

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет**  
**імені Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інтелектуальних інформаційних систем**

**ДОПУЩЕНО ДО ЗАХИСТУ**

Завідувач кафедри інтелектуальних  
інформаційних систем, д-р техн. наук, проф.

\_\_\_\_\_ Ю. П. Кондратенко

« \_\_\_\_ » \_\_\_\_\_ 2024 р.

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

**ВЕБРЕСУРС ДЛЯ ПОКРАЩЕННЯ КОМФОРТНОГО ПЕРЕВЕЗЕННЯ**  
**ПАСАЖИРІВ АВТОБУСІВ**

Спеціальність 122 «Комп'ютерні науки»

**122 – БКР – 402.2010212**

*Виконав студент 4-го курсу, групи 402*

\_\_\_\_\_ *А. А. Кириленко*

«17» червня 2024 р.

*Керівник: д-р техн. наук, проф*

\_\_\_\_\_ *Е. А. Лисенков*

«17» червня 2024 р.

**Миколаїв – 2024**

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет ім. Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інтелектуальних інформаційних систем**

Рівень вищої освіти     **бакалавр**  
Спеціальність     **122 «Комп'ютерні науки»**  
                                      *(шифр і назва)*  
Галузь знань     **12 «Інформаційні технології»**  
                                      *(шифр і назва)*

**ЗАТВЕРДЖУЮ**

Завідувач кафедри інтелектуальних  
інформаційних систем, д-р техн. наук, проф.  
\_\_\_\_\_  
Ю. П. Кондратенко  
« \_\_\_\_ » \_\_\_\_\_ 2024 р.

**ЗАВДАННЯ**  
**на виконання кваліфікаційної роботи**

Видано студенту групи 402 факультету комп'ютерних наук Кириленко Артему Андрійовичу.

1. Тема кваліфікаційної роботи «Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту».

Керівник роботи Лисенков Едуард Анатолійович, д-р техн. наук, професор.

Затв. наказом Ректора ЧНУ ім. Петра Могили від «28» грудня 2023 р. № 271

2. Строк представлення кваліфікаційної роботи студентом «17» червня 2024 р.

3. Вхідні (початкові) дані до роботи: експертні оцінки інформаційних транспортних систем та моделей машинного навчання з метою покращення комфортного перевезення пасажирів автобусного транспорту.

Очікуваний результат: вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту

4. Перелік питань, що підлягають розробці (зміст пояснювальної записки):

– провести огляд сучасного стану інформаційної сфери автобусних перевезень та виявити проблемні аспекти.

– дослідити наявні вебресурси та застосунки, спрямовані на покращення комфорту пасажирів автобусного транспорту.

– розробити базу даних та алгоритми обробки інформації про маршрути та розклади;

– розробити програмний код та функціональність вебресурсу з урахуванням вимог пасажирів та адміністраторів.

5. Перелік графічного матеріалу: презентація.

6. Завдання до спеціальної частини: «Охорона праці при перевезенні пасажирів автобусів»

7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис
Спеціальна частина з охорони праці	Канд. техн. наук, доцент Алексєєва А. О.	
Методична частина	д-р техн. наук, проф. Лисенков Е.А	

Керівник роботи д-р техн. наук, проф. Лисенков Е.А.

*(наук. ступінь, вчене звання, прізвище та ініціали)*

\_\_\_\_\_

*(підпис)*

Завдання прийнято до виконання Кириленко А.А.

*(прізвище та ініціали)*

\_\_\_\_\_

*(підпис)*

Дата видачі завдання « 14 » січня 2024 р.

