B*: сервери для обробки та зберігання цінових даних, геолокаційні датчики та системи збору даних, інтернет-зв'язок для отримання актуальної інформації про ціни, транспортний засіб.

Середовище моделювання *C*: ринкове середовище, включаючи різні географічні регіони з різними ринковими умовами для пального.

Отже, модель основного процесу системи моніторингу цін на пальне може бути представлена наступним кортежом:

$$P = < O, A, Z, B, C >. \quad (2.10)$$

Модель (2.10) процесу моніторингу цін на пальне є засобом, спрямованим на забезпечення трьох основних аспектів: доступності актуальної інформації про ціни, аналізу ринку та надання рекомендацій щодо оптимальних моментів для заправки. Вона відображає реальний процес, проста у розумінні та використанні. Ця модель надає користувачам доступ до інформації через різноманітні пристрої та може бути легко налаштована. Вона забезпечує повноту та достовірність інформації, а також стабільність у змінних умовах. Модель є цілісною системою, що може розвиватися та пристосовуватися до нових умов та вимог користувачів.

2.5 Часова модель КФСС

В даному розділі розроблено та викладено архітектуру мережі та модель затрат часу для нашої КФС. Основні позначення системи приведено в таблиці 2.1.

Таблиця 2.1 - Позначки та визначення

Позначка	Визначення
N	Кількість транспортних засобів з кіберфізичною системою
S	Набір периферійних систем, де $S = \{s_1, s_2, \dots s_N\}$
L	Кількість віртуальних машин в S_N
J	Кількість виконуваних завдань
V_N	Набір віртуальних машин в S_N, де $V_N = \{v_{n,1}, v_{n,2}, \dots v_{n,L}\}$
T	Набір виконуваних завдань, де $T = \{t_1, t_2, \dots t_J\}$
T_{total}	Загальні витрати часу на виконання завдань

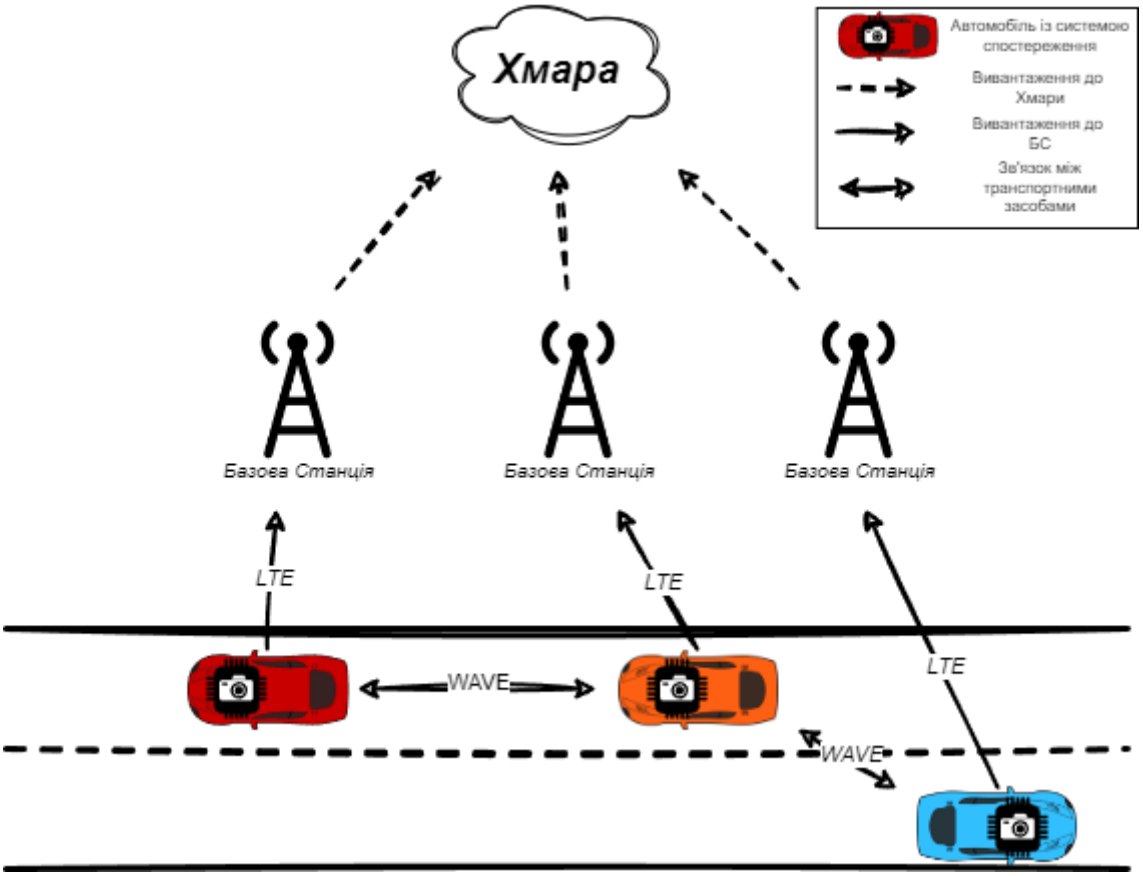


Рисунок 2.7 - Архітектура відеоспостереження з периферійними обчисленнями, розташованими в автомобілях

У цій мережевій моделі ми використовуємо технологію периферійного обчислення в системі відеоспостереження. Як показано на рисунку 2.7, пропонується архітектура відеоспостереження з периферійним обчисленням, де термінали спостереження встановлені в автомобілях, а згенеровані відеодані вивантажуються напряму на базову станцію з використанням LTE зв'язку. Периферійні вузли з потужними обчислювальними можливостями (транспортні засоби з КФС) беруть на себе обов'язки з обчислювальними завданнями. Потім результати обчислень відправляються з периферичних вузлів до хмарного середовища через базові станції. У цьому сценарії вздовж дороги розташовано N транспортних засобів, на кожному з яких встановлено КФС для відеоспостереження, позначених як $S = \{s_1, s_2, \ldots s_N\}$. Набір пристроїв КФС складається з двох компонентів: периферійних вузлів та точок доступу. Нехай $T =$

$\{t_1, t_2, \ldots t_J\}$ позначає набір з J обчислювальних завдань, що генеруються з вузлів. Периферійні вузли володіють потужними обчислювальними можливостями та місткістю для передобробки, яка складається з віртуальних машин (VM). У $s_w \in L$ віртуальних машин, визначених як $V_N = \{v_{n,1}, v_{n,2}, \ldots v_{n,L}\}$

Наступна модель затрат часу буде розроблятись по типу роботи [64] в якій, автори також розробляли модель затрат часу, модель балансу навантаження і модель ентропії конфіденційності для системи вивантаження яка базувалась на статичних об'єктах для спостереження за трафіком транспортних засобів.

Затрати часу складаються з часу обчислення на відповідному периферійному вузлі та часу вивантаження. Затрати часу є важливим параметром, який визначає якість сервісу в реальному часі на пристроях моніторингу.

Коли відео завдання передаються на пристрій обробки зображення, КФС повинна надавати віртуальні машини для виконання обчислень. Обчислювальна здатність периферійних вузлів пов'язана з кількістю вільних віртуальних машин. Таким чином, кількість вільних віртуальних машин у s_n позначається як x_n, а операційна здатність кожної віртуальної машини позначається як φ. Припустимо, що відеозавдання з тривалістю F_j обробляється в s_n, а час обчислення s_n виражено як:

$$h_j = \sum_{n=1}^{N} k_{j,n} \frac{F_j}{x_n \cdot \varphi}, \tag{2.11}$$

де $k_{j,n} = 1$ якщо дані t_j надійшли в s_n, в іншому випадку 0.

Коли обробка відео в s_n завершена, то обрахований результат вивантажується з периферійних пристроїв в хмарний центр. Час вивантаження результату обчислень t_j з вагою $f_{j,output}$, можна вирахувати як:

$$o_j = \sum_{n=1}^{N} k_{j,n} \left(\frac{f_{j,output}}{y} + \frac{f_{j,output}}{\lambda} \right), \tag{2.12}$$

де η позначає швидкість передачі між периферійними вузлами і базовими станціями, а λ - швидкість передачі між базовими станціями та хмарним центром обробки даних.

З вирахуваними двома компонентами, загальні витрати часу на виконання завдань можна визначити за наступною формулою:

$$T_{total} = \sum_{j=1}^{J} (h_j + o_j). \quad (2.13)$$

2.6 Висновки

У даному розділі було проведено докладний опис загальної моделі КФСС, що включає в себе опис вузлової підсистеми, клієнтської та серверної підсистеми, а також основного процесу та часової моделі.

У підрозділі "Модель вузлової підсистеми КФСС" була представлена детальна інформація про позиціонування та математичну модель апаратної частини периферійного вузла, програмну складову цього вузла та їх взаємозв'язок.

У підрозділі "Модель клієнтської та серверної підсистеми КФСС" було описано структуру цих підсистем, їх функціональні вимоги та математичну модель. Це дозволило докладно розібратися у взаємодії клієнтської та серверної частин КФСС.

Також було розглянуто модель основного процесу та часову модель, що є важливими компонентами для розуміння роботи та ефективності КФСС.

Отже, даний розділ надає повний огляд та розуміння структури та функціональності КФСС, що є важливим у подальшому дослідженні та впровадженні цієї системи.

3 МЕТОДОЛОГІЯ ДОСЛІДЖЕННЯ КФСС

У цьому розділі будуть розглянуті різні методи аналізу даних, які можуть бути використані для реалізації КФС для моніторингу цін на пальне та геолокаційних даних. Це включає в себе статистичні методи, методи машинного навчання, методи комунікації, а також методи аналізу великих обсягів даних. Для кожного методу буде розглянуто його переваги, недоліки та області застосування.

3.1 Методи комунікації між підсистемами КФСС

Методи комунікації між підсистемами можна описати наступними зв’язками між:

- периферійними вузлами;
- периферійними вузлами та серверною підсистемою;
- серверною підсистемою та клієнтською підсистемою.

Зв’язок між периферійними вузлами може встановлюватися за допомого технологій бездротового зв'язку. Проте такі технології як Wi-Fi, Bluetooth або Zigbee(який часто використовується в розумних будинках), не придатні для комунікації між периферійними вузлами які знаходяться в транспортному засобі. Для цього вже є розроблена бездротова технологія – Dedicated Short-Range Communications (DSRC), яка призначена для обміну даними на короткій відстані між рухомими та стаціонарними об'єктами, такими як автомобілі, дорожні знаки та інфраструктура. Технологія DSRC використовується в ITS (Intelligent Transportation Systems) для реалізації системи V2X (Vehicle-to-Everything), яка дозволяє автомобілям обмінюватися інформацією з іншими автомобілями, дорожніми знаками, світлофорами та іншими елементами інфраструктури.

DSRC працює на відведеному спектрі радіочастот (5.9 ГГц) та використовує різні протоколи для комунікації, такі як WAVE (Wireless Access in Vehicular Environments) і IEEE 802.11p. Вона дозволяє передавати різноманітну інформацію, таку як швидкість, розташування, стан дорожньої ситуації, безпекові повідомлення

тощо. DSRC можна використовувати як спосіб комунікації між вузлами у системі IoV. Вона дозволяє автомобілям обмінюватися даними про дорожні умови, навігаційну інформацію, безпекові повідомлення та іншу важливу інформацію. Даний спосіб комунікації вже імплементується в автомобілі вироблені з 2023 року [65], але як вже було зазначено раніше даний метод призначений для “важливої інформації”, а інформація про ціни на АЗС, такою не є. Тому можливість комунікації між периферійними вузлами через технологію DSRC відпадає і залишається лише спосіб обміну інформацією тільки з використанням серверної підсистеми.

Для зв'язку між периферійними вузлами та серверною підсистемою найдоцільніше обрати один з видів мобільного зв’язку. Він зараз є найбільш доступним на території України, що надає можливість практичного застосування КФСС та її дослідження. Вибір мобільного зв’язку можна охарактеризувати також за допомогою наступних характеристик:

– широта охоплення. Мобільний зв'язок має широту охоплення, що означає, що він доступний в більшості регіонів, включаючи віддалені райони та області з обмеженим доступом до інших мережних технологій;

– мобільність. Мобільний зв'язок дозволяє периферійним вузлам зберігати свою з'єднаність навіть у русі. Це важливо для систем, розташованих у транспортних засобах, які можуть пересуватися по великим територіям;

– висока пропускна здатність. Технології 4G та 5G забезпечують високу швидкість передачі даних, що важливо для швидкої обробки та передачі великого обсягу інформації між периферійними вузлами та серверною підсистемою;

– надійність. Мобільний зв'язок зазвичай має стабільну та надійну роботу, що важливо для систем, які потребують постійного зв'язку для передачі даних та отримання відповідей в реальному часі;

– легкість налаштування та використання. Встановлення та налаштування мобільного зв'язку може бути досить простим і зручним процесом, особливо з використанням сучасних технологій управління мережами.

Також варто наголосити, що під час тестування КФСС буде застосовуватись мобільний зв’язок 4G, оскільки в Україні мережа 5G поки не розбудована навіть на рівні експериментальних зразків [66]. Це обмеження суттєво впливає на стратегію обробки відео та на збереження результатів роботи системи, оскільки швидкість інтернету суттєво знижується при русі периферійного вузла, що призводить до неможливості трансляції відео потоку.

Застосування мобільного зв’язку відкриває можливості використання хмарних платформ для серверної підсистеми, що дозволить обмін даними через інтернет, або мережеві протоколи для передачі даних через мережу, по типу TCP/IP або HTTP.

Обираючи HTTP в якості протоколу передачі даних ми обираємо простий спосіб для обміну даними між клієнтом і сервером. Кожен запит або відповідь якого складається з рядка заголовків та, за необхідності, тіла повідомлення. Можливість визначати методи запиту для: отримання даних від сервера або відправлення даних на сервер повністю задовольняє потреби для вузлової підсистеми, надаючи їй можливість взаємодіяти з програмним інтерфейсом серверної підсистеми.

Взаємодія між серверною підсистемою та клієнтською підсистемою може здійснюватися через реалізацію RESTful API для взаємодії за допомогою HTTP-запитів або WebSocket-ів для постійного зв'язку у режимі реального часу, або за допомогою мережевих протоколів, таких як TCP/IP або UDP.

Опишемо детально методи, які необхідні для комунікації між одним периферійним вузлом і сервером:

Метод відправки даних ідентифікованої АЗС. Цей метод може бути використаний для передачі важливої інформації про конкретну автозаправну станцію (АЗС), такої як її унікальний ідентифікатор, географічні координати та час знімку. Відправлення цих даних може допомогти в моніторингу та управлінні мережею КФСС, забезпечити швидку реакцію на будь-які проблеми або несправності і покращити ефективність управління. Характеристики цього методу наведено в таблиці 3.1.

Таблиця 3.1 – Атрибути методу відправки даних ідентифікованої АЗС

Атрибут	Опис
Шлях	/api/gas-station/location
Метод	POST
Заголовки	User-Agent, Content-Type, API-KEY, Image-Id
Тіло	координати, час знімку

Метод відправки знімку “Табло АЗС”. Цей метод може бути використаний для передачі фотографій з цінами на АЗС. Отримання знімків табло може допомогти в аналізі цінової політики, виявленні тенденцій зміни цін, а також у визначенні ефективності рекламних кампаній або акцій. Ця інформація може бути корисною для прийняття рішень щодо стратегій ціноутворення, маркетингових дій та управління запасами палива. Характеристики цього методу наведено в таблиці 3.2.

Таблиця 3.2 – Атрибути метод відправки знімку “Табло АЗС”

Атрибут	Опис
Шлях	/api/gas-station/board-image
Метод	POST
Заголовки	User-Agent, Content-Type, API-KEY, Image-Id
Тіло	Масив байтів які описують зображення

3.2 Методи розпізнавання образів та тексту

Для системи, що складатиметься з периферійних вузлів з камерами спостереження вбудованими в транспортні засоби, і виконуватиме такі завдання, як розпізнавання табло з цінами АЗС та відправлення відформатованого

зображення на віддалений сервер для подальшого розпізнавання тексту і класифікації інформації, можна використати наступні групи алгоритмів: алгоритми виявлення об'єктів на зображенні [67-69], алгоритми обрізки та форматування зображень(image cropping and formatting algorithms), алгоритми оптичного розпізнавання символів(OCR algorithms).

3.2.1 Алгоритми виявлення об'єктів на зображенні

Алгоритми виявлення об'єктів на зображенні. Для виявлення та виділення табло з цінами АЗС на зображенні.

Можна скористатись двома популярними методами об'єктного виявлення на зображеннях, такими як YOLO (You Only Look Once) [70, 71] або SSD (Single Shot MultiBox Detector) [72, 73].

Вони є доцільними для розгляду, оскільки вони належать до сімейства одноступеневих детекторів, перевагою яких є швидкість, а двоступеневі детектори більше зосереджені на точності.

В нашому випадку швидкість – є ключовою характеристикою, тому що периферійні вузли знаходитимуться в транспортних засобах, які будуть постійно в русі.

YOLO став одним з популярних методів об’єктного виявлення із-за своєї швидкості та ефективності. Цей метод пропонує визначення об'єктів та їх класів одночасно.

Ідея полягає в розділенні зображення на сітку, після чого для кожної клітини цієї сітки модель визначає потенційні об'єкти, їх класи та визначає рамки, приклад можете побачити на рисунку 3.1.

Цей підхід дозволяє YOLO працювати дуже швидко, що робить його ідеальним для застосувань, де важлива висока швидкість виявлення об'єктів.

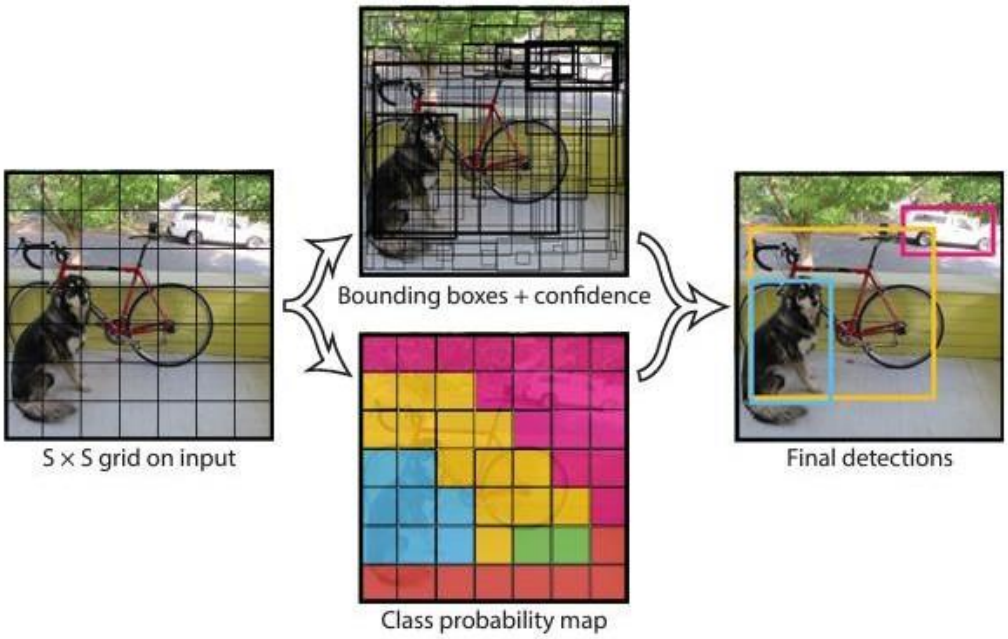


Рисунок 3.1 – Приклад роботи YOLO-алгоритму [74]

Як видно з рисунку 3.1 першим кроком є: поділ зображення на сітку S x S і якщо центр об’єкта потрапляє в клітинку сітки, то ця комірка відповідає за виявлення цього об’єкта.

Кожна комірка сітки описується обмежувальним прямокутником B і оцінкою достовірності для цих обмежень. Ці показники достовірності відображають наскільки модель впевнена в тому, що клітинка містить об’єкт і наскільки вона впевнена в передбачені. Формально достовірність можна описати наступним чином:

$$Cert = \Pr(Obj) * IOU_{truth\ pred}. \quad (3.1)$$

Якщо такій клітинці немає жодного об’єкта, то оцінка достовірності має бути рівною нулю. В інакшому випадку, оцінка достовірності повинна дорівнювати перетину наскрізного об’єднання – $IOU_{truth\ \ pred}$ між прогнозованим блоком і фундаментальною істиною.

Кожен обмежувальний прямокутник складається з п’яти прогнозів: *x*, *y*, *w*, *h* і достовірність. Прогнози *x*, *y* – це координати центру обмежувального прямокутника відносно меж комірки сітки. Прогнози *w*, *h* – це ширина та висота, які прогнозуються відносно всього зображення. Прогноз достовірності передбачає представляє IOU між прогнозованим блоком і будь яким базовим блоком фундаментальної істини.

Кожна комірка сітки також передбачає ймовірності умовного класу *C*, $Pr(Class_i|Object)$. Ці ймовірності залежать від комірки сітки, що містить об’єкт. Незалежно від кількості блоків *B* прогнозування відбувається лише за одним набором ймовірностей класу на комірку сітки.

Під час перевірки відбувається множення ймовірності умовного класу на передбачення достовірності окремих блоків, що дає оцінки достовірності для кожного блоку для конкретного класу. Загальний вигляд перевірки можна описати наступною формулою:

$$\Pr(Class_i|Obj) * \Pr(Obj) * IOU_{truth\ pred} = \Pr(Class_i) * IOU_{truth\ pred}. \quad (3.2)$$

Показники в (3.2) кодують як ймовірність того, що цей клас з’явиться в полі, так і те, наскільки добре прогнозований блок відповідає об’єкту.

Проте YOLO має і недоліки, оскільки накладає сильні просторові обмеження на передбачення меж обмеження, оскільки кожна клітина сітки передбачає лише дві рамки та може мати лише один клас. Це просторове обмеження обмежує кількість об'єктів, що знаходяться поблизу, які може передбачити така модель. Дана модель має проблеми з малими об'єктами, які з'являються у групах, для прикладу – стадо птахів. Оскільки YOLO-модель навчається передбачати межі обмежень з даних, вона має проблеми з узагальненням до об'єктів у нових або нестандартних відношеннях сторін або конфігураціях. YOLO також використовує відносно грубі ознаки для передбачення меж обмежень, оскільки її архітектура має кілька шарів зменшення розміру зображення. І останнім недоліком такої моделі, що навіть якщо модель навчається на функції втрат, яка наближає продуктивність

виявлення, функція втрат розглядає помилки однаково у малих та великих межах обмежень. Невелика помилка в великій рамці зазвичай є безпечною, але невелика помилка в малій рамці має набагато більший вплив на перетин над об’єднанням. І тому виходить, що основним джерелом помилок є неправильна локалізація.

Щодо SSD, то це також відомий своєю швидкістю та точністю алгоритм виявлення об’єктів на зображенні. Як і YOLO, він розділяє зображення на сітку, але додає ще один рівень складності. SSD використовує різні масштаби областей для виявлення об'єктів на різних рівнях деталізації зображення. Це робить його більш гнучким та точним у виявленні малих об'єктів, що дозволяє йому ефективно працювати в різних сценаріях. Приклад роботи SSD методу можна побачити на рисунку 3.2.

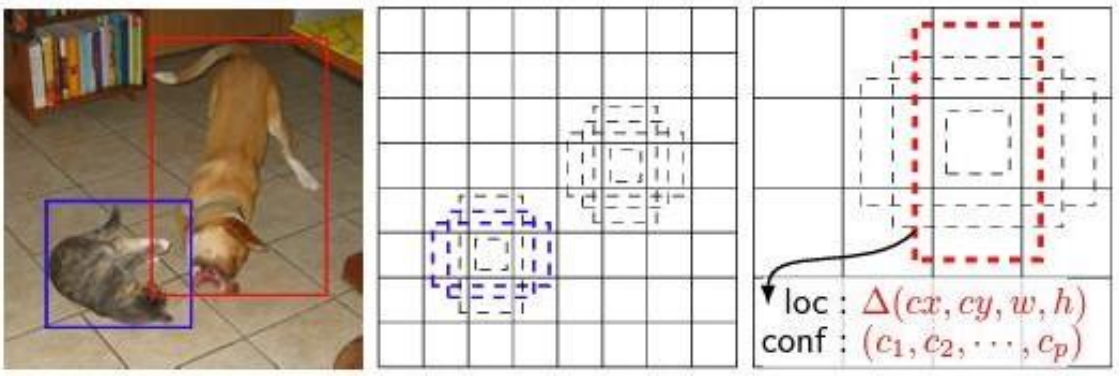


Рисунок 3.2 – Приклад роботи SSD-алгоритму [75]

SSD підхід базується на контроль процесу з використанням його очікуваних результатів в згортковій нейронній мережі, яка генерує фіксований набір обмежених комірок та оцінок присутності екземплярів класів об'єктів у цих комірках, за якими слідує крок пониження максимальної оцінки для отримання кінцевих виявлень. Початкові шари мережі базуються на стандартній архітектурі, яка використовується для класифікації зображень високої якості, яка називається – базовою мережею. Потім додається додаткова структура до мережі для отримання розпізнавання з наступними ключовими особливостями:

Багатошарові карти ознак для виявлення. Додаються згорткові шари ознак до кінця обрізаної базової мережі. Ці шари поступово зменшуються за розміром і дозволяють передбачення виявлень на різних масштабах. В порівнянні з YOLO, який працює на картах ознак єдиного масштабу, згорткова модель для передбачення виявлень в SSD відрізняється для кожного шару ознак.