**КАЛЕНДАРНИЙ ПЛАН**  
**виконання кваліфікаційної роботи**

Тема: Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту

№	Найменування роботи	Початок	Закінчення	Примітки
1	Подання заяви на затвердження теми та керівників КРБ	10.11.2023	15.11.2023	Виконано
2	Отримання завдання на виконання КРБ	10.01.2024	15.01.2024	Виконано
3	Складання календарного плану роботи на весь період виконання КРБ	16.01.2024	30.01.2024	Виконано
4	Отримання завдання на переддипломну практику	15.04.2024	29.04.2024	Виконано
5	Проходження переддипломної практики, збір та аналіз матеріалів до КРБ	29.04.2024	11.05.2024	Виконано
6	Розробка звіту з переддипломної практики	12.05.2024	15.05.2024	Виконано
7	Виконання КРБ: огляд наявних технологій, розробка ПЗ	13.05.2024	22.06.2024	Виконано
8	Перший попередній захист КРБ на засіданні комісії кафедри	27.05.2024	27.05.2024	Виконано
9	Доробка та остаточне оформлення КРБ	28.05.2024	09.06.2024	Виконано
10	Другий попередній захист КРБ на засіданні комісії кафедри	10.06.2024	10.06.2024	Виконано
11	Подання КРБ рецензенту	13.06.2024	13.06.2024	Виконано
11	Подання КРБ, її електронної копії та інших документів (відгуку, рецензії) до захисту	17.06.2024	21.06.2024	Виконано
12	Захист БКР перед екзаменаційною комісією (ЕК)	24.06.2024	28.06.2024	Виконано

Розробив студент Кириленко Артем Андрійович \_\_\_\_\_  
(прізвище, ім'я, по батькові студента) (підпис)

Керівник роботи д-р фіз.-мат. наук, проф Е.А. Лисенков \_\_\_\_\_  
(посада, прізвище, ім'я, по батькові) (підпис)

« 29 » \_\_\_\_\_ 01 \_\_\_\_\_ 2024 р.

## АНОТАЦІЯ

Актуальність даного дослідження виявляється у зростаючій потребі в оптимізації комфорту та зручності для пасажирів в автобусному транспорті через швидке розвиток технологій і змінювані вимоги споживачів. Дослідження оглядає сучасний стан інформаційної сфери автобусних перевезень, досліджує наявні вебресурси та застосунки, а також систематично оглядає літературу щодо використання машинного навчання у цій сфері.

Об'єктом дослідження є процес покращення комфорту та зручності для пасажирів автобусного транспорту.

Предметом дослідження є інформаційні технології та методи оптимізації комфорту автобусних перевезень, включаючи вебресурси, бази даних, алгоритми обробки інформації та програмну реалізацію.

Метою роботи є створення вебресурсу, який сприятиме покращенню комфорту та зручності для пасажирів автобусного транспорту шляхом оптимізації інформаційного забезпечення та використання сучасних технологій.

Пояснювальна записка складається зі вступу, чотирьох розділів, висновків та додатків.

У першому розділі проводиться теоретичний аналіз інформаційних аспектів забезпечення комфорту та зручності для пасажирів в сучасному автобусному транспорті.

У другому розділі проводиться аналіз методів та технологій, спрямованих на оптимізацію комфорту в автобусних перевезеннях.

У третьому описується процес моделювання та технічного проектування вебресурсу з метою оптимізації комфорту автобусних перевезеннях.

У четвертому розділі представлено процес програмної реалізації вебресурсу, а також розробку відповідної документації.

Дипломна робота складається з 82 сторінок, включаючи 13 рисунків та 2 додатків. У роботі використано 31 джерел.

*Ключові слова: автобусні перевезення, комфорт пасажирів, машинне навчання, інформаційні технології, оптимізація.*

## ABSTRACT

The relevance of this study is manifested in the growing need to optimize passenger comfort and convenience in bus transport due to the rapid development of technology and changing consumer requirements. The study reviews the current state of the information sphere of bus transportation, explores existing web resources and applications, and systematically reviews the literature on the use of machine learning in this area. The object of the study is the process of improving the comfort and convenience for bus passengers.

The subject of the study is information technologies and methods for optimizing the comfort of bus transportation, including web resources, databases, information processing algorithms, and software implementation. The purpose of the work is to create a web resource that will improve the comfort and convenience of bus passengers by optimizing information support and using modern technologies.

The explanatory note consists of an introduction, four chapters, conclusions, and appendices.

The first section provides a theoretical analysis of the information aspects of ensuring comfort and convenience for passengers in modern bus transport. The second section analyzes methods and technologies aimed at optimizing comfort in bus transportation.