Згорткові предиктори для виявлення. Кожен доданий шар ознак може створювати фіксований набір передбачень про виявлення за допомогою набору згорткових фільтрів, приклад з порівнянням YOLO методом можна побачити на рисунку 3.3 у верхній частині архітектури мережі SSD. Для шару ознак розміром $m \times n$ з p-каналами. Базовим елементом для передбачення параметрів потенційного виявлення є: невелике ядро розміром $3 \times 3 \times p$, яке виробляє оцінку для категорії або зміщення форми відносно координати типової рамки. На кожній позиції $m \times n$ місць, де застосовано ядро, воно виробляє вихідне значення. У порівнянні з архітектурою YOLO, яка використовує проміжний повністю з'єднаний шар замість згорткового фільтра для цього кроку, в SSD вихідні значення зміщення рамки виміряні відносно позиції типової рамки відносно кожного місця на карті ознак.

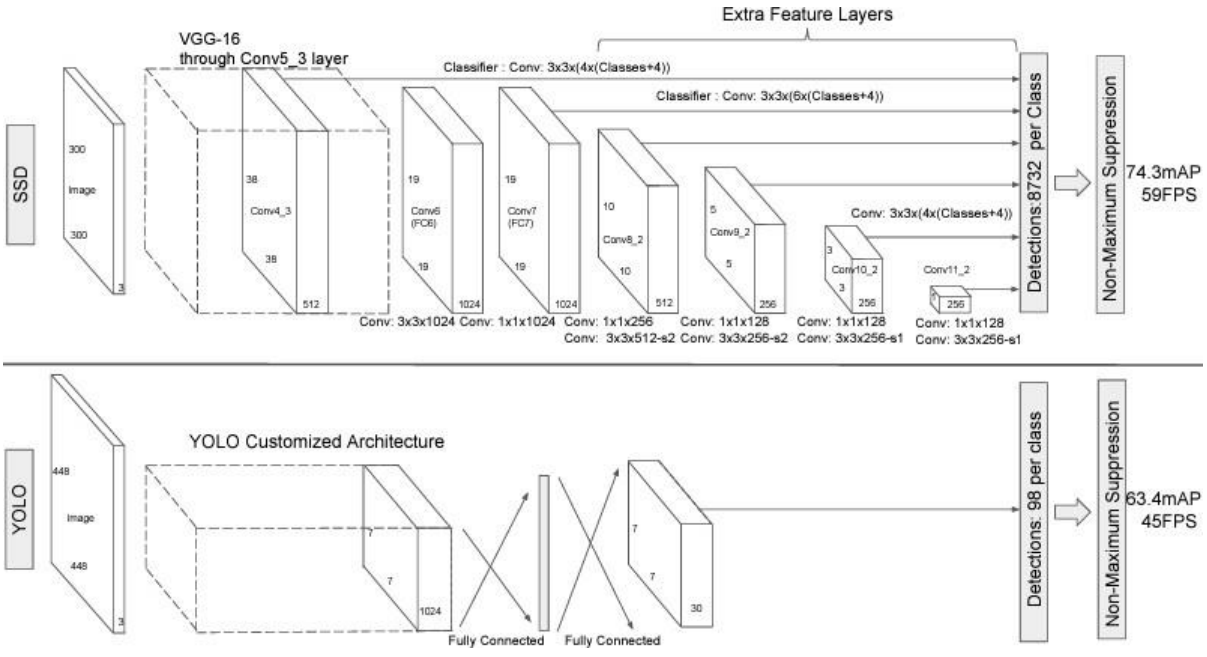


Рисунок 3.3 – Порівняння між двома моделями виявлення одиночного пострілу: SSD і YOLO [75]

Типові рамки та співвідношення сторін. На даному кроці відбувається асоціація набору типових рамок з кожною коміркою карти ознак, для кількох карт ознак у верхній частині мережі. Типові рамки щільно розташовані на карті ознак конвеєрним чином, щоб позиція кожної рамки відносно відповідної клітинки була зафіксованою. Для кожної клітинки карти ознак ми передбачаємо зміщення відносно форм типових рамок у клітинці, а також бали для кожного класу, які вказують на присутність екземпляра класу в кожній з цих рамок. Конкретно, для кожної рамки з k у даному місці ми обчислюємо c балів для класу та 4 зміщення відносно початкової форми типової рамки. Це призводить до загальної кількості $(c + 4)*k$ фільтрів, які застосовуються навколо кожного місця на карті ознак, що дає $(c + 4)*k*m*n$ виходів для карти ознак розміром $m \times n$. Для ілюстрації типових рамок дивіться рисунок 3.2.

Як видно з порівняння можна висловити гіпотезу, що YOLO алгоритм – є швидшим за SSD, а отже він здатен обробити більшу кількість кадрів за одну секунду у відеопотоці, тому при розробці програмного забезпечення для підсистеми КФСС(периферійного вузла) буде застосовуватись YOLO алгоритм.

3.2.2 Алгоритми оптичного розпізнавання символів

До алгоритмів оптичного розпізнавання тексту або символів які необхідні для створення КФСС можна віднести наступні методи:

Метод згорткових нейронних мереж (CNN) [76]. Цей метод використовується для розпізнавання тексту, використовуючи згорткові шари для виявлення функцій у зображеннях, а потім повністю зв'язані шари для класифікації символів.

Метод опорних векторів (SVM) [77]. SVM може бути використаний для класифікації областей на зображенні, які містять текст та не містять тексту.

Алгоритм на основі здобутку знань (KNN) [78]. KNN може бути застосований для визначення, які пікселі на зображенні ймовірно відповідають символам.

Методи на основі статистичного аналізу [79]. Ці методи можуть використовувати статистичні ознаки, такі як гістограми орієнтованих градієнтів (HOG) або Local Binary Patterns (LBP), для виявлення та розпізнавання тексту.

Методи на основі морфологічних операцій [80]. Вони використовуються для очищення зображень від шуму та виявлення текстових регіонів.

Для реалізації КФСС яка вимагає комплексне функціональне забезпечення стосовно спостереження за цінами на АЗС з використанням відеоспостереження, найкращим рішенням буде обрати методи на основі морфологічних операцій, зокрема розширення та ерозію, разом із методами глибокого навчання, такими як згорткові нейронні мережі (CNN).

Морфологічні операції дозволять виявити та виокремити табло з цінами АЗС на відеопотоках, але їх ефективність може бути обмеженою в разі складних умов освітлення або зміни погодних умов. Тому згорткові нейронні мережі можуть допомогти в розпізнаванні та класифікації зображень з табло цінами, забезпечуючи більшу точність та надійність системи.

Крім того, для підтримки роботи в реальному часі та автоматизації процесу розпізнавання та аналізу зображень, можна використати алгоритми на основі здобутку знань (KNN), які відносяться до методів машинного навчання та можуть швидко та ефективно класифікувати об'єкти на зображеннях. Такий підхід дозволить побудувати систему, яка забезпечить високу швидкість роботи та точність виявлення та аналізу цін на АЗС.

Наразі вже існує безліч інструментарій та сервісів, які надають можливість оптичного розпізнавання символів з застосуванням вище згаданих методів.

3.2.3 Аналіз сервісів оптичного розпізнавання символів

Для оптичного розпізнавання символів існує декілька комерційних і безкоштовних рішень по типу:

Tesseract – один з найбільш популярних і відкритих OCR-движків, розроблений в Google. Tesseract здатний розпізнавати текст у різних мовах та форматах.

OCRopus – відкритий OCR-движок, який підтримує розпізнавання тексту на різних мовах та має гнучку архітектуру для навчання та розширення.

OpenCV OCR – бібліотека комп'ютерного зору з відкритим кодом, також має функції OCR, які можна використовувати для розпізнавання тексту на зображеннях.

Google Cloud Vision API – хмарна платформа для розпізнавання об'єктів, яка має вбудовані функції OCR для розпізнавання тексту на зображеннях.

Amazon Textract – сервіс від Amazon, який автоматично витягує текст та дані з документів, зображень та PDF-файлів, використовуючи різні техніки, включаючи OCR.

Microsoft Azure Computer Vision – хмарний сервіс, який має функції OCR для розпізнавання тексту на зображеннях.

Проаналізуємо детальніше Google Cloud Vision API і Tesseract.

Google Cloud Vision API – це комерційне рішення, яке надає функціонал по типу: аналізу зображення та виявлення на них різноманітні об'єктів, таких як люди, тварини, автомобілі, будівлі, різноманітні об'єкти в природі тощо; функція по виявленню тексту на зображеннях і перетворюванню його на редактований текст для подальшого аналізу або обробки; класифікування зображення за категоріями або тегами, що допомагає розпізнавати зміст зображення; аналіз змісту зображення, такий як наявність насильства, образливого контенту або інших неприйнятних елементів; пошук подібних зображень в Інтернеті або власних базах даних. Також варто відзначити, що Google Cloud Vision API можна з легкістю інтегрувати з іншими сервісами Google Cloud, такими як Google Cloud Storage для збереження або Google Cloud Functions для автоматизації обробки зображень. Основним недоліком такого рішення є необхідність платити за використання ресурсу.

Скористаємося тестовим застосунком який надає можливість скористатись можливостями Google Cloud Vision API та визначимо його спроможності. Результат тесту можна побачити на рисунку 3.4 та 3.5.

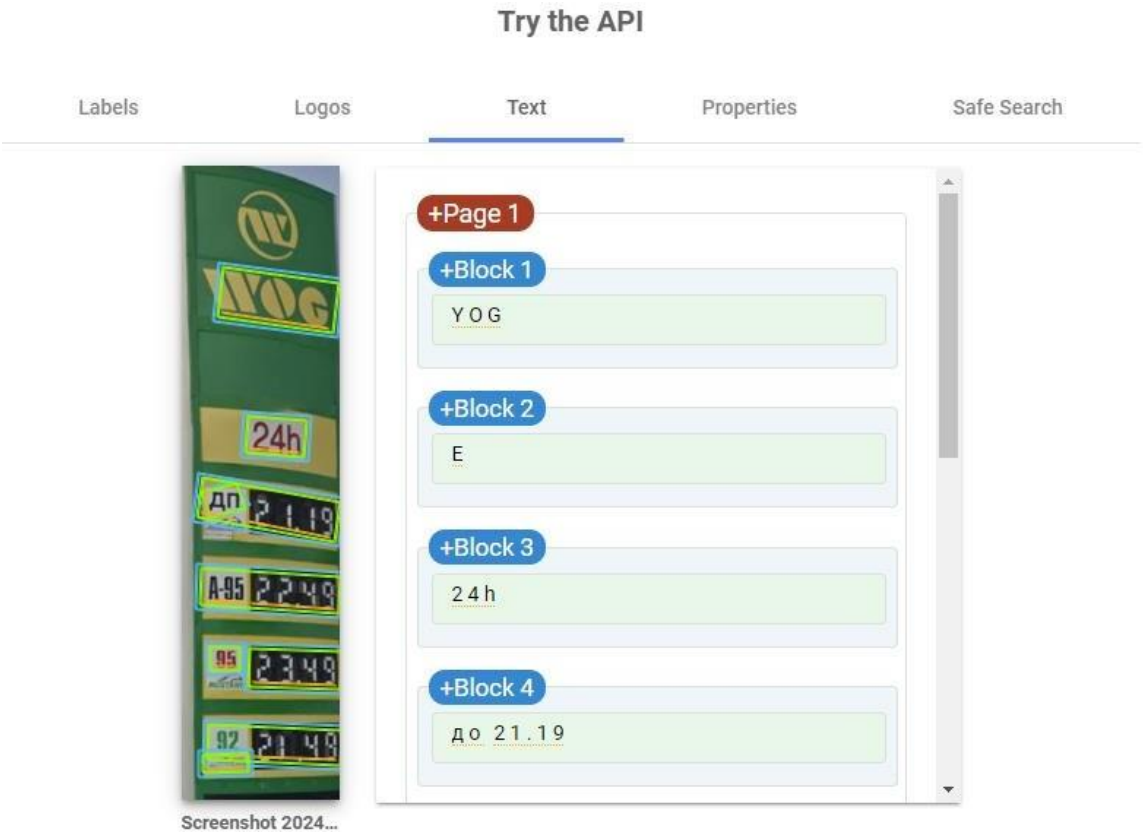


Рисунок 3.4 – Результат тестування розпізнавання тексту з Google Cloud Vision API

Згідно рисунку 3.4 можна зроби висновок, що текст на зображенні розпізнається але з 43-х символів допущено помилку в 4-х, де:

1) неправильне зчитування логотипу;
2) поява стороннього символу “E”;
3) неправильно розпізнаний символ. Символ П визначено як О;
4) опущено цифру “1” в числі 21.49.

Проаналізувавши помилки можна зробити висновок, що з усіх наявних помилок критичною є лише одна, а саме момент коли опущено цифру “1” в числі

21.49, адже такий результат безпосередньо впливає на подальший аналіз даних, при якому ціна за пальне не буде зачислена.

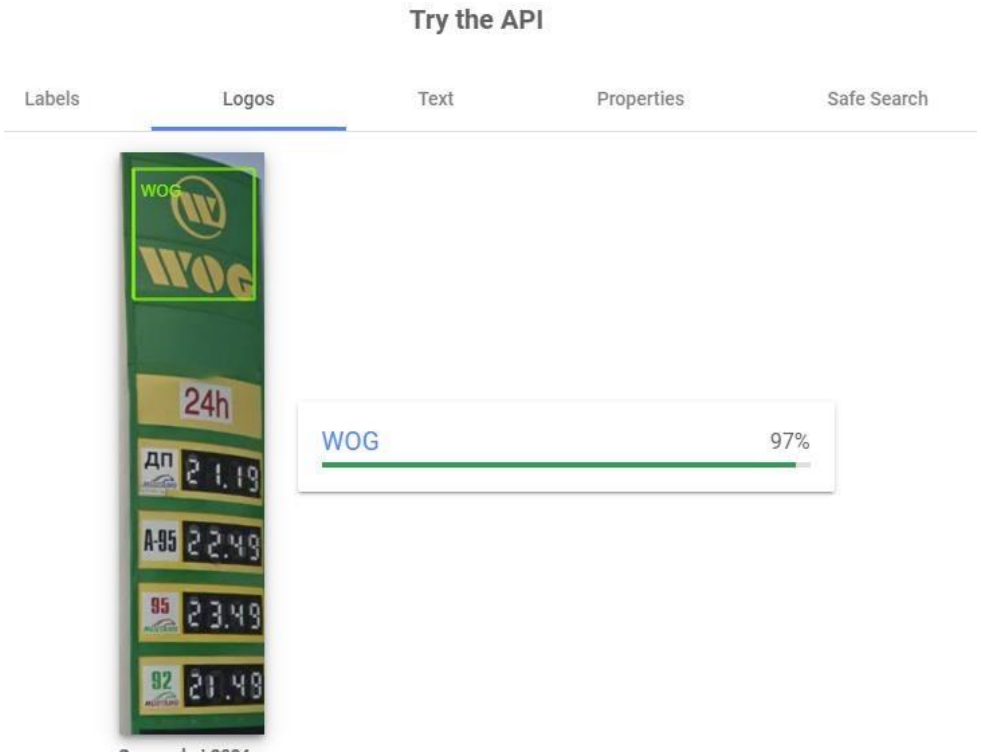


Рисунок 3.5 – Результат тестування розпізнавання логотипу з Google Cloud Vision API

Згідно рисунку 3.5 можна зроби висновок, що логотип АЗС на зображенні розпізнається з точністю 97%, що свідчить про високу точність Google Cloud Vision API в розпізнаванні логотипів.

Тепер скористаємось тестовим застосунком Tesseract і перевіримо його на точність в розпізнаванні тексту. Проаналізувавши безкоштовні версії можна дійти висновку, що жодна з доступних платформ не змогла розрізнити текст з тестового зображення з яким впорався Google Cloud Vision API.

Оскільки наша система вимагає високої точності та надійності у розпізнаванні тексту, то Google Cloud Vision API найкращий допоміжний сервус для розробки нашої КФСС.

3.3 Методи геолокаційного позиціонування

При обиранні методу геолокаційного позиціонування для периферійних вузлів слід розглянути різні технології та методи. Розглянемо загально відомі:

Глобальна система позиціонування (GPS). GPS є однією з найпоширеніших технологій для визначення геолокації. Вона базується на мережі супутників, що обертаються навколо Землі, і дозволяє точно визначати місцезнаходження в будь-якій точці на земній поверхні.

Система мобільної телефонії (Cellular triangulation). Телефонні мережі можуть також використовуватися для визначення геолокації шляхом вимірювання сигналів від різних мобільних веж у зоні дії. Цей метод може бути особливо корисним у випадках, коли GPS-сигнал недоступний, наприклад, у густонаселених місцях або всередині будівель.

Wi-Fi позиціонування. Використання Wi-Fi мережі для визначення місцезнаходження шляхом аналізу доступних точок доступу і сигналів може також бути ефективним методом геолокації, особливо у внутрішніх приміщеннях, де сигнал GPS може бути обмеженим.

Bluetooth Low Energy (BLE) Beacons. BLE маяки - це невеликі пристрої, що висилають Bluetooth сигнали і дозволяють визначати місцезнаходження відносно них. Вони можуть бути розміщені в різних точках для створення точної системи локалізації всередині приміщення.

Інерціальні системи навігації (INS). INS використовують акселерометри та гіроскопи для визначення руху і зміни місцезнаходження. Вони можуть бути корисними у випадках, коли сигнал GPS недоступний, наприклад, у тунелях або всередині будівель.

Обираючи GPS для визначення геолокаційного позиціонування периферійних вузлів КФСС, ми обираємо надійний та точний метод. GPS є широко використовуваним стандартом для визначення місцезнаходження і надає високу точність у більшості випадків. Важливо враховувати, що для успішної роботи GPS сигнал повинен бути доступний, тому це може бути обмеженням у зонах з

високими будівлями або всередині приміщень. Однак, в цих випадках можна комбінувати GPS з іншими методами, такими як Wi-Fi позиціонування або INS, для отримання більшої надійності та точності визначення місцезнаходження. Враховуючи, що наша система КФСС орієнтована на автозаправні станції, які в переважній своїй більшості розташовані на відкритій місцевості, то GPS як метод геолокаційного позиціонування добре себе покаже.

3.4 Метод автоматичного прийняття рішення підсистемою КФСС – периферійним вузлом

Метод автоматичного прийняття рішення підсистемою КФСС – периферійним вузлом, випливає з описаної моделі периферійного вузла, а саме: взаємозв'язку апаратної та програмної частини. Він необхідний для нормального та адекватного функціонування всієї підсистеми та КФСС.

Перед детальним описом методу сформуємо і опишемо умови успішного прийняття рішення підсистемою:

1) якщо периферійний вузол розпізнав об'єкт за допомогою нейронної мережі, тобто $C > 0.4$, то $f=f+1$ та $d[0] = 1$, інакше $d[0] = 0$;

2) якщо система геолокаційного позиціонування визначила місцезнаходження периферійного вузла, тобто $(x, y) \neq NULL$ і $G_{update_time} \neq NULL$, то $f=f+1$ та $d[1]=1$, інакше $d[1]=0$;

3) якщо периферійний вузол завершив обробку зображення, тобто виділив необхідну частину зображення для відправки на сервер, то $f=f+1$ та $d[2] = 1$, інакше $d[2]=0$;

4) якщо периферійний вузол має доступ до інтернету, тобто $CM_{state}=1$ і $CM_{speed} >0$, то $f=f+1$ та $d[3]=1$, інакше $d[3]=0$.

Розроблені умови для прийняття рішення щодо можливості автоматичного прийняття рішення підсистемою є основою для розроблення методу автоматичного прийняття рішення периферійним вузлом про відправлення даних визначеного об'єкта з геолокаційними даними.

Тепер розглянемо можливі дії підсистеми щодо сформованих умов в масиві *d*:

1) якщо *d[0]=0,* то підсистема не визначила жодного об’єкта і виконання наступних дій по автоматичному прийняттю рішення не має сенсу;

2) якщо *d[0]=1*, то підсистема розпізнала необхідний об’єкт і переходить до перевірки наступної умови;

3) якщо *d[1]=0,* то підсистема не визначила своє поточне місцезнаходження, а отже для передачі геолокаційних даних необхідно відтермінувати відправку даних до появи сигналу або скористатися останніми доступними даними по розташуванню транспортного засобу; після описаних дій значення *d[1]=1*;

4) якщо *d[1]=1,* то підсистема визначила найближче положення до зробленого знімку і переходить до перевірки наступної умови;

5) якщо *d[2]=0*, то підсистема не здійснила фіксацію поточного часу та кадру з відеопотоку, це ініціює повторну обробку зображення зі зміненими параметрами, допоки *d[2]=1*;

6) якщо *d[2]=1*, то підсистема завершила обробку даних і переходить до перевірки наступної умови;

7) якщо *d[3]=0,* то підсистема відкладає відправку даних щодо визначеного об’єкта та з визначеним інтервалом перевіряє чи не з’явився доступ до інтернету;

8) якщо *d[3]=1*, то підсистема відправила дані на віддалений сервер і завершує свої дії щодо визначених умов.

Згідно описаних умов та необхідних дій підсистеми сформуємо метод автоматичного прийняття рішення та відобразимо його у вигляді подальших кроків:

1) визначення цільового об’єкту(в нашому випадку це табло з цінами на АЗС) за допомогою нейронної мережі;

2) паралельна обробка даних цільового об’єкту з визначенням його геолокаційних даних;

3) перевірка сформованих умов успішного прийняття рішення підсистемою;

4) застосування можливих дій щодо сформованих умов успішного прийняття рішення по відправці даних;

5) генерація рішення на основі розроблених умов та дій щодо них:

a. Якщо f = 4, то підсистема приймає рішення по відправлені даних на сервер для подальшої обробки даних серверною підсистемою.
b. Якщо f ≠ 4, то підсистема повторно перевіряє масив умов і згідно описаних раніше дій приймає рішення.

Згідно описаного методу була сформована блок-схема алгоритму автоматичного прийняття рішення підсистемою КФСС – периферійним вузлом, яка представлена на рисунку 3.6.

Розроблений метод автоматичного прийняття рішення про відправку даних на сервер забезпечує: можливість коректної та безвідмовної роботи периферійного вузла, а також валідність та коректність даних з якими серверна підсистема КФСС буде в подальошому працювати.

3.5 Методи аналізу великих обсягів даних та виявлення аномальних значень у наборі даних

В процесі роботи КФСС, можна спрогнозувати, що під час збору даних вникатимуть неочікувані результати, коли при виявленні об’єктів буде захлопуватись сторонній об’єкт і при оптичному розпізнаванні символів відбуватиметься аналіз непотрібних даних. Для того щоб кінцевий результат роботи КФСС не спотворювався і таких ситуацій не виникало потрібно визначитись з методом виявлення хибних значень в наборі даних.

Виділяють три підходи по виявлення аномальних значень у наборі даних, це: застосування статичних методів, метод машинного навчання і візуалізація даних. Розглянемо окремо кожен з них:

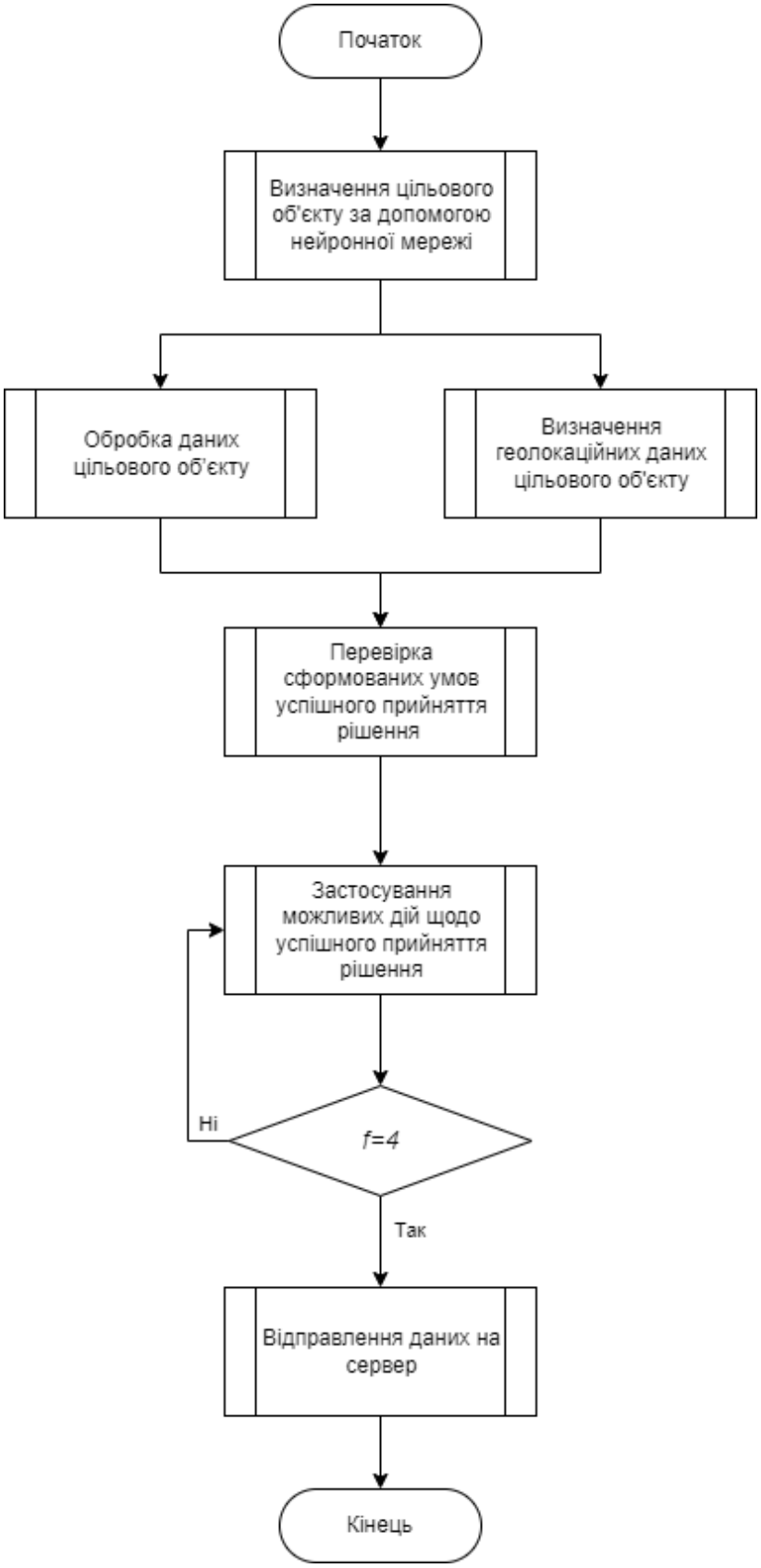


Рисунок 3.6 – Блок-схема алгоритму автоматичного прийняття рішення підсистемою КФСС – периферійним вузлом

Почнемо з найпростішого, це: візуалізація даних – це метод при якому застосовуються графіки розсіювання. Цей метод передбачає використання графіків розсіювання для візуалізації залежностей між різними змінними у наборі даних. Аномальні значення можуть бути видимими як відхилення від звичайного розподілу точок на графіку. Даний метод підходить тільки для візуального підтвердження хибних даних з набору і може бути доцільним у використанні адміністраторами КФСС, але безкорисним для автоматичного виявлення аномалій з набору.

Серед статистичних методів можна виділити метод стандартних відхилень(Z-score) та міжквартильний або інтерквартильний розмах(IQR). Обидва методи є корисними для виявлення аномалій в наборі даних, але вони мають різні застосування і особливості. Метод стандартних відхилень часто використовується в нормальних розподілах даних, оскільки він передбачає, що розподіл даних має гаусовий розподіл. В той час як метод міжквартильний розмах часто використовується, коли даних не можна припустити, що вони мають нормальний розподіл або коли є значна кількість викидів. Дані методи є основою і базою для наступного виду – методів машинного навчання.

Методи машинного навчання поділяють на два типи: кластеризацію та алгоритми виявлення викидів. Порівнюючи ці два підходи, варто враховувати, що кластеризація спрямована на групування схожих даних, тоді як алгоритми виявлення викидів виявляють незвичайні шаблони або відхилення. Обидва методи можуть бути корисними в аналізі даних, і їх вибір залежить від конкретної задачі та природи даних.

Проте для цілей які поставленні перед КФСС методи кластерізації підходять найкраще, оскільки вони дозволяють створити завчасно групу зі схожих даних по типу набору геолокаційних точок які відображатимуть регіони об’єднані за локацією та цінами на АЗС. Приклад кластеризації можна побачити на рисунку 3.7, на якому зображено дані на вході та результат роботи алгоритму.

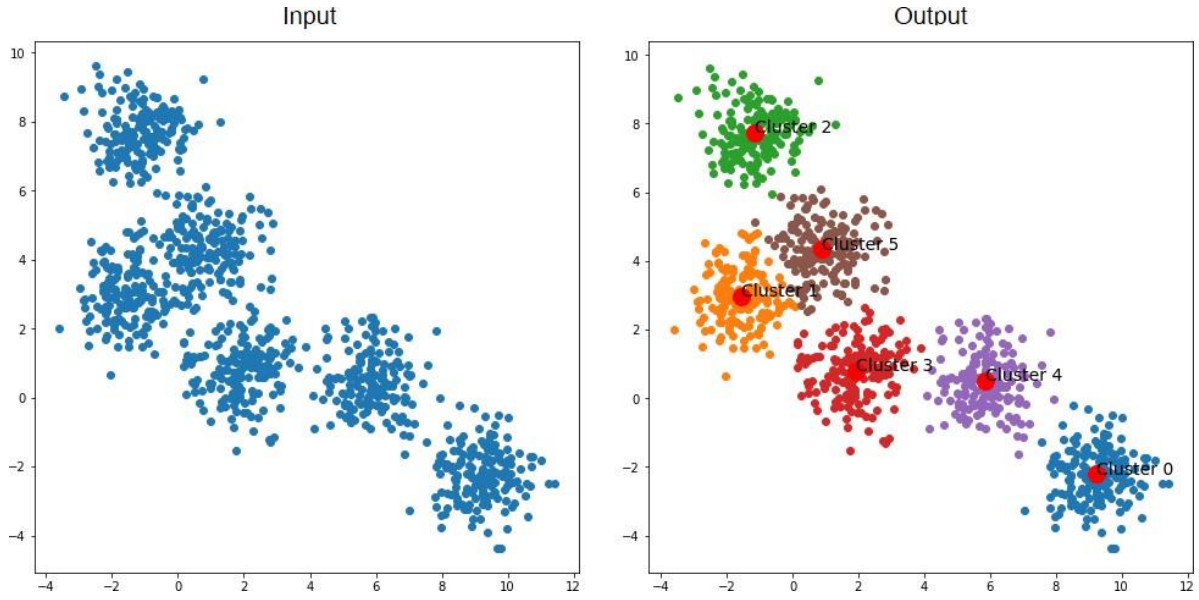


Рисунок .3.7 – Результат роботи нейронної мережі по кластеризації [81]

Основна мета кластеризації в КФСС – знайти природні структури в даних та згрупувати їх за подібністю. Виділяють цілих чотири алгоритми з кластеризації:

– центроїдне кластеризування;
– кластеризування на основі щільності;
– кластеризування на основі розподілу;
– ієрархічне кластеризування.

З цих чотирьох кластеризування на основі щільності найкраще підходить для КФСС. Цей метод дозволяє виявляти групи даних зі схожими характеристиками та відокремлювати аномальні відхилення. Крім того, такий підхід може бути ефективним у випадках, коли дані мають складну структуру та неоднорідний розподіл. Наглядний приклад роботи такого алгоритму можна побачити на рисунку 3.8.

Алгоритм кластеризації на основі щільності, такий як DBSCAN (Density-Based Spatial Clustering of Applications with Noise), виявляє кластери в даних, аналізуючи їх просторову щільність. Основна ідея полягає в тому, щоб знайти області в просторі даних, де точки розташовані густо, і визначити їх як кластери, відокремивши їх від інших областей, де точки розташовані розріджено.

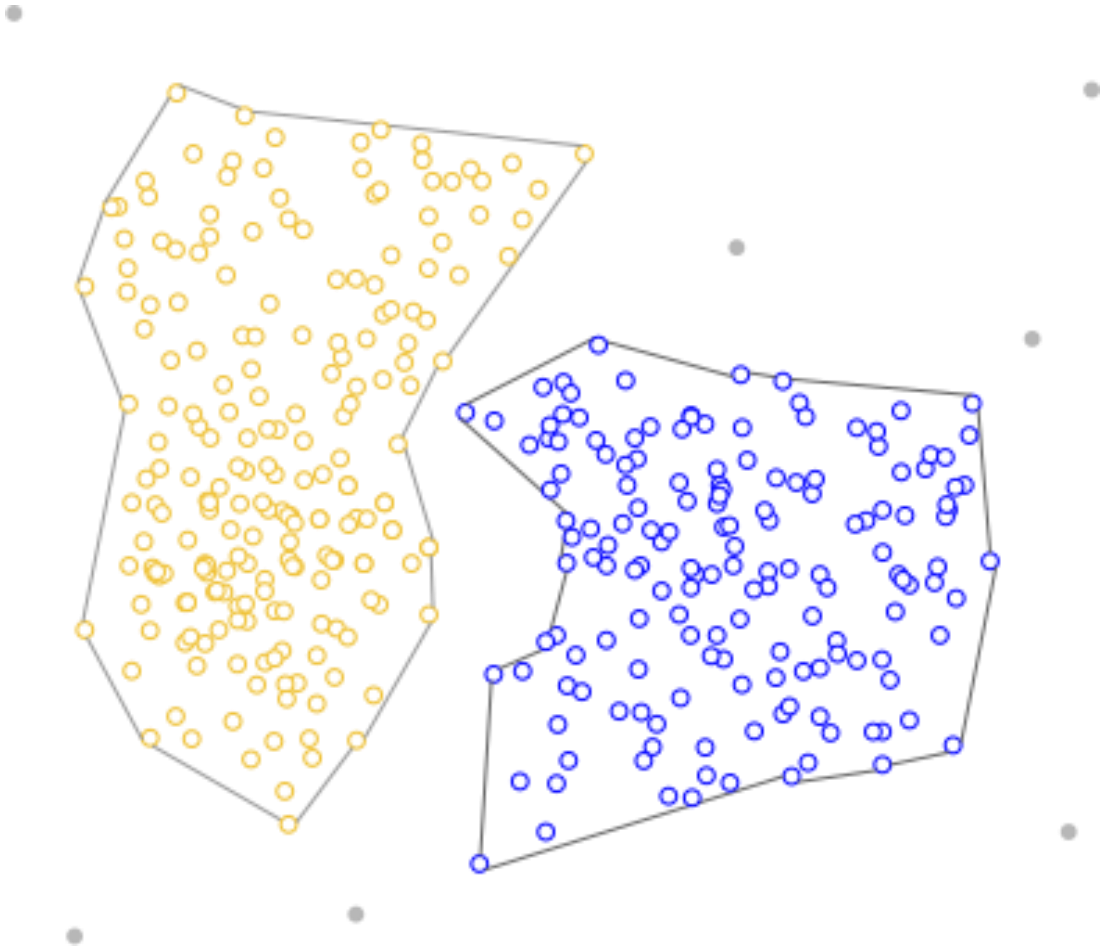


Рисунок 3.8 – Приклад кластеризації на основі щільності [82]

Алгоритм роботи DBSCAN можна описати наступним чином:

1. Визначення параметрів. Перед початком роботи з алгоритмом потрібно визначити два основні параметри: радіус ε і мінімальну кількість сусідів (MinPts). Радіус ε визначає максимальну відстань між точками, яка вважається ще одна для кожного об'єкта. Мінімальна кількість сусідів вказує, скільки точок має бути в околі радіуса ε для того, щоб точка вважалася ядром кластера.

2. Вибір початкової точки. Вибирається будь-яка невідвідана точка.

3. Пошук сусідів. Знаходяться всі точки в межах радіуса ε від початкової точки. Якщо кількість сусідів більше або дорівнює MinPts, то ця точка вважається ядром кластера.

4. Розповсюдження кластера. Якщо точка виявляється ядром кластера, то всі точки, які є її сусідами (включаючи сусідів її сусідів), також додаються до цього

кластера. Цей процес продовжується, доки не будуть відвідані всі сусіди ядра кластера.

5. Пошук нових кластерів. Після того, як усі сусіди поточного ядра кластера відвідані, алгоритм переходить до наступної невідвіданої точки і починає новий кластер, якщо вона також є ядром і має достатню кількість сусідів.

6. Визначення викидів. Точки, які не належать жодному кластеру, і не є достатньо близькими до будь-якого ядра кластера, вважаються викидами або шумом.

Такий алгоритм ефективно працює з даними, що містять кластери різних форм та розмірів, виявляючи їх автоматично без необхідності заздалегідь визначати кількість кластерів. Даний алгоритм може робити ефективний розділ між кластерами та викидами, що робить його корисним для аналізу даних з різними структурами та шумами. В нашій ситуації такий метод буде корисним при обробці великих обсягів даних з різних джерел, таких як інформація про ціни на АЗС, геолокаційні дані та інші.

3.6 Висновки

У даному розділі було проведено аналіз та вибір методів комунікації між підсистемами, методів розпізнавання образів та тексту, методів геолокаційного позиціонування та методів аналізу великих обсягів даних та виявлення аномальних значень у наборі даних.

У підрозділі "Методи комунікації між підсистемами КФСС" були розглянуті різні можливі зв'язки між підсистемами, такі як комунікація між периферійними вузлами, периферійними вузлами та серверною підсистемою, а також між серверною та клієнтською підсистемами.

У підрозділі "Методи розпізнавання образів та тексту" детально розглядалися алгоритми виявлення об'єктів на зображенні та оптичного розпізнавання символів, а також проводився аналіз сервісів оптичного розпізнавання символів, що було важливим для вибору найбільш підходящого рішення.

У розділі також розглядалися методи геолокаційного позиціонування та методи аналізу великих обсягів даних та виявлення аномальних значень у наборі даних, що є ключовими аспектами для правильного функціонування та ефективності системи.

Отже, даний розділ надає вичерпний аналіз методів, які були використані в процесі дослідження та розробки КФСС, що є важливим для розуміння його функціональності та можливостей.

4 КІБЕРФІЗИЧНА СИСТЕМА МОНІТОРИНГУ ЦІН НА ПАЛЬНЕ НА АЗС З ВИКОРИСТАННЯМ ГЕОЛОКАЦІЙНИХ ДАНИХ

В цьому розділі ми вдамося в подробиці по реалізації кожної підсистеми, опишемо технічні характеристики обраних апаратних складових та програмного забезпечення.

Згідно розроблених підсистем проведемо дослідження та проаналізуємо результати.

4.1 Периферійний вузол

4.1.1 Технічна архітектура

Для підсистеми периферійного вузла вибір апаратного забезпечення базується на розробленій в другому розділі моделі підсистеми (2.2).

В якості модуля зв’язку було обрано смартфон, який відіграє роль точки доступу до інтернету за допомогою мобільного зв’язок 4G.

В якості однопалатного комп’ютера було обрано Raspberry Pi 4 Model B 4GB. Технічні характеристики якого визначено в таблиці 4.1.

Такий міні-комп’ютер завдяки своїм характеристикам та операційній системі надає можливість створення периферійного вузла згідно визначених функціональних задач для даної підсистеми.

В якості GPS-модуля було обрано – модуль NEO-6M v2 зображення якого на рисунку 4.1 і технічні характеристики наведено в таблиці 4.2.

Цей модуль було обрано із-за простори інтеграції його з однопалатним комп’ютером Raspberry Pi.

Таблиця 4.1 – Технічні характеристики однопалатного комп’ютера

Характеристика	Опис
Частота процесора	1.5 МГц
Кількість ядер процесора	4
Частота пам'яті	3200 МГц
Максимальний об'єм пам'яті	4 GB
Вбудована мережева карта	Так
Максимальна швидкість мережевого адаптера	1000 Мбіт/с

Рисунок 4.1 – GPS модуль NEO-6M v2 [84]

Таблиця 4.2 – Технічні характеристики GPS модуля NEO-6M v2

Характеристика	Опис
Живлення	3-5 В
Чіп	NEO-6M
Антена	Керамічна
Енергонезалежна пам'ять	EEPROM
Інтерфейс	UART
Швидкість передача даних	9600 б/с

В якості камери стеження було обрано звичайну веб-камеру Logitech Webcam C270 HD. Дана камера була обрана із-за простої наявності і можливості інтеграції з однопалатним комп’ютером Raspberry Pi. Як покажуть далі результати дослідження вибір такої камери не найкращий для реалізації КФСС, із-за її обмеженості в кадрах за секунду до 30 та низької якості зображення.

В якості джерела живлення був обраний зовнішній акумулятор (павербанк) – Xiaomi Redmi Power Bank 20000mAh, технічні характеристики якого можна побачити в таблиці 4.3. Даний акумулятор був обраний із-за своєї ємності та можливості безперебійного подавання живлення однопалатному комп’ютеру, одночасно з цим маючи можливість отримувати живлення від транспортного засобу якщо він перебуває в русі.

Таблиця 4.3 – Технічні характеристики Xiaomi Redmi Power Bank 20000mAh

Характеристика	Опис
Підключення пристроїв	USBx2
Ємність, мАгод	20000
Зарядка	USB, Type-C, Micro-USB
Параметри заряду	5.1В 2.4а / 9.0В 2.0А / 12.В 1.5А
Розмір, мм	154x73.6x27.3
Вага, г	480

Тепер коли всі частини підсистеми визначено налаштуємо зв'язки між Raspberry Pi та GPS-модулем з’єднавши їх згідно схеми на рисунку 4.2. Опис з’єднань занесено в таблицю 4.4.

Усі інші з’єднання використовують спеціально відведені для них порти для живлення – USB-C, для камери звичайний – USB.

На цьому з апаратною частиною все, далі розглянемо програмне забезпечення необхідне для визначеного функціонування підсистеми.

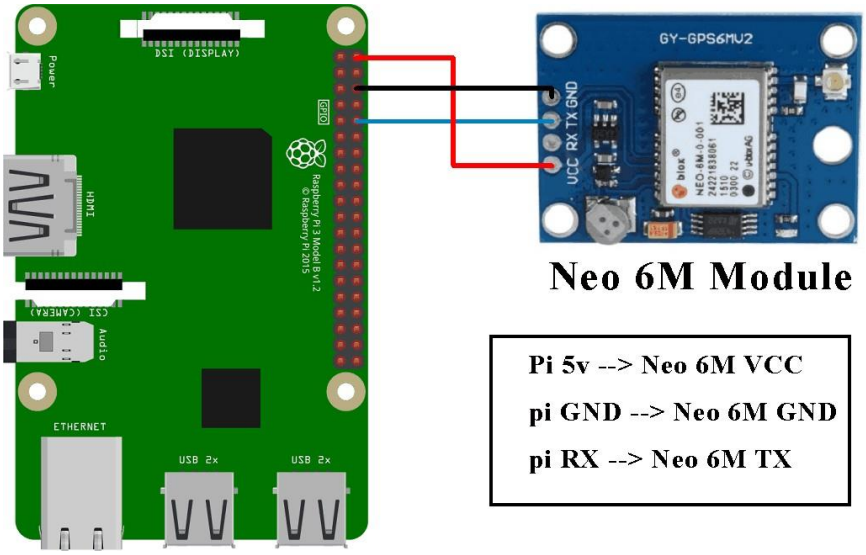


Рисунок 4.2 – Схема підключення GPS-модуля і Raspberry Pi

Таблиця 4.4 – З’єднання між Raspberry Pi і Neo 6M v2

Raspberry Pi	Neo 6M v2
5V	VCC
GND	GND
TX	RX

Згідно розробленої моделі в розділі 2 вдамося в характеристики програмного забезпечення та його налаштування. Оскільки однопалатним комп’ютером було обрано з сімейства Raspberry Pi, то і операційна система на ньому відповідна звичайна UNIX-подібна – Raspberry Pi OS. Дана ОС спеціально оптимізована для роботи на одноплатних комп'ютерах Raspberry Pi і виділяється своєю: легкістю використання, широким вибором програмного забезпечення, сумісність з різними перифирійними пристроями та безкоштовністю.

Підключений GPS-модуль потребував додаткового налаштування параметрів збірки ядра Linux і конфігурації обладнання на Raspberry Pi. Для цього в файл за шляхом “/boot/config.txt” було додано наступні параметри:

1. “dtparam=spi=on” – відповідає за увімкнення SPI (Serial Peripheral Interface) для взаємодії з пристроями, які підключені через цей інтерфейс, такими як датчики або інші мікроконтролери.

2. “enable_uart=1” – Увімкнення UART (Universal Asynchronous Receiver/Transmitter), що дозволяє здійснювати послідовний обмін даними з іншими пристроями через GPIO (General Purpose Input/Output) порти.

Далі згідно розробленого методу автоматичного прийняття рішення про відправлення даних підсистемою КФСС і описаного алгоритму дій, було написано скрипт на Python в обов’язки якого входить:

- збір даних з GPS-модуля;
- аналіз потоку відеоданих даних;
- відправка даних на сервер;
- відтермінування відправки даних на сервер.

Виокремимо з Python скрипту декілька важливих фрагментів коду які є критичними по виконанню визначених функції підсистеми:

1) за збір даних з GPS-модуля відповідає наступна функція:

```
def read_gps_data(serial_port):
    gps_data = {}
    while True:
        line = serial_port.readline().decode().strip()
        if line.startswith('$GPGGA'):
            parts = line.split(',')
            if len(parts) >= 10:
                latitude = float(parts[2][:2]) + float(parts[2][2:]) / 60
                longitude = float(parts[4][:3]) + float(parts[4][3:]) / 60
                gps_data['latitude'] = latitude
                gps_data['longitude'] = longitude
                break
    return gps_data
```

Ця функція виконує зчитування даних з GPS-модуля, підключеного до вказаного послідовного порту ‘serial_port’. Частота зчитування з післідовного порту визначена в 1 секунду, а швидкість зчитування в 9600 б/с.

2) за аналіз потоку відеоданих даних за допомого нейронної мережі відповідає наступна частина коду:

```
if confidences[] > 0.4:
   gps_data = read_gps_data(serial_port)
... Валідація інших даних ...
   best_crop = None
   best_sharpness = -1
   for xyxy in xyxys[]:
      crop_obj = frame[int(xyxy[1]):int(xyxy[3]), int(xyxy[]):int(xyxy[2])]
      sharpness = calculate_sharpness(crop_obj)
      if sharpness > best_sharpness:
         best_crop = crop_obj
         best_sharpness = sharpness
data_queue.put((gps_payload, best_crop, headers))
```

Якщо коротко, то в даній частині коду перевіряється, чи впевненість у виявленому об'єкті більше за 40 %. Якщо впевненість у виявленому об'єкті достатня, то зчитується геолокаційні дані з GPS-порту. Далі відбувається вибір найкращого зображення об'єкта. В процесі цього проходить ітерація по координатам об'єктів, вибирається обрізане зображення на основі цих координат, а потім обчислюється різкість цього зображення за допомогою функції calculate_sharpness(). Геолокаційні дані та найкраще вирізане зображення разом з заголовками додаються до черги даних.

3) за відправку даних про визначений об’єкт відповідає наступна функція:

```
def send_data_worker():
    while True:
        gps_payload, image_payload, headers = data_queue.get()
        gps_sent = send_gps_data_to_server(gps_payload, headers)
        image_sent = send_image_data_to_server(image_payload, headers)
        if not (gps_sent and image_sent):
            print("Failed to send data, will retry later...")
            data_queue.put((gps_payload, image_payload, headers))
            time.sleep(60)  # Retry after 1 minute
        data_queue.task_done()
```

Дана функція відповідає за надсилання даних на сервер, а у випадку невдалої спроби надсилання дані відправляються до черги для повторної спроби пізніше, а черга отримує повідомлення про завершення обробки кожного набору даних.

Як видно з вище описаних функцій визначений метод автоматичного прийняття рішення периферійним вузлом, був реалізований на практиці та застосований в подальшому тестуванні КФСС.

4.1.2 Тренування нейронної мережі YOLO по розпізнаванню образів

В якості нейронної мережі по розпізнаванню образів була обрана YOLOv8 від Ultralytics. Тренування моделі відбувалося на наборі даних зі 100-а знімків, на попередньо навченій моделі – ‘yolov8n.pt’. Навчання відбувалося протягом 100-ні епох. Результати тренування відображені на рисунку 4.3.