The third section describes the process of modeling and technical design of a web resource to optimize the comfort of bus transportation. The fourth chapter presents the process of programmatic implementation of the web resource, as well as the development of relevant documentation.

The thesis consists of 82 pages, including 13 figures and 2 appendixes. The work uses 31 sources.

*Keywords: bus transportation, passenger comfort, machine learning, information technology, optimization.*

## ЗМІСТ

ВСТУП.....	2
1 ТЕОРЕТИЧНІ ЗАСАДИ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ КОМФОРТУ ТА ЗРУЧНОСТІ ДЛЯ ПАСАЖИРІВ НА СУЧАСНОМУ АВТОБУСНОМУ ТРАНСПОРТІ.....	6
1 1.1. Огляд сучасного стану інформаційної сфери автобусних перевезень .....	6
1.2. Дослідження наявних вебресурсів та застосунків для покращення комфорту пасажирів автобусних перевезень .....	12
1.3. Систематичний огляд літератури щодо використання машинного навчання у сфері громадських автобусних перевезень .....	17
2 АНАЛІЗ МЕТОДІВ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ОПТИМІЗАЦІЇ КОМФОРТУ АВТОБУСНИХ ПЕРЕВЕЗЕНЬ .....	28
2.1. Визначення математичних моделей для оптимізації комфорту пасажирів .....	28
2.2. Аналіз методів та інформаційних технологій для реалізації вебресурсу.....	35
3 МОДЕЛЮВАННЯ ТА ТЕХНІЧНЕ ПРОЄКТУВАННЯ ВЕБРЕСУРСУ ДЛЯ ОПТИМІЗАЦІЇ КОМФОРТУ АВТОБУСНИХ ПЕРЕВЕЗЕНЬ.....	43
3.1. Визначення бази даних та алгоритмів обробки інформації про маршрути та розклади.....	43
3.2. Розробка моделі машинного навчання для з метою прогнозування часу руху автобусів .....	46
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА РОЗРОБКА ДОКУМЕНТАЦІЇ ВЕБРЕСУРСУ ДЛЯ ОПТИМІЗАЦІЇ КОМФОРТУ АВТОБУСНИХ ПЕРЕВЕЗЕНЬ.....	50
4.1. Вибір мов програмування, фреймворків та інших програмних інструментів ..	50
4.2. Детальний опис програмного коду та функціональності вебресурсу .....	53
ВИСНОВКИ .....	65

## ВСТУП

Актуальність даного дослідження виявляється у зростаючій потребі в оптимізації комфорту та зручності для пасажирів в автобусному транспорті через швидке розвиток технологій і змінювані вимоги споживачів. Сучасні пасажирів мають високі очікування щодо доступності інформації, зручності у виборі маршрутів та комфорту під час поїздок. Дослідження оглядає сучасний стан інформаційної сфери автобусних перевезень, досліджує наявні вебресурси та застосунки, а також систематично оглядає літературу щодо використання машинного навчання у цій сфері.

Аналіз методів та інформаційних технологій для оптимізації комфорту автобусних перевезень дозволить виявити найефективніші підходи та технології для створення вебресурсу. Моделювання та технічне проектування вебресурсу дозволить розробити ефективну систему збору та обробки інформації про маршрути та розклади, що відповідатиме потребам сучасних пасажирів.

Програмна реалізація вебресурсу та розробка відповідної документації допоможе забезпечити просте та зручне користування пасажирами та адміністраторами.

Отже, дане дослідження має велике значення для подальшого розвитку і покращення комфорту пасажирів автобусного транспорту, а також для вдосконалення інформаційного забезпечення та ефективності в цій галузі.

Метою роботи є створення вебресурсу, який сприятиме покращенню комфорту та зручності для пасажирів автобусного транспорту шляхом оптимізації інформаційного забезпечення та використання сучасних технологій.

Завдання, які мають бути розв'язані для досягнення мети:

1. Провести огляд сучасного стану інформаційної сфери автобусних перевезень та виявити проблемні аспекти.
2. Дослідити наявні вебресурси та застосунки, спрямовані на покращення комфорту пасажирів автобусного транспорту.