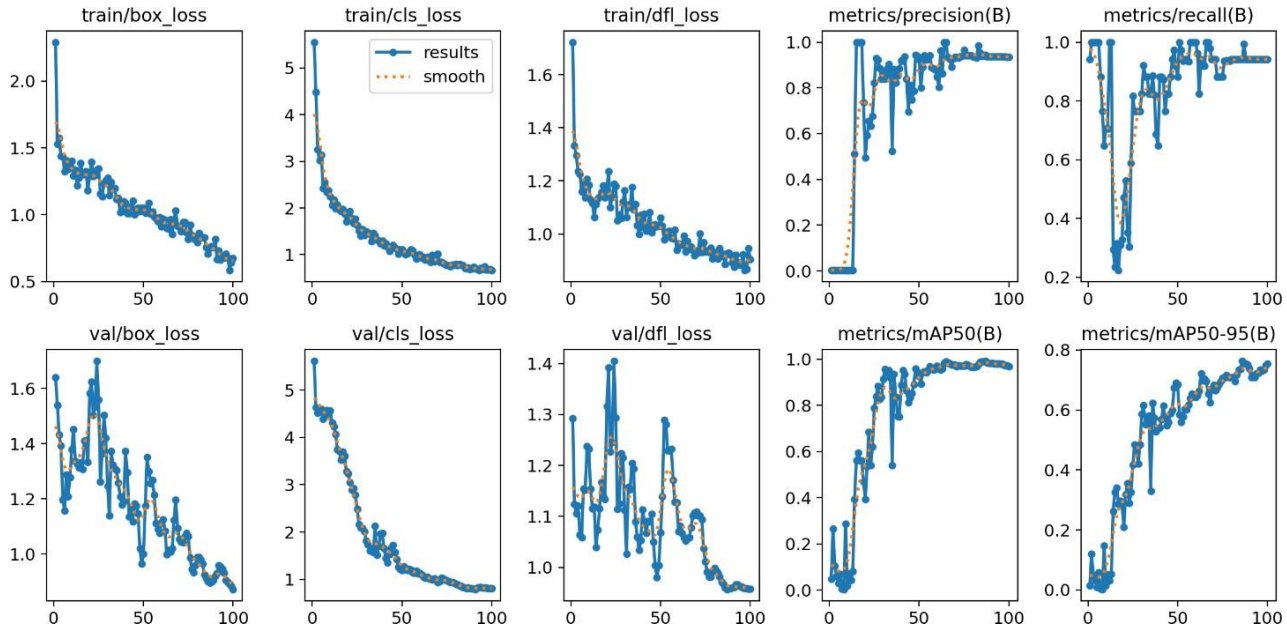


Рисунок 4.3 – Результати тренування моделі по розпізнаванню табло цін АЗС

Навчену модель було конвертовано в спеціальний формат Tensorflow Lite, який був розроблений спеціально для розгортання моделей машинного навчання на мобільних пристроях та пристроях інтернету речей. Даний тип моделі був обраний із-за високого навантаження на процесор при використанні PyTorch моделі.

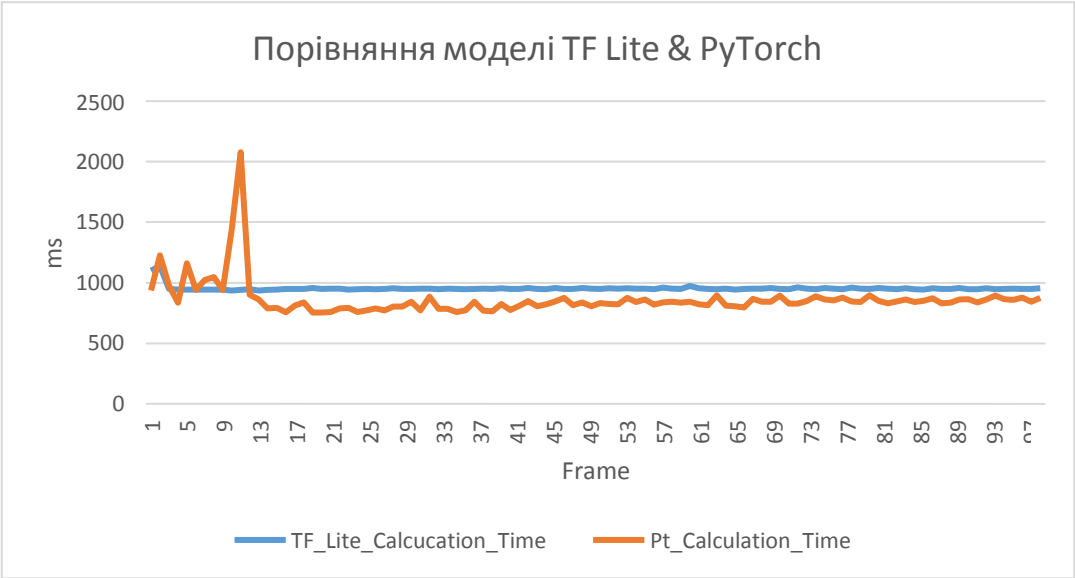


Рисунок 4.4 – Порівняння швидкості моделі TF Lite і PyTorch

Згідно графіку порівняння швидкості моделі TF Lite і PyTorch на рисунку 4.4 можна дійти висновку, що модель TF Lite стабільна в обрахунках але в середньому використовує більше часу для визначення об’єкта. Середній час на обрахунок моделлю TF Lite становить 953.033 мс, коли модель PyTorch в середньому використовує 863.844 мс.

Проте також варто відзначити, що при використанні PyTorch моделі, завантаження процесора Raspberry Pi варіювалося від 360 до 390 % і це свідчить про те, що модель використовує більше одного ядра процесора для обчислень, що в подальшому призводитиме до браку ресурсів в підсистемі і неможливості виконанні визначених функціональних вимог. В той самий момент при використанні моделі TF Lite середня завантаженість процесора Raspberry Pi становила від 110 до 120 %, що свідчить про ефективний розподіл ресурсів даною моделлю.

4.1.3 Результати роботи підсистеми

Підсистема КФСС для моніторингу цін на АЗС була тестувалася на трьох видах даних:

– фото;
– записах відео;
– відеопотоках.

При аналізі фото, з десяти фото підсистема визначила 8 з 10 табло з цінами на АЗС. Результати цих тестів занесено в таблицю 4.5, яка відображає номер зображення і “впевненість” в опізнаному об’єкті.

Таблиця 4.5 – Результати тестування підсистеми на фото

№ Фото	1	2	3	4	5	6	7	8	9	10
Впевненість	0.71	0.81	0.75	-	0.83	0.56	-	0.44	0.39	0.55

Як видно з результатів тестування з 10-и знімків, які містили табло з цінами АЗС, нейронна мережа розпізнала тільки 8. Дані обрані для тестування включали як і фото подібних АЗС на яких тренувалася нейронна мережа, так і абсолютно нові типи. Фото №1-3, 4 – це фото схожих АЗС з тренування нейронної мережа, а фото №4, 6-10 – абсолютно нові АЗС, які нейронна мережа опізнала вперше. Саме тому, видно таку розбіжність в показнику “Впевненості”.

Проаналізувавши результати тестування фото знімків в якості цілі для подальшого дослідження підсистеми було обрано АЗС на яких тренувалася нейронна мережа.

Під час експерименту на дослідження роботи підсистеми з записами відео було відібрано 7 відеозаписів на яких записана інформація з відеореєстратора транспортного засобу. На відеозаписах автомобіль рухається повз АЗС біля яких стоїть табло з цінами. З 7-ми відеозаписів підсистема розпізнала табло з цінами на 6-х. Відеозапис на якому підсистема не розпізнала табло виділявся темною порою доби і це свідчить про недостатню навченість моделі по розпізнаванню образів в погано освітленій місцевості.

Враховуючи попередньо описаний недолік підсистеми по розпізнаванню образів в погано освітленій місцевості, дослідження роботи підсистеми з відеопотоками краще проводити в добре освітлювальний день. З порівняльного аналізу нейронної мережі(рисунок 4.3) можна зробити висновок, що підсистема може обробляти відеопотік з швидкістю 1.049 кадр за секунду, а отже швидкість руху транспортного засобу прямо впливає на можливість підсистеми визначити об’єкт. Для дослідження підсистеми на можливість роботи з відеопотоками було визначено, що найкраще дотримуватись швидкості руху автомобілем 40 км/год(11.1 м/с), це обмеження дозволить краще проаналізувати працездатність підсистеми.

Під час експерименту на дослідження роботи підсистеми з відеопотоками було здійснено 3 однакових виїзди, під час яких автомобіль здійснював рух повз 4 АЗС. Результати експерименту занесено в таблицю 4.6, в якій “1” – позначає, що підсистема розпізнала табло АЗС, а “0” – не розпізнала об’єкт.

Таблиця 4.6 – Результати тестування підсистеми з відеопотоками

№ Експерименту	№ АЗС			
	1	2	3	4
1	1	1	1	0
2	1	1	1	1
3	1	1	1	0

Згідно результатів в таблиці 4.6 можна дійти висновку, що підсистема проявила стабільність та точність у визначенні об’єктів згідно яких була навчена модель нейронної мережі. Підсистема опізнала 83.3% з усього набору даних. Також варто відзначити, що АЗС №4 мала унікальне табло з цінами, що вплинуло на результати.

Під час експерименту на дослідження роботи підсистеми з відеопотоками серйозних затримок GPS-модуля та доступу до інтернету не було спостережено. Під час трьох експериментів з таблиці 4.6 знімки цін та їх метадані надійшли на сервер з 1-2 секундною затримкою. Проте, варто відзначити, що всі експерименти проводилися в окрузі міста і з визначеною швидкістю транспортного засобу, тому не виключено, що підсистема КФСС може показати гірші показники у взаємодії з серверною підсистемою за околицями міста.

4.2 Серверна підсистема

4.2.1 Технічна архітектура

Для серверної підсистеми вибір апаратного забезпечення базується на розробленій в другому розділі моделі підсистеми (2.9). Згідно розробленої моделі було визначено, що для серверної підсистеми найкраще підійде мікросервісна архітектура, де один мікросервіс відповідає за окремий функціонал підсистеми який був визначений математичною моделлю.

Для розміщення мікросервісів було обрано хмарне середовище Google Cloud Platform(GCP), оскільки воно надає платформу для розробки та хостингу вебзастосунків, а також, раніше визначений, сервіс оптичного розпізнавання символів Google Cloud Vision API.

В хмарному середовищі GCP було обрано екземпляр віртуальної машини технічні характеристики якої визначено в таблиці 4.7.

Для збереження результатів по типу зображень було використано Google Cloud Storage, а для збереження метаданих для зображень базу даних Mongo DB, яку розвернуто на окремій і віртуальній машині в GCP, технічні характеристики якої занесено в таблицю 4.8.

Таблиця 4.7 – Технічні характеристики віртуальної машини для серверної підсистеми

Модель	E2
vCPU	4(2 ядра)
ОП	16 Gb
Система	Debian GNU/Linux 12 (bookworm)

Кінцевий результат з тим як виглядають екземпляри віртуальних машин зображено на рисунку 4.5.

Таблиця 4.8 – Технічні характеристики віртуальної машини для БД

Модель	E2-medium
vCPU	1-2 vCPU (1 shared core)
ОП	4 Gb
Система	Debian GNU/Linux 12 (bookworm)

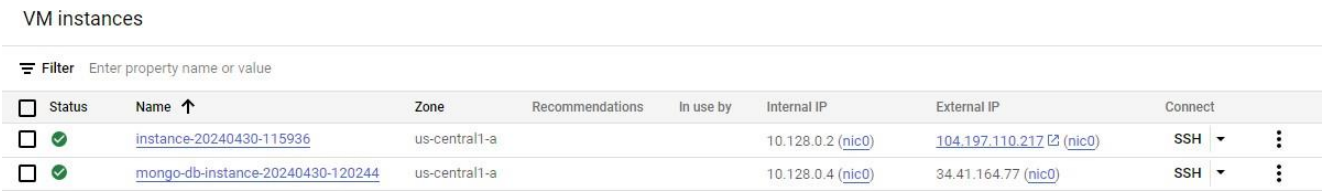


Рисунок 4.5 – Екземпляри віртуальних машин

З метою тестування та дослідження роботи підсистеми було розроблено чотири мікросервіси які виконують функції визначені математичною моделлю (2.9), а саме:

– ‘api-gateway’ –програмний інтерфейс який надає можливість взаємодіяти з серверною підсистемою, являє собою веб-сервіс який використовує порт 8080;

– ‘ed-checkout’ – сервіс який відповідає за обробку вхідних даних та взаємодію з периферійними вузлами, являє собою веб-сервіс який використовує порт 8081;

– ‘orc-service’ – сервіс який відповідає за розпізнавання тексту на зображені, являє собою веб-сервіс який використовує порт 8082;

– ‘da-service’ – сервіс для аналітики великих даних, являє собою веб-сервіс який використовує порт 8083.

Для зручного розгортання цих мікросервісів було використано утиліту Docker. Кожен мікросервіс був упакований у власний контейнер, і всі контейнери були об'єднані в одній віртуальній мережі 'net'. Це дозволило забезпечити ізоляцію, масштабованість та ефективну взаємодію між сервісами підсистеми.

4.2.2 Результати роботи підсистеми

Аналіз результатів роботи підсистеми складається з двох частин: аналіз результатів оптичного розпізнавання та аналіз результатів динамічної кластеризації.

Для аналізу результатів оптичного розпізнавання серверною підсистемою було обрано 5 зображень на яких периферійний вузол визначив табло з цінами на АЗС. Сервіс оптичного розпізнавання символів Google Cloud Vision розпізнав текст на всіх зображеннях, проте з втратою певної інформації. Результати експериментів винесено в таблицю 4.9, де визначено: номер зображення, кількість Правильно визначених символів і кількість символів які необхідно було визначити для подальшої роботи підсистеми.

Як видно з результатів оптичного розпізнавання при кожному експерименті спостерігається втрата певної кількості інформації, а саме:

- для зображення № 1 ≈ 9.375 %;
- для зображення № 2 ≈ 16.7%;
- для зображення № 3 ≈ 20.84%;
- для зображення № 4 ≈ 5.71%;
- для зображення № 5 ≈ 11.11%.

Таблиця 4.9 –Результати тестування підсистеми на оптичне розпізнавання символів

№ Зображення	Правильно визначено	Необхідно було визначити
1	29	32
2	20	24
3	19	24
4	33	35
5	16	18

Загальна точність оптичного розпізнавання становить 87.97%, що є критичним показником і суттєво впливає на подальший результати роботи підсистеми, а отже звичайного використання сервісу по розпізнаванню тексту недостатньо для повноцінного функціонування підсистеми і вона вимагає розробки

нового методу і алгоритму, який міг би відновлювати втрачену інформацію з урахування очікуваного результату та отриманого результату.

Для кластеризації даних було обрано алгоритм DBSCAN. В якості даних для кластеризації були використані геолокаційні дані АЗС та статистичні дані щодо цін на пальне за березень 2024 року, отримані з сайту “Мінфін” [83]. Дані були зібрані з трьох міст: Хмельницького, Києва та Харкова.

В першому випадку для кластерного аналізу даних було обрано широту та довготу АЗС. В якості вхідних параметрів для DBSCAN було обрано параметр ε=0.15, він визначає максимальну відстань між двома точками, щоб вони вважалися сусідніми та параметр ‘min_samples= 15’ який визначає мінімальну кількість сусідів, які повинна мати кожна точка, щоб вона вважалася ядром кластера.

В результаті роботи підсистеми було проаналізовано 295 екземплярів вхідних даних які було розподіллено на 3 кластери які містили в собі інформацію про: ціни на пальне, вид пального та геолокаційні дані АЗС. Візуальну репрезентацію результатів кластеризації зображено на рисунку 4.6 та 4.7. Рисунок 4.6 відображає двовимірний простір з широтою та довготою, де кожна точка – це окрема АЗС, а колір – окремий кластер.

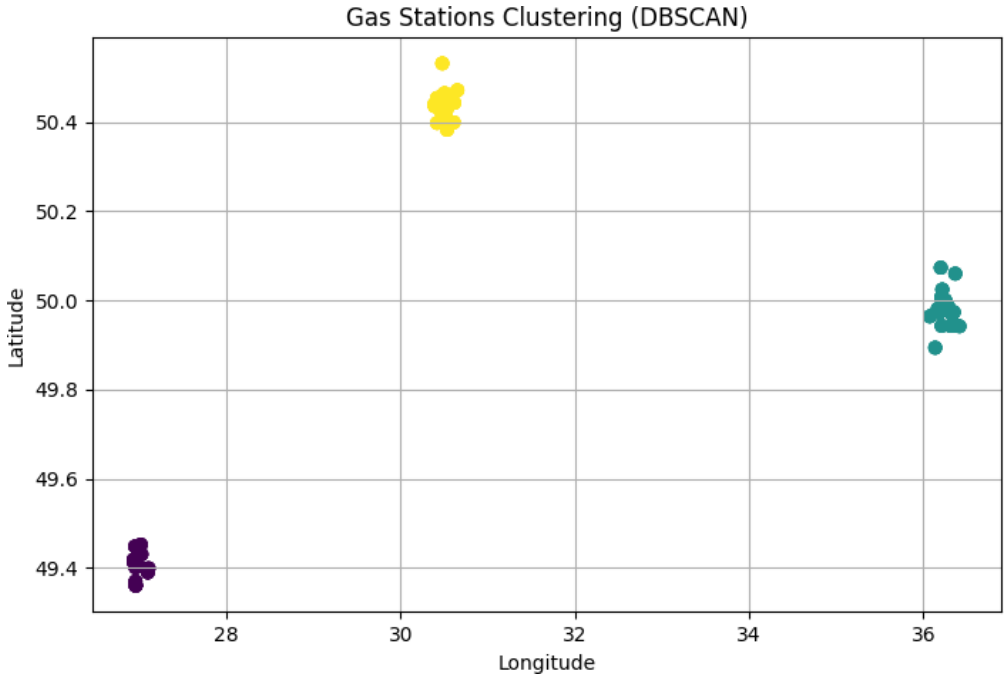


Рисунок 4.6 – Результати кластеризації за координатами АЗС в 2D

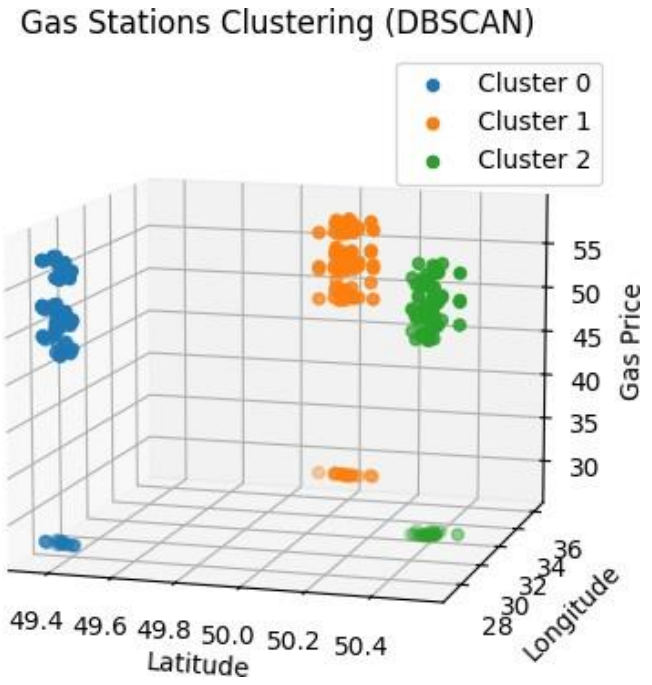


Рисунок 4.7 – Результати кластеризації за координатами АЗС в 3D

Рисунок 4.7 відображає тривимірний простір з широтою, довготою, та ціною на пальне. де кожна точка – це окрема ціна на пальне, а колір – окремий кластер. В даному прикладі: ‘Cluster 0’ – відображає ціни в м. Хмельницький, ‘Cluster 1’ – відображає ціни в м. Харків, “Cluster 2” – відображає ціни в м. Київ.

В другому випадку для кластерного аналізу даних було використано ціни на АЗС. В якості вхідних параметрів для DBSCAN було обрано ε=0.028, ‘min_samples=3’.

В результаті роботи підсистеми було проаналізовано 295 ціни які було розподіллено на 5 кластерів які містили в собі інформацію про ціни на пальне та геолокаційні дані АЗС. Візуальну репрезентацію результатів кластеризації зображено на рисунку 4.8. Очікуваним результатом мала бути матриця з рівномірно розподіленими цінами по 59 цін кожного виду пального, а в результаті вийшло, що пропорція не була збережена, а отже гіпотеза про те що для ефективної роботи КФСС можна використовувати метод кластеризації DBSCAN для визначення приналежності даних до одного виду – спростована.

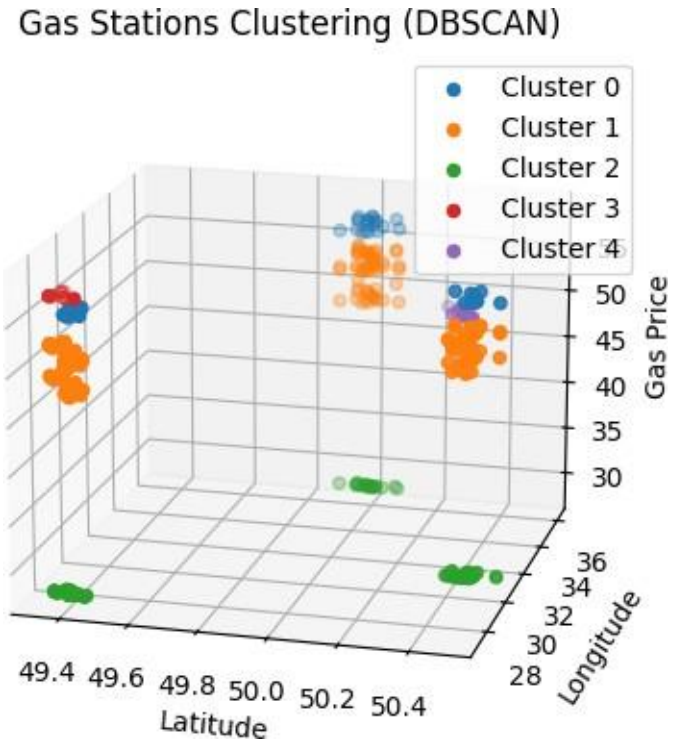


Рисунок 4.8 – Результати кластеризації за цінами на АЗС в 3D

В нашому випадку для ефективного розподілу цін по класам краще буде використати метод найближчого сусіда або як його ще називають – алгоритм класифікації на основі відстаней. Результати використання кластеризації та алгоритму класифікації на основі відстаней винесено в таблицю 4.10.

Таблиця 4.10 – Результати тестування підсистеми на кластеризацію даних за цінами на АЗС

Очікуваний результат	Кластеризація DBSCAN	Метод найближчого сусіда
[59, 59, 59, 59, 59]	[40, 177, 59, 9, 10]	[61, 59, 59, 59, 57]

4.3 Висновки

У даному розділі кваліфікаційної роботи вперше розглянуто технічну архітектуру двох підсистем КФСС, сформовано результати роботи підсистем та проаналізовано їх. Під час опису технічної архітектури підсистем було розглянуто

апаратну та програмну складову кожної системи, на основі яких проводилися подальші дослідження КФСС. Також під час опису периферійного вузла було розглянуто процес тренування нейронної мережі, яка застосовувалася в подальшій роботі підсистеми.

Згідно результатів роботи підсистем можна сформувати наступні висновки:

1. Щодо периферійного вузла на основі практичних результатів:

– підтверджено гіпотезу, що для ефективного розпізнавання образів в КФСС можна використовувати нейронну мережу YOLO;

– підтверджено гіпотезу щодо доцільності та необхідності методу автоматичного прийняття рішення підсистемою КФСС – периферійним вузлом.

2. Щодо серверної підсистеми на основі практичних результатів:

– спростовано гіпотезу, що для ефективного розпізнавання символів достатньо використовувати лише один сервіс Google Cloud Vision API, та висунуто нову гіпотезу щодо необхідності розробки методу відновлення втраченої інформації з урахування очікуваного та отриманого результатів;

– спростовано гіпотезу, що для розподілу цін по приналежності до типу пального можна використовувати метод кластеризації DBSCAN і запропоновано інший метод класифікації цін на основі відстаней.

ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень розроблено підсистеми кіберфізичної системи моніторингу цін на АЗС.

У першому розділі проведений огляд застосувань кіберфізичних систем в різних сферах діяльності та виокремлено кіберфізичні системи в сфері інтернету транспортного засобу як основний орієнтир в подальшій кваліфікаційній роботі. Проведено порівняльний аналіз переваг та недоліків існуючих рішень, в яких розглядалися наукові роботи з сфери інтернету транспортних засобів та кіберфізичних систем спостереження, на основі яких визначено методологічні підходи та здійснена постановка задачі.