3. Провести систематичний огляд літератури з використання машинного навчання у сфері громадських автобусних перевезень.
4. Визначити математичні моделі для оптимізації комфорту пасажирів та аналізувати методи та інформаційні технології для реалізації вебресурсу.
5. Розробити базу даних та алгоритми обробки інформації про маршрути та розклади.
6. Вибрати мови програмування, фреймворки та інші програмні інструменти для програмної реалізації вебресурсу.
7. Розробити програмний код та функціональність вебресурсу з урахуванням вимог пасажирів та адміністраторів.
8. Підготувати інструкцію з використання вебресурсу для пасажирів та адміністраторів.

Об'єктом дослідження є процес покращення комфорту та зручності для пасажирів автобусного транспорту.

Предметом дослідження є інформаційні технології та методи оптимізації комфорту автобусних перевезень, включаючи вебресурси, бази даних, алгоритми обробки інформації та програмну реалізацію.

Теоретичною основою дослідження є вивчення та переосмислення актуальних підходів та концепцій у сфері інформаційного забезпечення комфорту та зручності для пасажирів на сучасному автобусному транспорті. Для досягнення поставленої мети дослідження використовуються наступні методи: методи аналізу та синтезу: для огляду сучасного стану інформаційної сфери автобусних перевезень, дослідження наявних вебресурсів та застосунків для покращення комфорту пасажирів, а також систематичного огляду літератури щодо використання машинного навчання у сфері громадських автобусних перевезень; методи аналізу методів та інформаційних технологій: для визначення математичних моделей оптимізації комфорту пасажирів та аналізу можливих методів та інформаційних технологій для реалізації вебресурсу; методи

## Кафедра інтелектуальних інформаційних систем

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту моделювання та технічного проектування: для визначення бази даних та алгоритмів обробки інформації про маршрути та розклади; методи програмної реалізації та розробки документації: для вибору мов програмування, фреймворків та інших програмних інструментів, розробки програмного коду та функціональності вебресурсу, а також створення інструкції з використання вебресурсу для пасажирів та адміністраторів.

Ці методи дозволяють систематизувати та дослідити різноманітні аспекти інформаційного забезпечення комфорту та зручності для пасажирів автобусного транспорту та розробити відповідний вебресурс для оптимізації цих процесів.

# **1 ТЕОРЕТИЧНІ ЗАСАДИ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ КОМФОРТУ ТА ЗРУЧНОСТІ ДЛЯ ПАСАЖИРІВ НА СУЧАСНОМУ АВТОБУСНОМУ ТРАНСПОРТІ**

## **1.1. Огляд сучасного стану інформаційної сфери автобусних перевезень**

Обсяг і швидкість, з якою сьогодні генеруються дані про транспорт і автобусні перевезення зокрема, значно перевищили масштаби, з якими вони збиралися, оброблялися й аналізувалися на початку цього століття. Поєднання нових парадигм цифровізації, таких як Інтернет речей (IoT), поширення розумних міст, різке зниження вартості зберігання даних, останні розробки у сфері бездротових технологіях, а також недороге та широке використання датчиків або персональних пристроїв значно покращила здатність людини осягати більш детальне розуміння транспортної та інформаційної реальності завдяки більш різноманітним, вдосконаленим способам збору, передачі, зберігання, об'єднання, пошуку та обробки транспортних даних [19, с. 250–253].

Через обсяг, різноманіття (джерело, тип і формат) і мінливість (часті зміни даних, що ускладнює розшифровку їх точного значення в їхньому контексті) транспортних даних, такі інтенсивні завдання, як інтеграція, візуалізація, запити та аналіз, великомасштабні системи реального часу стають все більш важливими при розробці та впровадженні інтелектуальних транспортних систем (ІТС). На жаль, загально визнано, що поточні програми ІТС мають обмежені функції моніторингу та аналізу даних. Як наслідок, вищезазначені виклики вимагають реалістичного, ефективного моніторингу, прийняття рішень і функцій управління даними, вимогам до яких загалом не відповідають розгорнуті на даний момент ІТС. Крім того, масова поява потокових джерел даних у режимі реального часу може призвести до безпрецедентного уявлення та покращення транспортних послуг, діяльності та операцій, пов'язаних із політикою, після розробки передових підходів агрегації, злиття та інкрементального навчання.