У другому розділі розроблено модель кіберфізичної системи спостереження, в якій її було розділено на три підсистеми та розписано складові кожної з підсистем в окремо розроблених моделях. Також в даному розділі було розроблено модель основного процесу та часову модель КФСС.

У третьому розділі визначено необхідну методологію для реалізації кіберфізичної системи спостереження, а саме методи комунікації між підсистемами; методи розпізнавання образів і тексту; методи визначення геолокаційного позиціонування; методи аналізу великих даних. В даному розділі було сформовано новий метод автоматичного прийняття рішення підсистемою КФСС – периферійним вузлом, необхідність якого зумовлена технічними обмеженнями визначених методів комунікації між підсистемами.

У четвертому розділі розглянуто технічну архітектуру двох підсистем кіберфізичної системи спостереження, кожну з них реалізовано та досліджено на практиці, за результатами якої підтверджено дві гіпотези, спростовано дві гіпотези та висунуто одну нову гіпотезу.

Наукова новизна отриманих результатів:

1) вперше запропоновано та розроблено кіберфізичну систему спостереження в середовищі з обмеженою швидкістю інтернету, основна ціль якого – це моніторинг корисної інформації;

2) вперше розроблено метод автоматичного прийняття рішення периферійним вузлом, який забезпечує безвідмовну роботу підсистеми кіберфізичної системи спостереження.

Практична значущість отриманих результатів полягає в отриманих практичних результатах дослідження підсистем кіберфізичної системи моніторингу цін на пальне на АЗС з використанням геолокаційних даних, які надають вичерпний аналіз методів підсистем та результати їх роботи.

За темою дипломної роботи опубліковано одні тези в журналі “Молодіжної наукової ліги” в рамках конференції “Міжнародна студентська наукова конференція «Тренди та перспективи розвитку мультидисциплінарних досліджень»” (26 квітня 2024 р., м. Одеса, Україна) та прийнята одна стаття до публікації англійською мовою у міжнародний науковий журнал Хмельницького національного університету «Computer systems and information technologies»:

1) М. С. Бойко, В. В. Яцків. Кіберфізична система спостереження як частина інтернету транспортних засобів. *Міжнародна студентська наукова конференція «Тренди та перспективи розвитку мультидисциплінарних досліджень»*. Одеса, Україна, 2024. С. 159-161.

2) M. Boiko, V. Yatskiv. Surveillance cyber-physical system as a part of internet of vehicles. Computer systems and information technologies. *ISSN 2710-0766.* 2024. Vol. 2.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Lampropoulos G., Siakas K., Anastasiadis T. Internet of Things in the Context of Industry 4.0: An Overview. *International Journal of Entrepreneurial Knowledge*. 2019. Vol. 7, no. 1. P. 4–19.

2. Schroth C. The internet of services: Global industrialization of information intensive services. *International Conference on Digital Information Management.* 2007, P. 635-642

3. Shafiq S. I., Szczerbicki E., Sanin C. Manufacturing Data Analysis in Internet of Things/Internet of Data (IoT/IoD) Scenario. *Cybernetics and Systems*. 2018. Vol. 49, no. 5-6. P. 280–295.

4. Systems Engineering for Industrial Cyber–Physical Systems Using Aspects / I. Akkaya et al. *Proceedings of the IEEE*. 2016. Vol. 104, no. 5. P. 997–1012. URL: https://doi.org/10.1109/jproc.2015.2512265 (date of access: 09.05.2024).

5. Distributed Control of Cyber Physical System on Various Domains: A Critical Review / M. Hamzah et al. *Systems*. 2023. Vol. 11, no. 4. P. 208.

6. Increasing the Production System Productivity Using Inertial Navigation / P. Božek et al. *Manufacturing Technology*. 2015. Vol. 15, no. 3. P. 274–278.

7. Pintér T., Božek P. Industrial Robot Control Using Inertial Navigation System. *Advanced Materials Research*. 2012. Vol. 605-607. P. 1600–1604.

8. Kazancoglu Y., Ozkan-Ozen Y. D. Analyzing Workforce 4.0 in the Fourth Industrial Revolution and proposing a road map from operations management perspective with fuzzy DEMATEL. *Journal of Enterprise Information Management*. 2018. Vol. 31, no. 6. P. 891–907.

9. Applications of Wireless Sensor Networks and Internet of Things Frameworks in the Industry Revolution 4.0: A Systematic Literature Review / M. Majid et al. *Sensors*. 2022. Vol. 22, no. 6. P. 2087. URL: https://doi.org/10.3390/s22062087.

10. Abulamddi M. F. H. A Survey of Approaches Reconciling between Safety and Security Requirements Engineering for Cyber-Physical Systems. *Journal of*

Computer and Communications. 2017. Vol. 05, no. 01. P. 94–100. URL: https://doi.org/10.4236/jcc.2017.51008.

11. A fog computing industrial cyber-physical system for embedded low-latency machine learning Industry 4.0 applications / P. O'Donovan et al. *Manufacturing Letters*. 2018. Vol. 15. P. 139–142. URL: https://doi.org/10.1016/j.mfglet.2018.01.005

12. Domingo M. C. An overview of the Internet of Things for people with disabilities. *Journal of Network and Computer Applications*. 2012. Vol. 35, no. 2. P. 584–596. URL: https://doi.org/10.1016/j.jnca.2011.10.015.

13. Park K.-J., Zheng R., Liu X. Cyber-physical systems: Milestones and research challenges. *Computer Communications*. 2012. Vol. 36, no. 1. P. 1–7. URL: https://doi.org/10.1016/j.comcom.2012.09.006.

14. Sampigethaya K., Poovendran R. Aviation Cyber–Physical Systems: Foundations for Future Aircraft and Air Transport. *Proceedings of the IEEE*. 2013. Vol. 101, no. 8. P. 1834–1855. URL: https://doi.org/10.1109/jproc.2012.2235131.

15. Huang Y., Zhao M., Xue C. J. Joint WCET and Update Activity Minimization for Cyber-Physical Systems. *ACM Transactions on Embedded Computing Systems*. 2015. Vol. 14, no. 1. P. 1–21. URL: https://doi.org/10.1145/2680539.

16. Squire R., Song H. Cyber-physical systems opportunities in the chemical industry: A security and emergency management example. *Process Safety Progress*. 2014. Vol. 33, no. 4. P. 329–332. URL: https://doi.org/10.1002/prs.11676.

17. Performance Assessment of State Estimation in Cyber-Physical Energy Systems / B. H. Hassan et al. *2021 IEEE Madrid PowerTech*, Madrid, Spain, 28 June – 2 July 2021. 2021. URL: https://doi.org/10.1109/powertech46648.2021.9494760.

18. Cheng P., Shi L., Sinopoli B. Guest Editorial Special Issue on Secure Control of Cyber-Physical Systems. *IEEE Transactions on Control of Network Systems*. 2017. Vol. 4, no. 1. P. 1–3. URL: https://doi.org/10.1109/tcns.2017.2667233.

19. Depression Identification Using Machine Learning Classifiers / S. Srivastava et al. *International Journal of Innovative Research in Computer Science and Technology*. 2023. Vol. 11, no. 6. P. 1–5. URL: https://doi.org/10.55524/ijircst.2023.11.6.1.

20. Gürdür Broo D., Boman U., Törngren M. Cyber-physical systems research and education in 2030: Scenarios and strategies. *Journal of Industrial Information Integration*. 2021. Vol. 21. P. 100192. URL: https://doi.org/10.1016/j.jii.2020.100192.

21. Dafflon B., Moalla N., Ouzrout Y. The challenges, approaches, and used techniques of CPS for manufacturing in Industry 4.0: a literature review. *The International Journal of Advanced Manufacturing Technology*. 2021. Vol. 113, no. 7-8. P. 2395–2412. URL: https://doi.org/10.1007/s00170-020-06572-4.

22. Jen-Yi C., Chuan-His L., Yi-Hsin Y. The Study of Indigenous Students' Learning Effect on Geometry Course with CPS Mobile Learning and Atayal Culture. *International Journal of Learning and Teaching*. 2020. P. 1–6. URL: https://doi.org/10.18178/ijlt.6.1.1-6.

23. Human–Cyber–Physical Systems (HCPSs) in the Context of New-Generation Intelligent Manufacturing / J. Zhou et al. *Engineering*. 2019. Vol. 5, no. 4. P. 624–636. URL: https://doi.org/10.1016/j.eng.2019.07.015.

24. Convergence of IoT and CPS in Robotics / P. Jain et al. *Advances in Science, Technology & Innovation*. Cham, 2021. P. 15–30. URL: https://doi.org/10.1007/978-3-030-66222-6_2.

25. Feng Y., An X., Li S. Application of context-aware in intelligent transportation CPS. *2017 36th Chinese Control Conference (CCC)*, Dalian, China, 26–28 July 2017. 2017. URL: https://doi.org/10.23919/chicc.2017.8028553.

26. A2CPS: A Vehicle-Centric Safety Conceptual Framework for Autonomous Transport Systems / J. K. Naufal et al. *IEEE Transactions on Intelligent Transportation Systems*. 2018. Vol. 19, no. 6. P. 1925–1939. URL: https://doi.org/10.1109/tits.2017.2745678.

27. Security issues in Internet of Vehicles (IoV): A comprehensive survey / H. Taslimasa et al. *Internet of Things*. 2023. P. 100809. URL: https://doi.org/10.1016/j.iot.2023.100809.

28. Intrusion Detection Method for Internet of Vehicles Based on Parallel Analysis of Spatio-Temporal Features / L. Xing et al. *Sensors*. 2023. Vol. 23, no. 9. P. 4399. URL: https://doi.org/10.3390/s23094399 .

29. AI-based Intrusion Detection for Intelligence Internet of Vehicles / D. Man et al. *IEEE Consumer Electronics Magazine*. 2021. P. 1. URL: https://doi.org/10.1109/mce.2021.3137790.

30. Intrusion Detection System Using machine learning Algorithms / R. Tahri et al. *ITM Web of Conferences*. 2022. Vol. 46. P. 02003. URL: https://doi.org/10.1051/itmconf/20224602003.

31. Connected and Automated Vehicles: Infrastructure, Applications, Security, Critical Challenges, and Future Aspects / M. Sadaf et al. *Technologies*. 2023. Vol. 11, no. 5. P. 117. URL: https://doi.org/10.3390/technologies11050117.

32. Philip Chen C. L., Zhang C.-Y. Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences*. 2014. Vol. 275. P. 314–347. URL: https://doi.org/10.1016/j.ins.2014.01.015 .

33. Hock Soon N., Sundararajan N., Saratchandran P. ABR traffic management using minimal resource allocation (neural) networks. *Computer Communications*. 2002. Vol. 25, no. 1. P. 9–20. URL: https://doi.org/10.1016/s0140-3664(01)00331-0 .

34. Paganini F., Doyle J., Low S. Scalable laws for stable network congestion control. *40th Conference on Decision and Control*, Orlando, FL, USA. URL: https://doi.org/10.1109/cdc.2001.980095.

35. Silva A. V. B. Estimating Fuel Consumption from GPS Data : master's thesis. 2014. URL: https://repositorio-aberto.up.pt/handle/10216/77150.

36. Routing in Internet of Vehicles: A Review / J. Cheng et al. *IEEE Transactions on Intelligent Transportation Systems*. 2015. Vol. 16, no. 5. P. 2339–2352. URL: https://doi.org/10.1109/tits.2015.2423667.

37. Perkins C. E., Bhagwat P. Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. *ACM SIGCOMM Computer Communication Review*. 1994. Vol. 24, no. 4. P. 234–244. URL: https://doi.org/10.1145/190809.190336.

38. Johnson D. B., Maltz D. A. Dynamic Source Routing in Ad Hoc Wireless Networks. *The Kluwer International Series in Engineering and Computer Science*. Boston, MA. P. 153–181. URL: https://doi.org/10.1007/978-0-585-29603-6_5.

39. Balamurugan N. M., Balamurugan S. A. A. Performance Analysis of AD-HOC on-Demand Distance Vector Routing Protocol. *Indian Journal of Science and Technology*. 2016. Vol. 9, no. 4. URL: https://doi.org/10.17485/ijst/2016/v9i4/87973.

40. Geographic routing in city scenarios / C. Lochert et al. *ACM SIGMOBILE Mobile Computing and Communications Review*. 2005. Vol. 9, no. 1. P. 69–72. URL: https://doi.org/10.1145/1055959.1055970.

41. Karp B., Kung H. T. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. Fort Belvoir, VA : Defense Technical Information Center, 2005. URL: https://doi.org/10.21236/ada440078.

42. Cyber physical surveillance system for Internet of Vehicles / D. Singh et al. *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, Singapore, 5–8 February 2018. 2018. URL: https://doi.org/10.1109/wf-iot.2018.8355218.

43. Trust-Aware Service Offloading for Video Surveillance in Edge Computing Enabled Internet of Vehicles / X. Xu et al. *IEEE Transactions on Intelligent Transportation Systems*. 2020. P. 1–10. URL: https://doi.org/10.1109/tits.2020.2995622.

44. Huang Y., Zhao M., Xue C. J. Joint WCET and Update Activity Minimization for Cyber-Physical Systems. *ACM Transactions on Embedded Computing Systems*. 2015. Vol. 14, no. 1. P. 1–21. URL: https://doi.org/10.1145/2680539.

45. Salim N. Deep Learning Approaches for Big Data Analysis. *Proceeding of the Electrical Engineering Computer Science and Informatics*. 2019. Vol. 6, no. 1. URL: https://doi.org/10.11591/eecsi.v6.2008

46. Machine learning on big data: Opportunities and challenges / L. Zhou et al. *Neurocomputing*. 2017. Vol. 237. P. 350–361. URL: https://doi.org/10.1016/j.neucom.2017.01.026;

47. Deep learning applications and challenges in big data analytics / M. M. Najafabadi et al. *Journal of Big Data*. 2015. Vol. 2, no. 1. URL: https://doi.org/10.1186/s40537-014-0007-7 (date of access: 02.05.2024).

48. Rahmaty M. Machine learning with big data to solve real-world problems. *Journal of Data Analytics*. 2023. Vol. 2, no. 1. P. 9–16. URL: https://doi.org/10.59615/jda.2.1.9.

49. Kolisetty V., Rajput D. A Review on the Significance of Machine Learning for Data Analysis in Big Data. *Jordanian Journal of Computers and Information Technology*. 2019. P. 1. URL: https://doi.org/10.5455/jjcit.71-1564729835

50. Kwan M. Gis methods in time- geographic research: geocomputation and geovisualization of human activity patterns. *Geografiska Annaler: Series B, Human Geography*. 2004. Vol. 86, no. 4. P. 267–280. URL: https://doi.org/10.1111/j.0435-3684.2004.00167.x.

51. Goodchild M. F. Geographic information systems and science: today and tomorrow. *Procedia Earth and Planetary Science*. 2009. Vol. 1, no. 1. P. 1037–1043. URL: https://doi.org/10.1016/j.proeps.2009.09.160 .

52. Tofan C. A. Optimization Techniques of Decision Making - Decision Tree. *Advances in Social Sciences Research Journal*. 2014. Vol. 1, no. 5. P. 142–148. URL: https://doi.org/10.14738/assrj.15.437.

53. Augusto O. B., Bennis F., Caro S. A new method for decision making in multi-objective optimization problems. *Pesquisa Operacional*. 2012. Vol. 32, no. 2. P. 331–369. URL: https://doi.org/10.1590/s0101-74382012005000014.

54. Chen L., Bai Q. Optimization in Decision Making in Infrastructure Asset Management: A Review. *Applied Sciences*. 2019. Vol. 9, no. 7. P. 1380. URL: https://doi.org/10.3390/app9071380.

55. Kostenko E., Kuznichenko V., Lapshyn V. Comparison of Decision-Making Methods Comparison of Decision-Making Methods. *Research in Applied Economics*. 2014. Vol. 6, no. 3. P. 17. URL: https://doi.org/10.5296/rae.v6i3.5704

56. Chen J., Ran X. Deep Learning With Edge Computing: A Review. *Proceedings of the IEEE*. 2019. Vol. 107, no. 8. P. 1655–1674. URL: https://doi.org/10.1109/jproc.2019.2921977.

57. An Overview on Edge Computing Research / K. Cao et al. *IEEE Access*. 2020. Vol. 8. P. 85714–85728. URL: https://doi.org/10.1109/access.2020.2991734;

58. A review of optimization methods for computation offloading in edge computing networks / K. Sadatdiynov et al. *Digital Communications and Networks*. 2022. URL: https://doi.org/10.1016/j.dcan.2022.03.003

59. Методи інтелектуальної обробки зображень та відео на основі метрик візуальної якості для перспективних застосувань / С. Абрамов та ін. *European Science*. 2023. Sge20-02. С. 51–98. URL: https://doi.org/10.30890/2709-2313.2023-20-02-010

60. Щербінін К. С. Методи та моделі сегментації відео для систем метричного пошуку : автореф. дис. … канд. техн. наук. Харків, 2011. 20 с.

61. Uploader Intent for Online Video: Typology, Inference, and Applications / C. Kofler et al. *IEEE Transactions on Multimedia*. 2015. Vol. 17, no. 8. P. 1200–1212. URL: https://doi.org/10.1109/tmm.2015.2445573.

62. Viewpoints and Views in Hardware Platform Modeling for Safe Deployment / U. Pohlmann et al. *the 2nd Workshop*, York, United Kingdom, 22 July 2014. New York, New York, USA, 2014. URL: https://doi.org/10.1145/2631675.2631682.

63. Allan R. Automotive Networks Strive To Satisfy Safety And Bandwidth Needs. *Home | Electronic Design*. URL: https://www.electronicdesign.com/technologies/communications/article/21754006/automotive-networks-strive-to-satisfy-safety-and-bandwidth-needs (date of access: 11.04.2024).

64. Trust-Aware Service Offloading for Video Surveillance in Edge Computing Enabled Internet of Vehicles / X. Xu et al. *IEEE Transactions on Intelligent Transportation Systems*. 2020. P. 1–10. URL: https://doi.org/10.1109/tits.2020.2995622 (date of access: 22.04.2024).

65. Avnet Abacus. A*vnet EMEA: Quality Electronic Components & Services*. URL: https://my.avnet.com/abacus/solutions/markets/automotive-and-transportation/automotive/communications-and-connectivity/v2v-communication (date of access: 03.04.2024).

66. Станіслав Ю. Складний шлях до 5G в Україні. Чому його варто розпочати з питання кібербезпеки? Розповідає дослідник Станіслав Юхименко – Forbes.ua. Forbes.ua | *Бізнес, мільярдери, новини, фінанси, інвестиції, компанії.* URL: https://forbes.ua/innovations/skladniy-shlyakh-do-5g-v-ukraini-chomu-yogo-

varto-rozpochati-z-pitannya-kiberbezpeki-rozpovidae-doslidnik-stanislav-yukhimenko-19032024-19979 (дата звернення: 03.04.2024).

67. Padilla R., Netto S. L., da Silva E. A. B. A Survey on Performance Metrics for Object-Detection Algorithms. *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, Niterói, Brazil, 1–3 July 2020. 2020. URL: https://doi.org/10.1109/iwssip48289.2020.9145130

68. Du L., Zhang R., Wang X. Overview of two-stage object detection algorithms. *Journal of Physics: Conference Series*. 2020. Vol. 1544. P. 012033. URL: https://doi.org/10.1088/1742-6596/1544/1/012033 (date of access: 03.05.2024).

69. Mahaur B., Singh N., Mishra K. K. Road object detection: a comparative study of deep learning-based algorithms. *Multimedia Tools and Applications*. 2022. URL: https://doi.org/10.1007/s11042-022-12447-5

70. A Review of Yolo Algorithm Developments / P. Jiang et al. *Procedia Computer Science*. 2022. Vol. 199. P. 1066–1073. URL: https://doi.org/10.1016/j.procs.2022.01.135.

71. Hussain M. YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. *Machines*. 2023. Vol. 11, no. 7. P. 677. URL: https://doi.org/10.3390/machines11070677.

72. Accurate and fast single shot multibox detector / L. Guo et al. *IET Computer Vision*. 2020. Vol. 14, no. 6. P. 391–398. URL: https://doi.org/10.1049/iet-cvi.2019.0711 (date of access: 03.05.2024).

73. 073 Kumar A., Srivastava S. Object Detection System Based on Convolution Neural Networks Using Single Shot Multi-Box Detector. *Procedia Computer Science*. 2020. Vol. 171. P. 2610–2617. URL: https://doi.org/10.1016/j.procs.2020.04.283 (date of access: 03.04.2024).

74. You Only Look Once: Unified, Real-Time Object Detection / J. Redmon et al. 2016. URL: https://doi.org/10.48550/arXiv.1506.02640.(date of access: 03.04.2024)

75. SSD: Single Shot MultiBox Detector / W. Liu et al. *Springer International Publishing*. 2016. URL: https://doi.org/10.1007/978-3-319-46448-0_2.

76. Prangishvili A., Namicheishvili O., Ramazashvili M. Convolutional Neural Networks. *Works of Georgian Technical University*. 2020. No. 3(517). P. 33–56. URL: https://doi.org/10.36073/1512-0996-2020-3-33-56.

77. Support vector machines / M. A. Hearst et al. *IEEE Intelligent Systems and their Applications*. 1998. Vol. 13, no. 4. P. 18–28. URL: https://doi.org/10.1109/5254.708428 (date of access: 03.05.2024).

78. Wang Y., Wang Z.-O. A Fast KNN Algorithm for Text Categorization. *2007 International Conference on Machine Learning and Cybernetics*, Hong Kong, China, 19–22 August 2007. 2007. URL: https://doi.org/10.1109/icmlc.2007.4370742.

79. Comparing LBP, HOG and Deep Features for Classification of Histopathology Images / T. J. Alhindi et al. *2018 International Joint Conference on Neural Networks (IJCNN)*, Rio de Janeiro, 8–13 July 2018. 2018. URL: https://doi.org/10.1109/ijcnn.2018.8489329.

80. Wang Z. A new clustering method based on morphological operations. *Expert Systems with Applications*. 2020. Vol. 145. P. 113102. URL: https://doi.org/10.1016/j.eswa.2019.113102

81. Tran L. A. Centroid Neural Network: An Efficient and Stable Clustering Algorithm. *Medium*. URL: https://pub.towardsai.net/centroid-neural-network-an-efficient-and-stable-clustering-algorithm-b2fa8cbb2a27 (date of access: 16.03.2024).

82. Clustering Algorithms | Machine Learning | Google for Developers. *Google for Developers*. URL: https://developers.google.com/machine-learning/clustering/clustering-algorithms (date of access: 17.03.2024).

83. Ціни на бензин, дизпаливо, газ на АЗС України (березень 2024). *Ставки, індекси, тарифи*. URL: https://index.minfin.com.ua/ua/markets/fuel/2024-03/ (дата звернення: 27.04.2024).

84. GPS модуль NEO-6M v2. *МікроАмпер.* URL: *https://uamper.com/index.php?route=product/product&path=179&product_id=46&gad_source=1&gclid=Cj0KCQjwxeyxBhC7ARIsAC7dS3_-j89EWXZXRN-Tymp8nHEUujUR1AOEdw0Usn0wHex2oIZwmJlTgOYaAht_EALw_wcB*

ДОДАТОК А
(обов’язковий)

КОПІЇ ПУБЛІКАЦІЇ

1) М. С. Бойко, В. В. Яцків. Кіберфізична система спостереження як частина інтернету транспортних засобів. Міжнародна студентська наукова конференція «Тренди та перспективи розвитку мультидисцип-лінарних досліджень»: Одеса, Україна, 2024. С. 159-161.

2) M. Boiko, V. Yatskiv. Surveillance cyber-physical system as a part of internet of vehicles. Computer systems and information technologies. ISSN 2710-0766. 2024. Vol. 2.

Бойко Микита Сергійович, здобувач вищого ступеня
факультету інформаційних технологій
Хмельницький національний університет, м. Хмельницький, Україна

Науковий керівник: Яцків Василь Васильович, д-р. техн. наук,
професор, завідувач кафедри кібербезпеки
Західноукраїнський національний університет, м. Тернопіль, Україна

КІБЕРФІЗИЧНА СИСТЕМА СПОСТЕРЕЖЕННЯ ЯК ЧАСТИНА ІНТЕРНЕТУ ТРАНСПОРТНИХ ЗАСОБІВ

В сучасному світі розробка інноваційних технологій, спрямованих на оптимізацію та удосконалення різноманітних аспектів взаємодії людини з машинами, набуває все більшого значення. Однією з перспективних сфер в цьому контексті є інтернет транспортних засобів (IoV), який відіграє ключову роль у сучасній інтернет-екосистемі. IoV – це підгалузь інтернету речей (Internet of Things - IoT), яка передбачає інтеграцію розумних пристроїв з транспортними засобами, що створює потужну інформаційну базу, доступну для аналізу через вбудовані датчики або відеореєстратори.

Україна не відстає у розвитку інтернету транспортних засобів, зокрема вже існують системи, що базуються на цій концепції. Однак вони мають обмежений функціонал та базуються переважно на геолокаційних даних.

Мета нашої роботи полягає в моделюванні та дослідженні методів для реалізації кіберфізичної системи спостереження(КФСС) цілями якої є моніторинг цін на автозаправних станціях.

В процесі дослідження було здійснено аналіз існуючих наукових робіт в сфері IoV, а саме: способи впровадження кіберфізичних систем в широкому спектрі галузей [1], протоколи маршрутизації [2], концепції безпечного IoV [3] та методи вивантаження завдань відеоспостереження в розподіленому обчислювальному середовищі [4].

Згідно поставленої мети було визначено абстрактну модель КФСС та розділено її на три типи підсистем: вузли, клієнти та сервери. Для кожної з цих підсистем було розроблено математичну модель, яка описує складові підсистеми. Загальну модель системи можна описати у вигляді наступної множини:

$$CPS = \{hss_1, \dots hss_m, fss_{m+1}, fss_{m+2}, bss_{m+3} \dots, bss_{m+9}\} \quad (1)$$

де:
hss_i – це підсистеми периферійних вузлів.
$fsso_j$ – це клієнтські підсистеми;
bss – це серверні підсистеми.
m – кількість периферійних вузлів.

Проаналізувавши технологічні можливості інфраструктури в Україні була змодельована архітектура мережі КФСС (рис. 1), на якій зображено LTE як основний спосіб передачі інформації від периферійних вузлів системи (транспортних засобів) до інших підсистем КФСС.

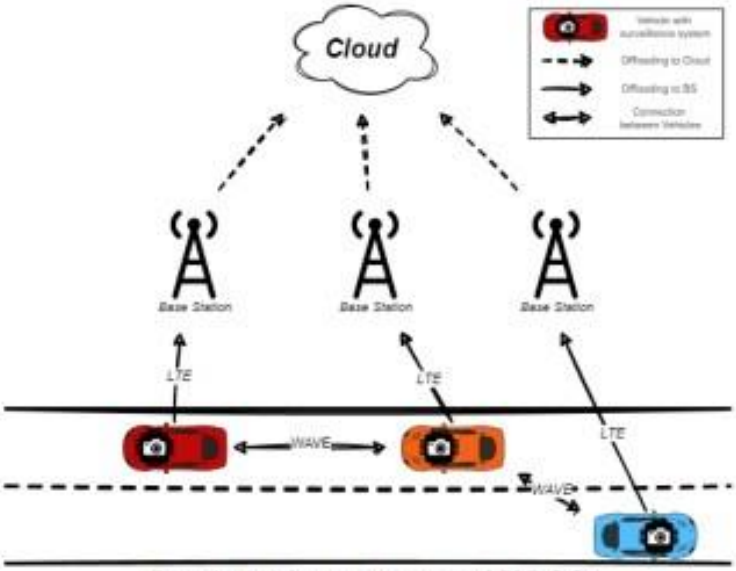


Рис. 1. **Архітектура мережі КФСС**

Згідно розробленої архітектури (рис. 1) мережі була змодельована часова модель КФСС яку можна описати наступним чином:

$$T_{total} = \sum_{j=1}^{J}(h_j + o_j) \qquad (1)$$

де:
h_i – це час обчислення в одному периферійному вузлі;
o_j – це час вивантаження завдання по обробці відео або зображення;
J – це кількість виконуваних завдань.

Під час розробки КФСС було порівняно два одноступеневі алгоритми виявлення об'єктів на зображенні: YOLO (You Only Look Once) та SSD (Single Shot MultiBox Detector). Та було визначено, що метод YOLO швидший за SSD, оскільки може обробити більше кадрів за секунду. Середня швидкість YOLO може варіюватись від 40 до 155 кадрів на секунду, тоді як SSD – від 22 до 46 кадрів на секунду. Проте під час дослідження метод YOLO показав гірші результати у виявленні табло з цінами АЗС, що призводило до аномалій при обиранні сторонніх об'єктів для аналізу тексту.

Для оптичного розпізнавання символів було проаналізовано два сервіси: Google Cloud Vision API та Tesseract. Google Cloud Vision API виділявся своєю високою точністю та підтримкою великої кількісті мов, але відштовхував своєю комерційністю, а саме – вартістю використання. У той самий час, Tesseract – це безкоштовне відкрите програмне забезпечення з високою гнучкістю та можливістю налаштування, проте вимагає додаткового навчання для досягнення оптимальних результатів. В результаті для розробки КФСС було обрано Google Cloud Vision API оскільки наша система вимагає високої точності та надійності у розпізнаванні тексту.

Для визначення геолокаційного позиціонування периферійних вузлів було

обрано глобальну систему позиціонування(GPS), оскільки це надійний та точний метод, який є широко використовуваним стандартом для визначення місцезнаходження, а особливо коли периферійні вузли нашої КФСС – це апаратний засіб який встановлюється в транспортний засіб.

За результатами досліджень роботи підсистем КФСС можна зробити висновок, що згідно проведених експериментів дана система показує себе працездатною та має перспективи на розвиток в сфері IoV. Проте варто відзначити, що система показала аномалії у виявленні об'єктів та оптичному розпізнаванні символів, що стимулювало розробку нових стратегій обробки та аналізу даних, з урахуванням можливих помилок нейронних мереж.

Реалізація нашої КФСС демонструє нове вікно можливостей для систем моніторингу та спостереження в цілій галузі інтернету транспортних засобів та масі інших.

Список використаних джерел:

1. Muzaffar, H.; Md. Monirul, I.; Shahriar, H.; Md. Nasim A.; Most. Jannatul F.; Muhammed B. J.; Ali W. M. Distributed Control of Cyber Physical System on Various Domains: A Critical Review, 2023. p. 7. [MDPI].
2. Cheng, J., Cheng, J., Zhou, M., Liu, F., Gao, S., & Liu, C. (2015). Routing in Internet of Vehicles: A Review. IEEE Transactions on Intelligent Transportation Systems, 16(5), 2339–2352. doi:10.1109/tits.2015.2423667.
3. Singh, D., Tripathi, G., Shah, S. C., & da Rosa Righi, R. (2018). Cyber physical surveillance system for Internet of Vehicles. 2018 IEEE 4th World Forum on Internet of Things (WF-IoT). doi:10.1109/wf-iot.2018.8355218.
4. Xu, X., Wu, Q., Qi, L., Dou, W., Tsai, S.-B., & Bhuiyan, M. Z. A. (2020). Trust-Aware Service Offloading for Video Surveillance in Edge Computing Enabled Internet of Vehicles. IEEE Transactions on Intelligent Transportation Systems, 1–10. doi:10.1109/tits.2020.2995622.

UDC 004.72

MYKYTA BOIKO, VASYL YATSKIV
Khmelnytskyi National University, Khmelnytskyi, Ukraine

SURVEILLANCE CYBER-PHYSICAL SYSTEM AS A PART OF INTERNET OF VEHICLES

Integration of cyber-physical surveillance systems (CPSS) into the Internet of Vehicles (IoV) paradigm represents a transformative approach to enhancing transportation safety and efficiency. This article discusses the design, implementation, and application of CPSS as part of IoV ecosystems. Leveraging advancements in sensor technologies, communication protocols, and data analytics, CPSS within IoV enables real-time monitoring, analysis, and response to road conditions, incidents, and emergencies. Our research explores the architecture and functional capabilities of CPSS, including sensor deployment, data fusion, anomaly detection, and decision support mechanisms. We investigate the synergistic interaction between CPSS and IoV platforms, facilitating seamless data exchange, collaboration, and compatibility between automotive and infrastructural domains. Additionally, we discuss potential applications of CPSS in traffic management, law enforcement, emergency response, and urban planning, emphasizing its role in enhancing transportation safety, optimizing resource allocation, and preventing congestion and accidents. Through empirical evaluations and thematic studies, we demonstrate the effectiveness, scalability, and societal impact of integrating CPSS into IoV ecosystems. This research contributes to the development of intelligent transportation systems and underscores the transformative potential of CPSS within the IoV context.

Keywords: cyber-physical system, Internet of Vehicles, intelligent transportation systems, edge computing.

М.С. БОЙКО, В.В. ЯЦКІВ
Хмельницький національний університет

СИСТЕМА КІБЕР-ФІЗИЧНОГО СПОСТЕРЕЖЕННЯ ЯК ЧАСТИНА ІНТЕРНЕТУ ТРАНСПОРТНИХ ЗАСОБІВ

Інтеграція кіберфізичних систем спостереження (КФСС) у парадигму Інтернету транспортних засобів (IoV) представляє трансформаційний підхід до підвищення безпеки та ефективності транспортування. У цій статті розглядається дизайн, впровадження та застосування КФСС як частину екосистем IoV. Використовуючи досягнення в сенсорних технологіях, протоколах зв'язку та аналітиці даних, КФСС в рамках IoV дозволяє здійснювати моніторинг, аналіз і реагування на дорожні умови, інциденти та надзвичайні ситуації в реальному часі. Наше дослідження вивчає архітектуру та функціональні можливості КФСС, включаючи розгортання датчиків, об'єднання даних, виявлення аномалій та механізми підтримки прийняття рішень. Ми досліджуємо синергетичну взаємодію між платформами КФСС та IoV, сприяючи безперебійному обміну даними, співпраці та сумісності між автомобільними та інфраструктурними доменами. Крім того, ми обговорюємо потенційні застосування КФСС в управлінні дорожнім рухом, правоохоронній діяльності, реагуванні на надзвичайні ситуації та міському плануванні, підкреслюючи його роль у підвищенні безпеки транспортування, оптимізації розподілу ресурсів і запобіганню заторів і аварій.

За допомогою емпіричних оцінок і тематичних досліджень ми демонструємо ефективність, масштабованість і вплив на суспільство інтеграції КФСС в екосистеми IoV. Це дослідження робить внесок у розвиток інтелектуальних транспортних систем і підкреслює трансформаційний потенціал КФСС в контексті IoV.

Ключові слова: кіберфізична система, інтернет транспортних засобів, інтелектуальні транспортні системи, периферійні обчислення.

Introduction

Internet of Vehicles (IoV) is a system of loosely connected heterogeneous or homogeneous devices capable of receiving, transmitting, and processing information among each other. Mostly discussed alongside the use of Intelligent Transportation Systems (ITS) to create safe and reliable automated vehicle control technology [1]. This technology is equipped with embedded sensors based on automotive devices while simultaneously connected to environments, people, and systems [2]. The development of new vehicles is already being done with consideration for IoV and ITS technologies, which are integrated at the design and assembly stages of the vehicle. The future will be defined by the interactions between entities, sensors, smartphones, and vehicles, which are not exceptions in this context. Thus, considering potential methods of creating cyber-physical systems in the Internet of Things (IoT) network becomes crucial.

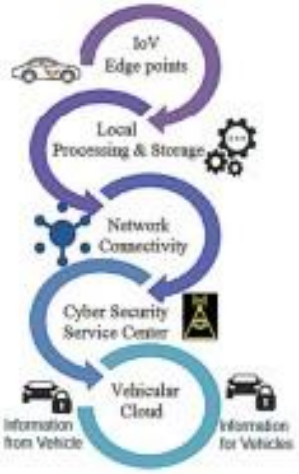


Fig.1. Concept of a secure Internet of vehicles [1]

Figure 1 illustrates the general conceptual scheme of CPSS in the IoV network, which can ensure driver safety through a cyber security service center and utilize its monitoring devices to gather useful information and disseminate it within or even beyond the IoV. Computers play a significant role in this system, as many smart vehicles have become intelligent due to advancements in technology and computational power of devices. Currently, smart cars have added stability control, optimal mileage, fuel injection, navigation, and theft prevention [3]. However, there is almost no discussion or development in the field of individual portable cyber-physical systems that can be involved in IoV usage, not only for safe vehicle control but also for collecting and disseminating useful information such as road traffic accidents, road conditions, or monitoring fuel prices at gas stations. Similar to the technological use of smartphones, a portable cyber-physical system (CPSS) for vehicles can add a lot of functionality based on the data it can accumulate and analyze.

In this article, we present an approach to designing the proposed CPSS model as part of IoV. We will consider available technical and system solutions for creating such a system. We will develop the network architecture and build a time cost model of the system. In the final section, we will discuss possible positioning of CPSS in the vehicle and the components it should consist of, describing them as a mathematical model.

Related works

In the Internet of Things (IoV), computational capabilities and storage of video surveillance fall under edge computing. Although the capacity of surveillance systems is strengthened by edge computing, there are still some challenges that arise, namely: time costs, task offloading, and confidentiality preservation in edge computing are being investigated according to recent research.

Task offloading aims at proper distribution of computational tasks and their efficient allocation to neighboring computing devices. From the perspective of task offloading, an effective real-time video system with an efficient resource provisioning strategy and low latency video cluster scheduler has been developed in [4]. The architecture of collaborative video processing to balance the limited network bandwidth and task offloading has been investigated in [5]. Both approaches effectively address the issue of limited resources. However, it is also important to consider the efficiency of task offloading.

In [6], the author proposed an algorithm using hypergraph segmentation to minimize network load of systems with an optimal way of considering the non-obligatory case. Taking into account the offloading capacity, a new network offloading scheme has been developed in [7] to enhance the power of edge computing by maximizing constant

productivity while ensuring energy expenditure with constant constraints. A task offloading model has been proposed in [8] to enhance long-term utility based on a powerful distributed offloading model during various wireless connections.

Additionally, in terms of time costs, a distributed smart surveillance system has been introduced in [9], which offloads computational burdens to alleviate high communication overheads and provides low-latency video analysis solutions using deep learning algorithms. A strategy of collective video processing has been proposed in [10] to achieve multimedia tasks with time-sensitive reactions and improve human detection accuracy through edge computing.

However, none of these approaches take security aspects into account. Peripheral nodes in video surveillance face challenges in confidentiality preservation. A paradigm supporting secure communication and preserving confidentiality has been presented in [11]. A new protocol has been developed in [12] to provide desired security features. Strategies for optimizing task offloading and data confidentiality protection are presented in [13] and [14].

However, current research on task offloading in edge computing mainly focuses on the efficiency of implementing computational tasks, neglecting the drawbacks of unbalanced service placement. In this article, we will consider the placement of the peripheral device system by moving them from the surrounding environment to vehicles.

System Model

In this section, the network architecture and time cost model for our CPS are developed and presented. The main system notations are provided in Table 1.

Table 1

Notations and Definitions

Notations	Definitions
N	Number of vehicles with a cyber-physical system
S	Set of peripheral systems, where $S = \{s1, s2, \ldots sN\}$
L	Number of virtual machines in S_N
J	Number of executed tasks
V_N	Set of virtual machines in S_N, where $V_N = \{v_{n,1}, v_{n,2}, \ldots v_{n,L}\}$
T	Set of executed tasks, where $T = \{t1, t2, \ldots tJ\}$
T_{total}	Total time spent on tasks

In this article, we utilize edge computing technology in the video surveillance system. As shown in Figure 2, an architecture of video surveillance with edge computing is proposed, where observation terminals are installed in vehicles, and generated video data is offloaded directly to the base station using LTE communication. Peripheral nodes with powerful computational capabilities (vehicles with CPS) take on computational tasks. Then, the computation results are sent from peripheral nodes to the cloud environment via base stations. In this scenario, along the road, there are N vehicles, each equipped with CPS for video surveillance, denoted as $S = \{s1, s2, \ldots sN\}$. The CPS device set consists of two components: peripheral nodes and access points. Let $T = \{t1, t2, \ldots tJ\}$ denote a set of J computational tasks generated from nodes. Peripheral nodes have powerful computational capabilities and preprocessing capacity, consisting of virtual machines (VM). There are L virtual machines in sw, defined as $VN = \{vn, 1, vn, 2, \ldots vn, L\}$.

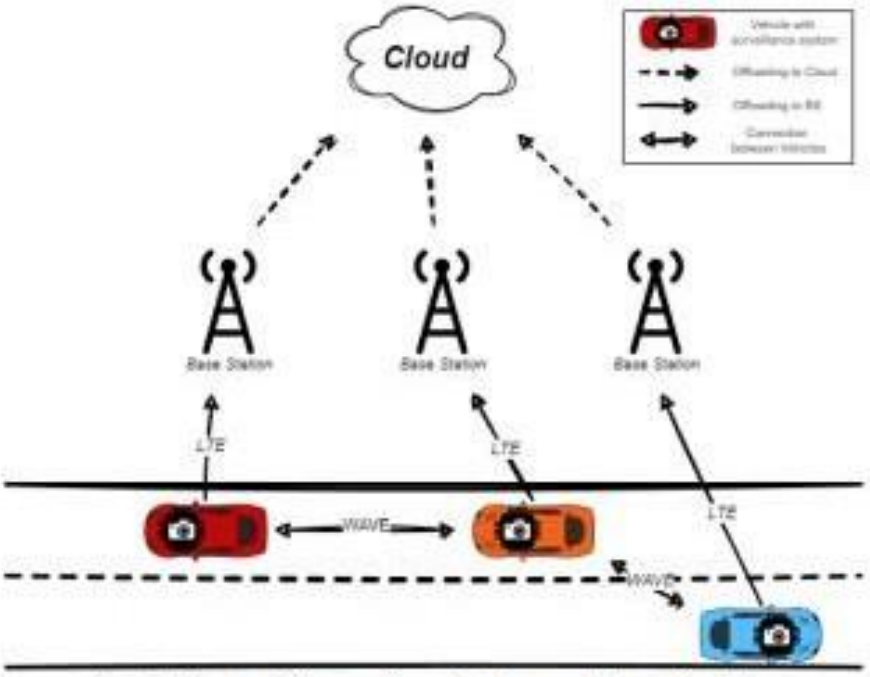


Fig. 2. Architecture of video surveillance with edge computing located in vehicles

The next time cost model will be developed based on the work type [15], in which the authors also developed a time cost model, load balance model, and entropy model for the offloading system based on static objects for monitoring vehicle traffic.

Time costs consist of computation time at the respective peripheral node and offloading time. Time costs are an important parameter that determines the quality of real-time service on monitoring devices. When video tasks are sent to the image processing device, the CPS should provide virtual machines for computation. The computational capacity of peripheral nodes is associated with the number of available virtual machines. Thus, the number of available virtual machines in s_n is denoted as x_n, and the operational capacity of each virtual machine is denoted as φ. Suppose the video task with duration F_j is processed in s_n, and the computation time of s_n is expressed as (1):

$$h_j = \sum_{n=1}^{N} k_{j,n} \frac{F_j}{x_n \cdot \varphi} \tag{1}$$

, where $k_{j,n} = 1$ if the data t_j arrived at s_n, otherwise 0.

Once the video processing in s_n is completed, the computed result is offloaded from the peripheral devices to the cloud center. The offloading time of the computation result t_j with weight $f_{j,output}$ can be calculated as (2):

$$o_j = \sum_{n=1}^{N} k_{j,n} \left(\frac{f_{j,output}}{\eta} + \frac{f_{j,output}}{\lambda} \right) \tag{2}$$

where η represents the transmission speed between peripheral nodes and base stations, and λ denotes the transmission speed between base stations and the cloud data processing center.

With the calculated two components, the total time costs for task execution can be determined by the following formula (3):

$$T_{total} = \sum_{j=1}^{J} (h_j + o_j) \tag{3}$$

Structure of CPSS

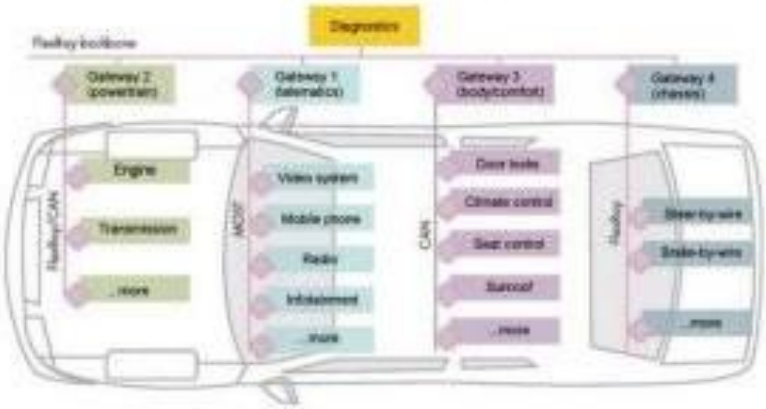


Fig. 3. Topology of the automotive network [16]

Before developing the CPS, it is necessary to determine which part of the automotive network our system belongs to. According to [17], the automobile topology can be divided into four parts, each with its own diagnostic gateways: telematics, powertrain, body/comfort, and chassis. In this topology, telematics is best suited for integrating our CPS, as it combines telecommunications, automotive technologies, electrical engineering, and informatics of the vehicle, and it utilizes the Media Oriented Systems Transport (MOST) communication protocol.

Let's define the subsystems of this system and describe them in the form of a mathematical model:

1. Communication module (S). At the initial stages, a smartphone can be used as an access point providing communication between the system and a remote server via the Internet.

Parameters: communication state (S_{state}), data transmission speed (S_{speed}), network access time (Saccess_time).

2. GPS module (G). Determines the location of our system (vehicle) and provides coordinates that can be used for various purposes such as navigation or location determination.

Parameters: Coordinates (x, y), location determination accuracy ($G_{accuracy}$), update time (G_{update_time}).

3. Surveillance camera (C). Designed to gather information about events on the road.

Parameters: Resolution ($C_{resolution}$), frame rate (C_{frame_rate}), processing delay time ($C_{processing_time}$).

4. Power source (P). Provides power supply for the entire peripheral system.

Parameters: Voltage ($P_{voltage}$), Power (P_{power}).

5. Single-board computer (R). Responsible for image processing, interaction with the GPS module, and data transmission to the remote server.

Parameters: Memory size (R_{memory}), processor speed ($R_{processor_speed}$), load level (R_{load}).

Thus, the our CPSS can be described as the following set (4):

$$CPSS = \{S, G, C, P, R\} \tag{4}$$

The mentioned subsystems represent the key components of our CPS in IoV for surveillance, each responsible for specific functionality and having its own parameters that determine their efficiency and productivity. Developing a mathematical model of the system has allowed for a better understanding of its operation and component interaction, which is crucial for the successful implementation and optimization of the system's functioning in real operating conditions.

Conclusion

This article explores the potential of Cyber-Physical Systems (CPS) in the realm of the Internet of Vehicles (IoV), particularly within the context of surveillance systems in the IoV network. It proposes an approach to designing CPS, examining existing technical and systemic solutions for their creation. The article delves into the network architecture, the system's time cost model, and the possible positioning of CPS within vehicles.

The aim of this article is to investigate the possibilities of utilizing CPS in the IoV sphere and to initiate a discussion on their implementation and potential benefits for the development of transportation infrastructure and road safety. It opens up new perspectives for improving transportation systems and creating effective monitoring and control mechanisms, thereby promoting safer and more efficient transportation usage.

References

1. Singh, D., Tripathi, G., Shah, S. C., & da Rosa Righi, R. (2018). Cyber physical surveillance system for Internet of Vehicles. 2018 IEEE 4th World Forum on Internet of Things (WF-IoT). doi:10.1109/wf-iot.2018.8355218

2. F. Yang, S. Wang, J. Li, Z. Liu and Q. Sun, "An overview of Internet of Vehicles," in China Communications, vol. 11, no. 10, pp. 1-15, Oct. 2014.

3. D. George, P. Demestichas. "Intelligent transportation systems." IEEE Vehicular Technology Magazine 5.1 (2010): 77-84

4. P.-H. Wu, C.-W. Huang, J.-N. Hwang, J.-Y. Pyun, and J. Zhang, "Video-quality-driven resource allocation for real-time surveillance video uplinking over OFDMA-based wireless networks," IEEE Trans. Veh. Technol., vol. 64, no. 7, pp. 3233–3246, Jul. 2015.

5. J. Wang, L. Zhao, J. Liu, and N. Kato, "Smart resource allocation for mobile edge computing: A deep reinforcement learning approach," IEEE Trans. Emerg. Topics Comput., early access, Mar. 4, 2019, doi. 10.1109/TETC.2019.2902661.

6. N. Tziritas et al., "Data replication and virtual machine migrations to mitigate network overhead in edge computing systems," IEEE Trans. Sustain. Comput., vol. 2, no. 4, pp. 320–332, Oct. 2017.

7. L. Chen, S. Zhou, and J. Xu, "Computation peer offloading for energyconstrained mobile edge computing in small-cell networks," IEEE/ACM Trans. Netw., vol. 26, no. 4, pp. 1619–1632, Aug. 2018.

8. T. Quang Dinh, Q. Duy La, T. Q. S. Quek, and H. Shin, "Learning for computation offloading in mobile edge computing," IEEE Trans. Commun., vol. 66, no. 12, pp. 6353–6367, Dec. 2018.

9. J. Chen, K. Li, Q. Deng, K. Li, and P. S. Yu, "Distributed deep learning model for intelligent video surveillance systems with edge computing," IEEE Trans. Ind. Informat., early access, Apr. 9, 2019, doi: 10.1109/TII.2019.2909473.

10. L. T. Tan and R. Q. Hu, "Mobility-aware edge caching and computing in vehicle networks: A deep reinforcement learning," IEEE Trans. Veh. Technol., vol. 67, no. 11, pp. 10190–10203, May 2018.

11. J. Kang, R. Yu, X. Huang, and Y. Zhang, "Privacy-preserved pseudonym scheme for fog computing supported Internet of vehicles," IEEE Trans. Intell. Transp. Syst., vol. 19, no. 8, pp. 2627–2637, Aug. 2018.

12. U. L. N. Puvvadi, K. D. Benedetto, A. Patil, K.-D. Kang, and Y. Park, "Cost-effective security support in real-time video surveillance," IEEE Trans. Ind. Informat., vol. 11, no. 6, pp. 1457–1465, Dec. 2015.

13. M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, "Distributed resource allocation in blockchain-based video streaming systems with mobile edge computing," IEEE Trans. Wireless Commun., vol. 18, no. 1, pp. 695–708, Jan. 2019.

14. X. Li, S. Liu, F. Wu, S. Kumari, and J. J. P. C. Rodrigues, "Privacy preserving data aggregation scheme for mobile edge computing assisted IoT applications," IEEE Internet Things J., vol. 6, no. 3, pp. 4755–4763, Jun. 2019.

15. Xu, X., Wu, Q., Qi, L., Dou, W., Tsai, S.-B., & Bhuiyan, M. Z. A. (2020). Trust-Aware Service Offloading for Video Surveillance in Edge Computing Enabled Internet of Vehicles. IEEE Transactions on Intelligent Transportation Systems, 1–10. doi:10.1109/tits.2020.2995622

16. R. ALLAN. Automotive networks strive to satisfy safety and bandwidth needs. illustration. Electronic Design, 57(21):28 – 33, 2009.

17. Pohlmann, U., Meyer, M., Dann, A., & Brink, C. (2014). Viewpoints and Views in Hardware Platform Modeling for Safe Deployment. Proceedings of the 2nd Workshop on View-Based, Aspect-Oriented and Orthographic Software Modelling - VAO '14. doi:10.1145/2631675.

Mykyta Boiko – bachelor, Khmelnytskyi National University, Khmelnytskyi, Ukraine,

e-mail: nikita2000223@gmail.com

Vasyl Yatskiv – Doctor of Technical Science, Professor, Western Ukrainian National University, Ternopil, Ukraine.

e-mail: jazkiv@ukr.net

https://orcid.org/0000-0001-9778-6625, Scopus Author ID: 27468042400,

https://scholar.google.com.ua/citations?hl=uk&user=o561posAAAAJ

Бойко Микита Сергійович – бакалавр, Хмельницький національний університет, Хмельницький, Україна.

Яцків Василь Васильович – доктор технічних наук, професор, завідувач кафедри кібербезпеки, Західноукраїнський національний університет, Тернопіль, Україна.

ДОДАТОК Б

(обов'язковий)

ЛІСТИНГ КОДУ

Периферійний вузол:

```
import cv2
import time
import uuid
import serial
import requests
import os
from ultralytics import YOLO
from queue  import  Queue
from threading import Thread

API_KEY = "API_KEY"

def read_gps_data(serial_port):
    gps_data = {}
    while True:
        line = serial_port.readline().decode().strip()
        if line.startswith('$GPGGA'):
            parts = line.split(',')
            if len(parts) >= 10:
                latitude = float(parts[2][:2]) + float(parts[2][2:]) / 60
                longitude = float(parts[4][:3]) + float(parts[4][3:]) / 60
                gps_data['latitude'] = latitude
                gps_data['longitude'] = longitude
                break
    return gps_data

def calculate_sharpness(image):
    # Convert the image to grayscale
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    # Calculate the Laplacian of the image
    laplacian = cv2.Laplacian(gray, cv2.CV_64F)
    # Calculate the variance of the Laplacian to measure sharpness
    sharpness = laplacian.var()
    return sharpness

def plot_bboxes(results, frame):
    xyxys = []
    confidences = []
    class_ids = []
    for result in results:
        boxes = result.boxes.cpu().numpy()
        xyxys.append(boxes.xyxy)
        confidences.append(boxes.conf)
        class_ids.append(boxes.cls)
        anotated_frame = results[0].plot()
    return anotated_frame, xyxys, confidences, class_ids

def crop_image(frame, xyxy):
```

```
    return frame[int(xyxy[1]):int(xyxy[3]), int(xyxy[0]):int(xyxy[2])]

def send_gps_data_to_server(payload, headers):
    headers['Content-Type'] = 'application/json'
    response = requests.post('http://localhost:8080/api/gas-station/location',
json=payload, headers=headers)
    if response.status_code != 200:
        print("Failed to send GPS data:", response.status_code)
        return False
    else:
        print("GPS data sent successfully")
        return True

def send_image_data_to_server(payload, headers):
    headers['Content-Type'] = 'image/jpeg'
    response = requests.post('http://localhost:8080/api/gas-station/board-
image', data=payload, headers=headers)
    if response.status_code != 200:
        print("Failed to send image data:", response.status_code)
        return False
    else:
        print("Image data sent successfully")
        return True

def send_data_worker():
    while True:
        gps_payload, image_payload, headers = data_queue.get()
        gps_sent = send_gps_data_to_server(gps_payload, headers)
        image_sent = send_image_data_to_server(image_payload, headers)
        if not (gps_sent and image_sent):
            print("Failed to send data, will retry later...")
            data_queue.put((gps_payload, image_payload, headers))
            time.sleep(60)  # Retry after 1 minute
        data_queue.task_done()

def main():
    serial_port = serial.Serial('/dev/ttyS0', 9600, timeout=1)
    model = YOLO('best_float32.tflite')
    cap = cv2.VideoCapture(0)

    while cap.isOpened():
        success, frame = cap.read()

        if success:
            start = time.perf_counter()
            results = model.predict(frame)
            end = time.perf_counter()
            total_time = end - start
            fps = 1 / total_time

            anotated_frame, xyxys, confidences, class_ids =
plot_bboxes(results, frame)
            cv2.putText(anotated_frame, f"FPS: {int(fps)}", (25, 25), 0, 1,
(255, 0, 0))
            cv2.imshow("YOLOv8", anotated_frame)

            if confidences[0] > 0.4:
                gps_data = read_gps_data(serial_port)
                ts = time.time()
                img_id = uuid.uuid4().hex
```

```
                headers = {
                    'User-Agent': 'Raspberry Pi Edge Device #1',
                    'API-KEY': API_KEY,
                    'Image-Id': img_id,
                    'Content-Type': None
                }
                gps_payload = {
                    'coordinates': gps_data,
                    'timestamp': ts
                }

                # Crop the detected object using the bounding box coordinates
                best_crop = None
                best_sharpness = -1

                for xyxy in xyxys[0]:

                    crop_obj = frame[int(xyxy[1]):int(xyxy[3]),
int(xyxy[0]):int(xyxy[2])]
                    sharpness = calculate_sharpness(crop_obj)

                    if sharpness > best_sharpness:
                        best_crop = crop_obj
                        best_sharpness = sharpness

                data_queue.put((gps_payload, best_crop, headers))

            if cv2.waitKey(1) & 0xFF == ord("q"):
                break
        else:
            break

    cap.release()
    cv2.destroyAllWindows()

if_name_== "_main_":
    data_queue = Queue()
    send_data_thread = Thread(target=send_data_worker, daemon=True)
    send_data_thread.start()
    main()
```

Кластеризація DBSCAN 2D:

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import DBSCAN
from sklearn.preprocessing import StandardScaler

# Read data from CSV file
data = pd.read_csv('gas_data.csv', delimiter=';', decimal=',')

# Select features for clustering
X = data[['latitude', 'longitude']]

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

```
# Apply DBSCAN clustering
eps = 0.15  # epsilon parameter for DBSCAN
min_samples = 15 # min_samples parameter for DBSCAN
dbscan = DBSCAN(eps=eps, min_samples=min_samples)
clusters = dbscan.fit_predict(X_scaled)

num_clusters = len(set(clusters)) - (1 if -1 in clusters else 0)
print("Кількість сформованих кластерів:", num_clusters)

# Visualize clusters on a map
plt.scatter(data['longitude'], data['latitude'],  c=clusters, cmap='viridis')
plt.xlabel('Довгота')
plt.ylabel('Широта')
plt.title('Gas Stations Clustering (DBSCAN)')
plt.grid(True)
plt.show()

for cluster in range(num_clusters):
    center = X.iloc[clusters == cluster].mean(axis=0)
    print(f"Центр кластеру {cluster}: {center}")

# Calculate and print average gas_price value of each cluster
for cluster in range(num_clusters):
    avg_gas_price = data.loc[(clusters == cluster) & (data['gas_type'] == 1),
'gas_price'].mean()
    print(f"Average A 95+ gas_price of cluster {cluster}:
{avg_gas_price:.2f}")
    avg_gas_price = data.loc[(clusters == cluster) & (data['gas_type'] == 2),
'gas_price'].mean()
    print(f"Average A 95 gas_price of cluster {cluster}: {avg_gas_price:.2f}")
    avg_gas_price = data.loc[(clusters == cluster) & (data['gas_type'] == 3),
'gas_price'].mean()
    print(f"Average A 92 gas_price of cluster {cluster}: {avg_gas_price:.2f}")
    avg_gas_price = data.loc[(clusters == cluster) & (data['gas_type'] == 4),
'gas_price'].mean()
    print(f"Average ДП gas_price of cluster {cluster}: {avg_gas_price:.2f}")
    avg_gas_price = data.loc[(clusters == cluster) & (data['gas_type'] == 5),
'gas_price'].mean()
    print(f"Average gas gas_price of cluster {cluster}: {avg_gas_price:.2f}")

cluster_sizes = [sum(clusters == cluster) for cluster in range(num_clusters)]
print("Розмір кожного кластера:", cluster_sizes)
```

Кластеризація DBSCAN 3D:

```
import pandas as pd
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D # Importing this module for 3D plot
from sklearn.cluster import DBSCAN
from sklearn.preprocessing import StandardScaler

# Read data from CSV file
data = pd.read_csv('gas_data.csv', delimiter=';', decimal=',')

# Select features for clustering
X = data[['gas_price']]
```

```
# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Apply DBSCAN clustering
eps = 0.028 # epsilon parameter for DBSCAN
min_samples = 3 # min_samples parameter for DBSCAN
dbscan = DBSCAN(eps=eps, min_samples=min_samples)
clusters = dbscan.fit_predict(X_scaled)

num_clusters = len(set(clusters)) - (1 if -1 in clusters else 0)
print("Number of formed clusters:", num_clusters)

# Visualize clusters in 3D
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

# Plot each point with corresponding cluster
for cluster in range(num_clusters):
    cluster_points = data.loc[clusters == cluster]
    ax.scatter(cluster_points['latitude'], cluster_points['longitude'],
cluster_points['gas_price'], label=f'Cluster {cluster}')

ax.set_xlabel('Latitude')
ax.set_ylabel('Longitude')
ax.set_zlabel('Gas Price')
ax.set_title('Gas Stations Clustering (DBSCAN)')
ax.legend()
plt.show()

cluster_sizes = [sum(clusters == cluster) for cluster in range(num_clusters)]
print("Розмір кожного кластера:", cluster_sizes)
```

Метод найближчого сусіда:

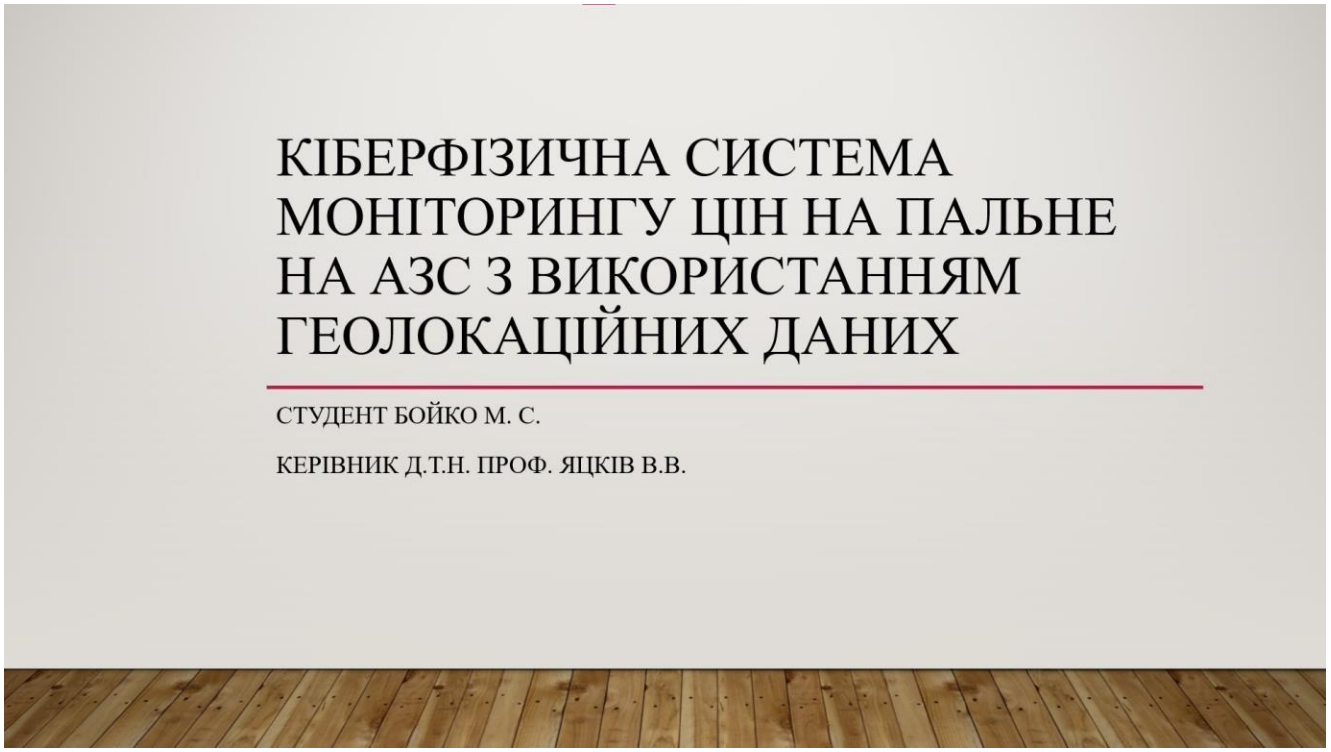
```
import pandas as pd

# Зчитуємо дані з CSV файлу
data = pd.read_csv('gas_data.csv', delimiter=';', decimal=',')

# Ціни за сьогоднішній день
gas_prices_today = {
    'А 95+': 56.33,
    'А 95': 52.15,
    'А 92': 49.13,
    'ДП': 52.41,
    'Газ': 27.67
}

# Функція для знаходження найближчого сусіда
def nearest_neighbor(gas_price):
    min_difference = float('inf')
    nearest_type = None
    for type_, price in gas_prices_today.items():
        difference = abs(gas_price - price)
        if difference < min_difference:
```

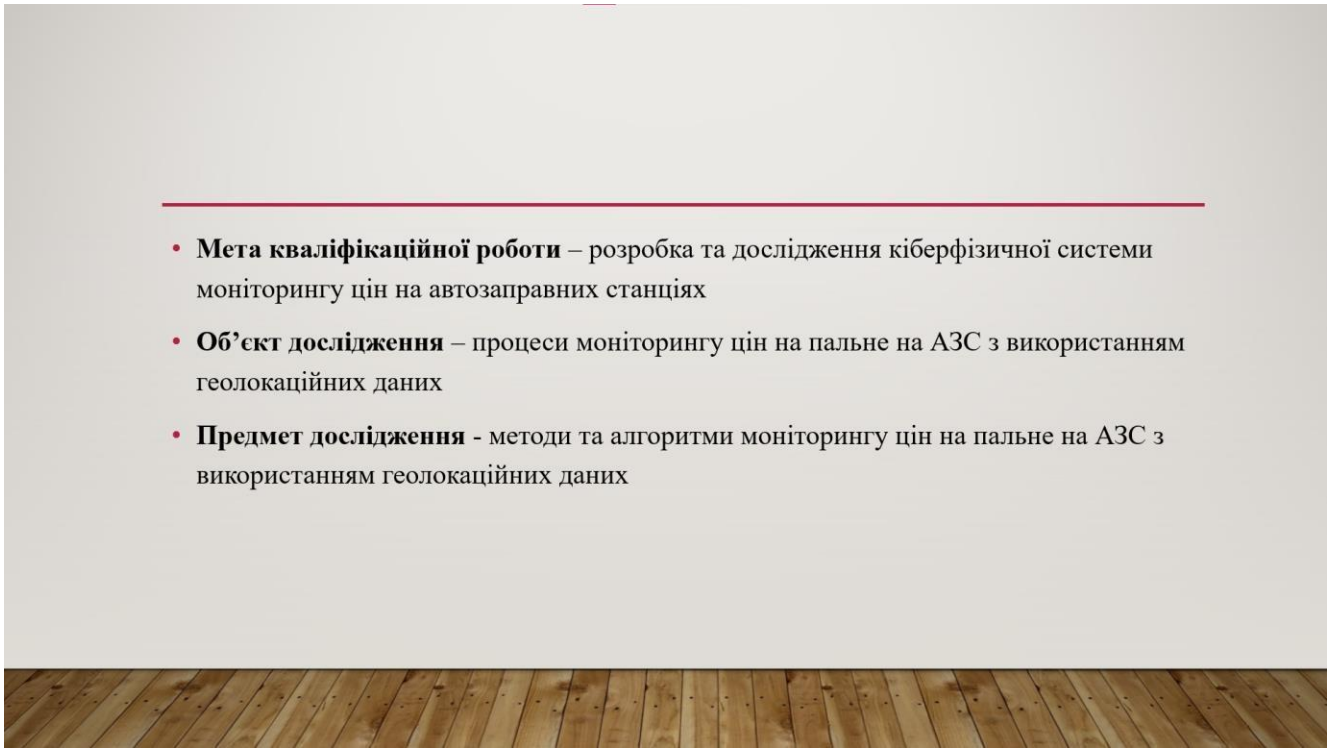
```
            min_difference = difference
            nearest_type = type_
    return nearest_type

# Додаємо новий стовпець, що вказує на найближчий тип пального за
сьогоднішніми цінами
data['nearest_gas_type'] = data['gas_price'].apply(nearest_neighbor)

print(data)

gas_type_counts = data['nearest_gas_type'].value_counts().to_dict()

print("Count of each gas type:")
for gas_type, count in gas_type_counts.items():
    print(f"{gas_type}: {count}")
```

ДОДАТОК В

(обов’язковий)

ПРЕЗЕНТАЦІЯ ДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ





НАУКОВА НОВИЗНА ОТРИМАНИХ РЕЗУЛЬТАТІВ

- вперше запропоновано та розроблено кіберфізичну систему спостереження в середовищі з обмеженою швидкістю інтернету, основна ціль якого – це моніторинг корисної інформації
- вперше розроблено метод автоматичного прийняття рішення периферійним вузлом, який забезпечує безвідмовну роботу підсистеми кіберфізичної системи спостереження

ПРАКТИЧНА ЗНАЧУЩІСТЬ ОТРИМАНИХ РЕЗУЛЬТАТІВ

- розробленні кіберфізичної системи кіберфізичної системи моніторингу цін на автозаправних станціях
- отримані практичні результати роботи підсистем кіберфізичної системи спостереження
- вичерпний аналіз ефективності використання підсистем кіберфізичної системи спостереження

ПУБЛІКАЦІЇ

- М. С. Бойко, В. В. Яцків. Кіберфізична система спостереження як частина інтернету транспортних засобів. Міжнародна студентська наукова конференція «Тренди та перспективи розвитку мультидисцип-лінарних досліджень»: Одеса, Україна, 2024. С. 159-161.
- M. Boiko, V. Yatskiv. Surveillance cyber-physical system as a part of internet of vehicles. Computer systems and information technologies. ISSN 2710-0766. 2024. Vol. 2.

АКТУАЛЬНІСТЬ ДОСЛІДЖЕННЯ

- Оптимізація вибору АЗС. Швидкий доступ до інформації про ціни на пальне на різних АЗС дозволить водіям ефективніше планувати свої маршрути та обирати АЗС з найбільш вигідними цінами. Це сприятиме зменшенню витрат на заправку автомобіля та покращить загальну економічну ситуацію водіїв.
- Підвищення конкурентоспроможності АЗС. Аналіз цінової політики конкурентів дозволить АЗС адаптувати свої ціни для привертання більшої кількості клієнтів. Це сприятиме підвищенню їхньої конкурентоспроможності на ринку пального.
- Економія часу та ресурсів. Застосування Кіберфізичних систем для моніторингу цін на пальне дозволить автоматизувати процес збору та аналізу даних, що значно заощадить час та ресурси. Це дозволить бізнесам та споживачам ефективніше використовувати свій час та зусилля.
- Екологічні переваги. Ефективне планування маршрутів може сприяти зменшенню викидів шкідливих речовин у атмосферу, оскільки дозволить уникнути зайвих поїздок та надмірного споживання пального.

ПОСТАНОВКА ЗАДАЧІ

1. Згідно виявлених проблем та аналізу існуючих рішень необхідно розробити модель кіберфізичної системи спостереження (КФСС), яка буде враховувати існуючі проблеми та функціональні вимоги до її підсистем. Розробити модель основного процесу КФС та її часову модель.
2. Вивчення та детальний аналіз методів необхідних для функціонування КФСС. Допускається розгляд відомих алгоритмів та програмного забезпечення які нададуть можливість впровадити розроблену КФС для моніторингу в реальне середовище.
3. Розробка основних функціональних можливостей КФСС, які надають можливість моніторингу цін на АЗС та комбінація всіх складових в одну систему.
4. Тестування та дослідження отриманих результатів роботи КФСС та її підсистем. Спростування, підтвердження та висунення гіпотез згідно результатів.

МОДЕЛЮВАННЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ МОНІТОРИНГУ ЦІН НА ПАЛЬНЕ НА АЗС З ВИКОРИСТАННЯМ ГЕОЛОКАЦІЙНИХ ДАНИХ

$$CPSS = \{hss_1, \dots, hss_m, fss_{m+1}, fss_{m+2}, bss_{m+3}, \dots, bss_{m+9}\},$$

де:

- hss_i – *периферійний вузол;*
- fss_i – *клієнтська підсистема;*
- bss_i – *серверна підсистема;*
- m – *кількість периферійних вузлів.*

МОДЕЛЮВАННЯ. МОДЕЛЬ ПЕРИФЕРІЙНОГО ВУЗЛА

$$HSS_i = \{CM, G, C, P, R\},$$

де:

- CM – *модуль зв'язку;*
- G – *GPS-модуль;*
- C – *камера стеження;*
- P – *джерело живлення;*
- R – *одноплатний комп'ютер;*
- i – *номер вузла.*

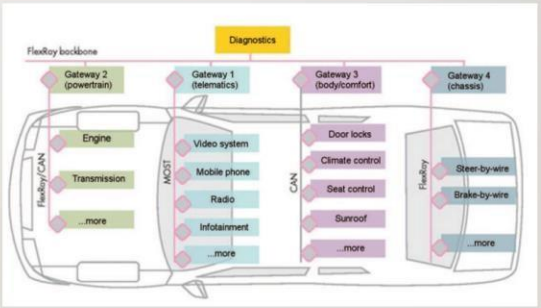


Рисунок 1 – Топологія автомобільної мережі

МОДЕЛЮВАННЯ. ЧАСОВА МОДЕЛЬ КФСС

Таблиця 1 - Позначення та визначення часової моделі

Позначення	Визначення
N	Кількість транспортних засобів з кіберфізичною системою
S	Набір периферійних систем, де S = {s_1, s_2, … s_N}
L	Кількість віртуальних машин в S_N
J	Кількість виконуваних завдань
V_N	Набір віртуальних машин в S_N, де V_N = {$v_{n,1}$, $v_{n,2}$, … $v_{n,L}$}
T	Набір виконуваних завдань, де T = {t_1, t_2, … t_J}
T_{total}	Загальні витрати часу на виконання завдань

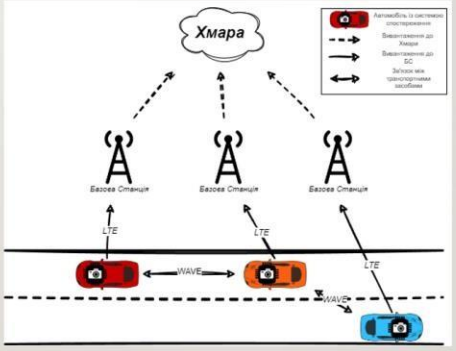


Рисунок 2 – Архітектура відеоспостереження з периферійними обчисленнями, розташованими в автомобілях

МОДЕЛЮВАННЯ. ЧАСОВА МОДЕЛЬ КФСС(2). ЧАС ОБЧИСЛЕННЯ

$$h_j = \sum_{n=1}^{N} k_{j,n} \frac{F_j}{x_n \cdot \varphi},$$

де:

- $k_{j,n}$ – *1 якщо дані* t_j *надійшли в* s_n, *в іншому випадку 0;*
- x_n – *кількість вільних віртуальних машин у* s_n;
- F_j – *тривалість відеозавдання;*
- φ – *операційна здатність кожної віртуальної машини.*

Позначення	Визначення
N	Кількість транспортних засобів з кіберфізичною системою
S	Набір периферійних систем, де S = {s_1, s_2, … s_N}
L	Кількість віртуальних машин в S_N
J	Кількість виконуваних завдань
V_N	Набір віртуальних машин в S_N, де V_N = {$v_{n,1}$, $v_{n,2}$, … $v_{n,L}$}
T	Набір виконуваних завдань, де T = {t_1, t_2, … t_J}
T_{total}	Загальні витрати часу на виконання завдань

МОДЕЛЮВАННЯ. ЧАСОВА МОДЕЛЬ КФСС(2). ЧАС ВИВАНТАЖЕННЯ РЕЗУЛЬТАТУ ОБЧИСЛЕНЬ

$$o_j = \sum_{n=1}^{N} k_{j,n} \left(\frac{f_{j,output}}{\eta} + \frac{f_{j,output}}{\lambda} \right),$$

де:

- $f_{j,output}$ – *вага завдання;*
- η – *швидкість передачі між периферійними вузлами і базовими станціями;*
- λ – *швидкість передачі між базовими станціями та хмарним центром обробки даних.*

Позначення	Визначення
N	Кількість транспортних засобів з кіберфізичною системою
S	Набір периферійних систем, де S = {s_1, s_2, … s_N}
L	Кількість віртуальних машин в S_N
J	Кількість виконуваних завдань
V_N	Набір віртуальних машин в S_N, де V_N = {$v_{n,1}$, $v_{n,2}$, … $v_{n,L}$}
T	Набір виконуваних завдань, де T = {t_1, t_2, … t_J}
T_{total}	Загальні витрати часу на виконання завдань

МОДЕЛЮВАННЯ. ЧАСОВА МОДЕЛЬ КФСС(3). ЗАГАЛЬНІ ВИТРАТИ ЧАСУ

$$T_{total} = \sum_{j=1}^{J} (h_j + o_j),$$

де:

- h_j – *час обчислення;*
- o_j – *час вивантаження результату обчислень;*
- J – *кількість виконуваних завдань.*

МЕТОДОЛОГІЯ ДОСЛІДЖЕННЯ КФСС

- методи комунікації між підсистемами КФСС;
- методи розпізнавання образів та тексту;
- методи геолокаційного позиціонування;
- метод автоматичного прийняття рішення підсистемою КФСС – периферійним вузлом;
- методи аналізу великих обсягів даних та виявлення аномальних значень у наборі даних.

МЕТОДИ КОМУНІКАЦІЇ МІЖ ПІДСИСТЕМАМИ КФСС

Методи комунікації між підсистемами описуються наступними зв’язками між:

- периферійними вузлами;
- периферійними вузлами та серверною підсистемою;
- серверною підсистемою та клієнтською підсистемою.

МЕТОДИ РОЗПІЗНАВАННЯ ОБРАЗІВ ТА ТЕКСТУ. АЛГОРИТМИ ВИЯВЛЕННЯ ОБ'ЄКТІВ НА ЗОБРАЖЕННІ

YOLO (YOU ONLY LOOK ONCE)

- **Точність.** Тяжко визначати об’єкти в групі.
- **Архітектура.** Конволюційну архітектура;
- **Робота з об'єктами різних розмірів.** Може мати проблеми з точним виявленням дуже малих або дуже великих об'єктів;
- **Вимоги до ресурсів.** Вимагає менше ресурсів для тренування та використання через свою більш просту архітектуру.

SSD (SINGLE SHOT MULTIBOX DETECTOR)

- **Точність.** Більш точний, коли об'єкти на зображеннях дуже малі або перекриваються
- **Архітектура.** Згорткові мережі для виявлення об'єктів на різних шарах зображення;
- **Робота з об'єктами різних розмірів.** Краще працює з об'єктами різних розмірів через свою мультишарову архітектуру;
- **Вимоги до ресурсів.** Більш вимогливий до обчислювальних ресурсів.

МЕТОДИ РОЗПІЗНАВАННЯ ОБРАЗІВ ТА ТЕКСТУ(2). АНАЛІЗ СЕРВІСІВ ОПТИЧНОГО РОЗПІЗНАВАННЯ СИМВОЛІВ

- неправильне зчитування логотипу;
- поява стороннього символу “E”;
- неправильна розпізнаний символ. Символ П визначено як О;
- опущено цифру “1” в числі 21.49.

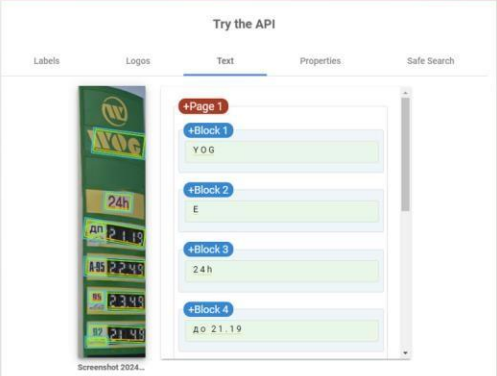


Рисунок 3 – Результат тестування розпізнавання тексту з Google Cloud Vision API

МЕТОД АВТОМАТИЧНОГО ПРИЙНЯТТЯ РІШЕННЯ ПІДСИСТЕМОЮ КФСС – ПЕРИФЕРІЙНИМ ВУЗЛОМ

Умови успішного прийняття рішення підсистемою:

- якщо периферійний вузол розпізнав об’єкт за допомогою нейронної мережі, тобто $C > 0.4$, то $f=f+1$ та $d[0] = 1$, інакше $d[0] = 0$;
- якщо система геолокаційного позиціонування визначила місцезнаходження периферійного вузла, тобто $(x, y) \neq NULL$ і $G_{update_time} \neq NULL$, то $f=f+1$ та $d[1]=1$, інакше $d[1]=0$;
- якщо периферійний вузол завершив обробку зображення, тобто виділив необхідну частину зображення для відправки на сервер, то $f=f+1$ та $d[2] = 1$, інакше $d[2]=0$;
- якщо периферійний вузол має доступ до інтернету, тобто $CM_{state}=1$ і $CM_{speed} >0$, то $f=f+1$ та $d[3]=1$, інакше $d[3]=0$.

МОЖЛИВІ ДІЇ ПІДСИСТЕМИ ЩОДО СФОРМОВАНИХ УМОВ В МАСИВІ D

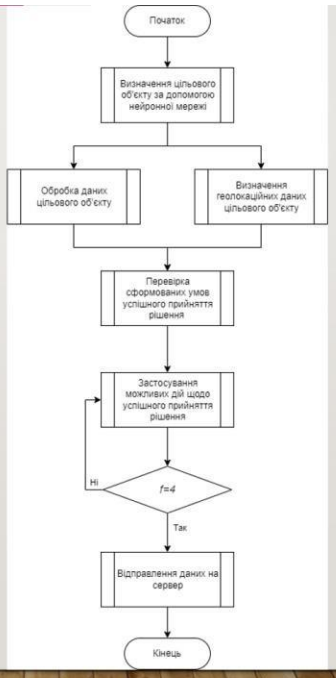
- якщо *d[0]=0*, то підсистема не визначила жодного об'єкта і виконання наступних дій по автоматичному прийняттю рішення не має сенсу;
- якщо *d[0]=1*, то підсистема розпізнала необхідний об'єкт і переходить до перевірки наступної умови;
- якщо *d[1]=0*, то підсистема не визначила своє поточне місцезнаходження, а отже для передачі геолокаційних даних необхідно відтермінувати відправку даних до появи сигналу або скористатися останніми доступними даними по розташуванню транспортного засобу; після описаних дій значення *d[1]=1*;
- якщо *d[1]=1*, то підсистема визначила найближче положення до зробленого знімку і переходить до перевірки наступної умови;
- якщо *d[2]=0*, то підсистема не здійснила фіксацію поточного часу та кадру з відеопотоку, це ініціює повторну обробку зображення зі зміненими параметрами, допоки *d[2]=1*;
- якщо *d[2]=1*, то підсистема завершила обробку даних і переходить до перевірки наступної умови;
- якщо *d[3]=0*, то підсистема відкладає відправку даних щодо визначеного об'єкта та з визначеним інтервалом перевіряє чи не з'явився доступ до інтернету;
- якщо *d[3]=1*, то підсистема відправила дані на віддалений сервер і завершує свої дії щодо визначених умов.

МЕТОД АВТОМАТИЧНОГО ПРИЙНЯТТЯ РІШЕННЯ

Згідно описаних умов та необхідних дій підсистеми сформуємо метод автоматичного прийняття рішення та відобразимо його у вигляді подальших кроків:

1. визначення цільового об'єкту(в нашому випадку це табло з цінами на АЗС) за допомогою нейронної мережі;
2. паралельна обробка даних цільового об'єкту з визначенням його геолокаційних даних;
3. перевірка сформованих умов успішного прийняття рішення підсистемою;
4. застосування можливих дій щодо сформованих умов успішного прийняття рішення по відправці даних;
5. генерація рішення на основі розроблених умов та дій щодо них:
   - Якщо $f = 4$, то підсистема приймає рішення по відправлені даних на сервер для подальшої обробки даних серверною підсистемою.
   - Якщо $f \neq 4$, то підсистема повторно перевіряє масив умов і згідно описаних раніше дій приймає рішення.

МЕТОД(2). АЛГОРИТМ



КІБЕРФІЗИЧНА СИСТЕМА МОНІТОРИНГУ ЦІН НА ПАЛЬНЕ НА АЗС З ВИКОРИСТАННЯМ ГЕОЛОКАЦІЙНИХ ДАНИХ

- огляд технічної архітектури периферійного вузла;
- тренування нейронної мережі YOLO по розпізнаванню образів;
- результати роботи підсистем.

ТЕХНІЧНА АРХІТЕКТУРА ПЕРИФЕРІЙНОГО ВУЗЛА

- *Модуль зв'язку*: Смартфон з 4G;
- *GPS-модуль*: NEO-6M v2;
- *Камера*: Logitech Webcam C270 HD;
- *Джерело живлення*: Xiaomi Redmi Power Bank;
- *Одноплатний комп'ютер*: Raspberry Pi 4 Model B 4GB;

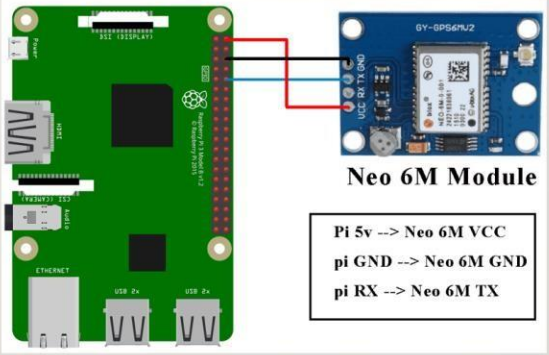
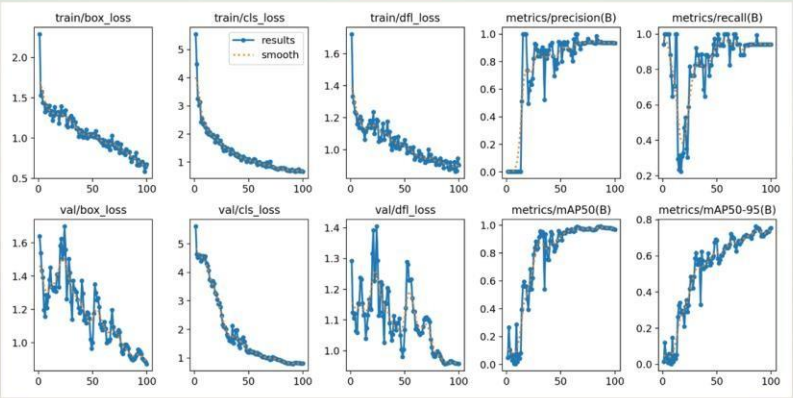


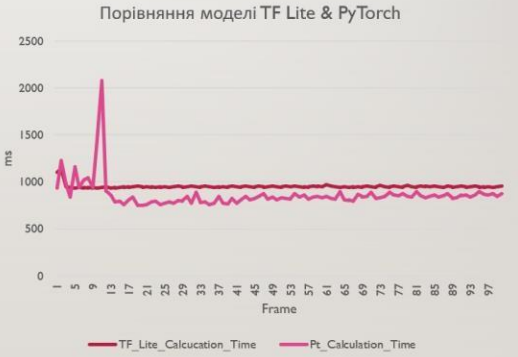
Рисунок 4 – Схема підключення GPS-модуля і Raspberry Pi

ТРЕНУВАННЯ НЕЙРОННОЇ МЕРЕЖІ YOLO ПО РОЗПІЗНАВАННЮ ОБРАЗІВ



ТРЕНУВАННЯ НЕЙРОННОЇ МЕРЕЖІ YOLO ПО РОЗПІЗНАВАННЮ ОБРАЗІВ

- Середній час на обрахунок моделлю TF Lite становить **953.033 мс**;
- Завантаження процесора при використанні TF Lite від **110** до **120 %**;
- Середній час на обрахунок моделлю PyTorch становить **863.844 мс**;
- Завантаження процесора при використанні PyTorch від **360** до **390 %**



РЕЗУЛЬТАТИ РОБОТИ ПЕРИФЕРІЙНОГО ВУЗЛА

Таблиця 2 – Результати тестування підсистеми на фото

№ Фото	1	2	3	4	5	6	7	8	9	10
Впевненість	0.71	0.81	0.75	-	0.83	0.56	-	0.44	0.39	0.55

Таблиця 3 – Результати тестування підсистеми з відеопотоками

№ Експерименту	№ АЗС			
	1	2	3	4
1	1	1	1	0
2	1	1	1	1
3	1	1	1	0

РЕЗУЛЬТАТИ РОБОТИ СЕРВЕРНОЇ ПІДСИСТЕМИ

Як видно з результатів оптичного розпізнавання при кожному експерименті спостерігається втрата певної кількості інформації, а саме:

- для зображення № 1 ≈ 9.375 %;
- для зображення № 2 ≈ 16.7%;
- для зображення № 3 ≈ 20.84%;
- для зображення № 4 ≈ 5.71%;
- для зображення № 5 ≈ 11.11%.

Таблиця 4 – Результати тестування підсистеми на оптичне розпізнавання символів

№ Зображення	Правильно визначено	Необхідно було визначити
1	29	32
2	20	24
3	19	24
4	33	35
5	16	18

РЕЗУЛЬТАТИ РОБОТИ СЕРВЕРНОЇ ПІДСИСТЕМИ(2). КЛАСТЕРИЗАЦІЯ ЗА ГЕОЛОКАЦІЙНИМИ ДАНИМИ

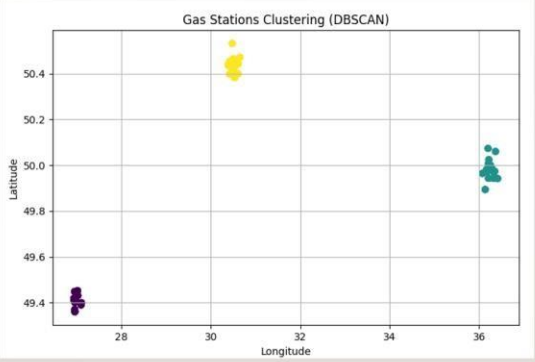


Рисунок 5 – Результати кластеризації за координатами АЗС в 2D

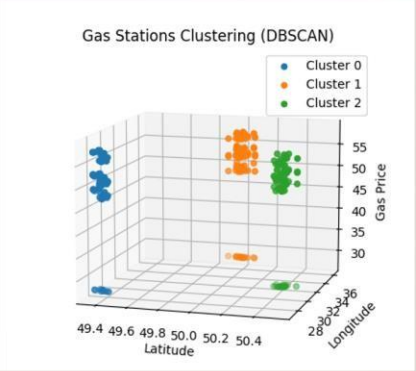


Рисунок 6 – Результати кластеризації за координатами АЗС в 3D

РЕЗУЛЬТАТИ РОБОТИ СЕРВЕРНОЇ ПІДСИСТЕМИ(3). КЛАСТЕРИЗАЦІЯ ЗА ЦІНАМИ

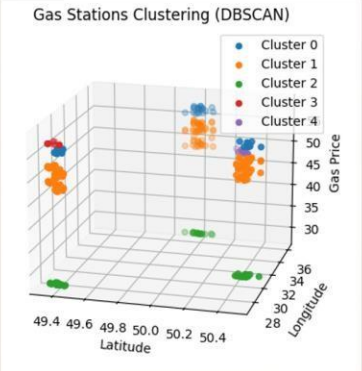


Рисунок 7 – Результати кластеризації за цінами на АЗС в 3D

Таблиця 5 – Результати тестування підсистеми на кластеризацію даних за цінами на АЗС

Очікуваний результат	Кластеризація DBSCAN	Метод найближчого сусіда
[59, 59, 59, 59, 59]	[40, 177, 59, 9, 10]	[61, 59, 59, 59, 57]

ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень розроблено підсистеми кіберфізичної системи моніторингу цін на АЗС.

- У першому розділі проведений огляд застосувань кіберфізичних систем в різних сферах діяльності та виокремлено кіберфізичні системи в сфері інтернету транспортного засобу як основний орієнтир в подальшій кваліфікаційній роботі. Проведено порівняльний аналіз переваг та недоліків існуючих рішень, в яких розглядалися наукові роботи з сфери інтернету транспортних засобів та кіберфізичних систем спостереження, на основі яких визначено методологічні підходи та здійснена постановка задачі.
- У другому розділі розроблено модель кіберфізичної системи спостереження, в якій її було розділено на три підсистеми та розписано складові кожної з підсистем в окремо розроблених моделях. Також в даному розділі було розроблено модель основного процесу та часову модель КФСС.

ВИСНОВКИ(2)

- У третьому розділі визначено необхідну методологію для реалізації кіберфізичної системи спостереження, а саме методи комунікації між підсистемами; методи розпізнавання образів і тексту; методи визначення геолокаційного позиціонування; методи аналізу великих даних. В даному розділі було сформовано новий метод автоматичного прийняття рішення підсистемою КФСС – периферійним вузлом, необхідність якого зумовлена технічними обмеженнями визначених методів комунікації між підсистемами.
- У четвертому розділі розглянуто технічну архітектуру двох підсистем кіберфізичної системи спостереження, кожну з них реалізовано та досліджено на практиці, за результатами якої підтверджено дві гіпотези, спростовано дві гіпотези та висунуто одну нову гіпотезу.

UNICHECK
by Turnitin

Ім'я користувача:
Кафедра КІ

ID перевірки:
1016240192

Дата перевірки:
09.05.2024 20:51:28 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
09.05.2024 20:58:49 EEST

ID користувача:
100005591

Назва документа: Бойко_Кіберфізична система моніторингу цін на пальне на АЗС з використанням геолокац...

Кількість сторінок: 95 Кількість слів: 18229 Кількість символів: 139005 Розмір файлу: 2.65 MB ID файлу: 1016022883

1.34% Схожість

Найбільша схожість: 0.59% з джерелом з Бібліотеки (ID файлу: 1011139258)

0.98% Джерела з Інтернету 45 ..... Сторінка 97

0.97% Джерела з Бібліотеки 38 ..... Сторінка 97

0.05% Цитат

Цитати 5 ..... Сторінка 98

Посилання 1 ..... Сторінка 98

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 27

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 0.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 13%**

ID: 125941 Назва: МКР Кіберфізична система моніторингу цін на пальне на АЗС з використанням геолокаційних даних Додано в БД: 2024-05-09 Автора: Бойко М.С. Керівники: Яцків В.В. Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	117199	964	1301 (1%)	18 (2%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Бойко Микита Сергійович

Тема: Кіберфізична система моніторингу цін на пальне на АЗС з використанням геолокаційних даних

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень – Кількість сторінок записки 81

1. Короткий зміст роботи та прийнятих рішень: *у роботі запропоновано та розроблено кіберфізичну систему спостереження з застосуванням геолокаційних даних цілю якої було визначено АЗС.*

2. Висновок про відповідність роботи дипломному завданню: *кваліфікаційна робота магістра відповідає виданому завданню.*

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: *у першому розділі проведено огляд концепції кіберфізичних систем та їх застосування у середовищі інтернету транспортних засобів. Досліджено відомі рішення та засоби в цій сфері. У другому розділі запропоновано модель кіберфізичної системи спостереження, в якій її було розділено на три підсистеми також у даному розділі було розроблено модель основного процесу та часову модель КФСС. У третьому розділі запропоновано метод визначено необхідну методологію для реалізації кіберфізичної системи спостереження, а саме методи комунікації між підсистемами; методи розпізнавання образів і тексту; методи визначення геолокаційного позиціонування; методи аналізу великих даних, також було сформовано новий метод автоматичного прийняття рішення підсистемою КФСС – периферійним вузлом. У четвертому розділі запропоновано розглянуто технічну архітектуру двох підсистем кіберфізичної системи спостереження, кожну з них реалізовано та досліджено на практиці.*

4. Позитивні сторони роботи: *Робота виявляє глибоке розуміння теми та добре пророблену роботу з літературними джерелами. Робота включає детальний аналіз отриманих результатів, що дозволяє зробити обгрунтовані висновки та рекомендації. Використані дані, статистика або інші докази підтверджують висунуті тези та допомагають підтримати аргументацію.*

5. Негативні сторони роботи: _____

6. Оцінка графічного оформлення та пояснювальної записки роботи: –

7. Відгук про роботу в цілому: *Кваліфікаційна робота представляє собою детальне та досконало дослідження проблеми, що виконане на високому рівні. Автор демонструє глибоке розуміння теми та використовує чітко визначені методи дослідження, що сприяють об'єктивному аналізу. Важливо відзначити його інноваційний підхід до вирішення проблеми, що відображається в оригінальних методах та новаторських підходах. Робота має чітку структуру та послідовний виклад матеріалу, що полегшує розуміння її змісту. Отримані результати дослідження є важливим внеском у відповідну галузь і можуть слугувати основою для подальших досліджень та розвитку теми. У цілому, кваліфікаційна робота вражає своєю глибиною та професійним підходом до дослідження та заслуговує на високу оцінку.*

8. Інші зауваження: –

9. Оцінка дипломної роботи: *Розглянувши позитивні та негативні сторони представленої кваліфікаційної роботи вважаю, що робота заслуговує оцінки «відмінно» 4.75 (A)*

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Бармак Олександр Володимирович д.т.н., професор, завідувач кафедри комп'ютерних наук ХНУ

"8" травня 2024 р. _____(підпис)

Завідувачу кафедри КІІС
д-р.техн.наук, проф. Говорущенко Т. О.

Бойко Микита Сергійович

ПІБ здобувача вищої освіти

ФІТ, 2 курсу, групи КІ2м-22-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

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

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

8 травня 2024 року

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОМАЦІЙНИХ СИСТЕМ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Кіберфізична система моніторингу цін на пальне на АЗС з використанням геолокаційних даних

Автор: Бойко Микита Сергійович

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-наукова

Науковий керівник: Яцків В.В., д.т.н, професор

Після аналізу звіту подібності зроблено такий висновок:

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

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) запозичення розміщені в розділах мають збіг зі звітом з науково-дослідної практики автора Бойко Микити "Кіберфізична система моніторингу цін на пальне на АЗС з використанням геолокаційних даних", який було додано в репозитраій ХНУ 21 березня 2024 року;
2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
3) окремі виявлені збіги є загальновживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості Unicheck, складає 1.34% і адресується до 15 першоджерела; та системою Anti-Plagiarism складає 1%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ В. В. Яцків

Гарант ОП _____ О. С. Савенко

Завідувач кафедри КІС _____ Т. О. Говорущенко