У цьому контексті Big Data та машинне навчання постулюються як нові ключові технології для транспортної галузі. Рішення та інструменти в рамках цієї нової технологічної парадигми можуть отримувати, керувати та аналізувати величезні обсяги структурованих і неструктурованих даних для покращення транспортну сферу та вирішити поставлені вище проблеми. Відповідь передбачає створення нових ІТС на основі вилучення цінності та знань із нової величезної кількості доступних даних. Кінцева мета полягає в тому, щоб транспортна галузь отримувала цінність від своїх даних, які для конкретного випадку транспортування перетворюються на майбутні розробки, зосереджені на способах надання безпечніших, чистіших і ефективніших методів транспортування та приємних перевезень для кінцевих користувачів (пасажирів). Крім того, прогнозна аналітика, не менш популярний технологічний аналог великих даних – і, по суті, складова частина останніх – дозволяє збирати та досліджувати дані, щоб виявити невидимі закономірності в транспортних проблемах, що дозволяє використовувати превентивні служби та програми з чіткими з'єднаннями. Наприклад, додатки, що використовують великі дані в транспорті, включають моніторинг інфраструктури, розробку послуг із доданою вартістю мобільності, краще розуміння потреб користувачів або візуалізацію потоків людей та їх еволюцію в містах.

Інформаційні технології приходять у транспортну галузь як для вирішення традиційних проблем, пов'язаних із даними, так і для того, щоб запропонувати нові можливості, програми та послуги завдяки новим джерелам даних і технологіям. Прикладом першого є обробка даних для контролю трафіку в реальному часі; ця конкретна проблема не є новою, але обсяг, швидкість і різноманітність потоків даних, які сьогодні збираються датчиками, є особливостями, які заважають процесу прийняття рішень. Наприклад, вимірювання за допомогою різних датчиків, таких як GPS-навігатори або мобільні телефони, дають можливість аналізувати використання транспортної інфраструктури клієнтами в точному часовому масштабі та просторовій роздільній здатності. Традиційно менш вивчені види

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту транспорту (наприклад, схеми спільного використання велосипедів), де надмірна кількість даних призвела до можливості включення специфічних особливостей цих видів транспорту, випустили нову технологічну можливість, яку можна зафіксувати та обробити для отримання збагаченого знання.

У розумних містах розробка та розгортання подібних інтелектуальних систем є складною вимогою, враховуючи співіснування та різноманітність все більшої кількості транспортних засобів у міському середовищі. Обмін ідеями про те, як люди переміщуються в містах, через міста та між ними для вирішення проблем мобільності з точки зору суспільства в цілому, безсумнівно, вимагатиме, щоб транспорт працював екологічно з міркувань безпеки, практичності та безперервності. З цією метою загальна мета, яку переслідують технології покращеної мобільності в розумних містах, полягає в досягненні кращої інтеграції та управління міським транспортом, а також розвитку інтелектуальних систем для безперебійної мультимодальної мобільності (від дверей до дверей). У зв'язку з цим концепція «розумної мобільності» стосується оптимізації транспортних послуг шляхом вирішення проблем, що потребують великих даних, таких як мінімізація заторів на дорогах і оцінка впливу транспортних рішень на навколишнє середовище [3].

На жаль, цілі оптимізації, що характеризують вищезазначені проблеми, часто суперечать одна одній, створюючи компроміси за Парето, у яких спосіб транспортування, обраний людиною з власних причин, не обов'язково є найкращим вибором для всього суспільства [1]. Саме на цьому етапі інструменти та аналітика з використанням машинного навчання вважаються необхідними для обробки величезних обсягів нової інформації та підтримки методів оптимізації, які мають справу з кількома суперечливими критеріями, перетворюючи їх у практичні знання у формі передових інтелектуальних послуг мобільності.

Рисунок 1.1 схематично ілюструє перетин трьох із зазначених вище технологій, які останнім часом стали основними компонентами інформаційної сфери у контексті транспорту та мобільності: IoT, хмарні обчислення та великі дані.

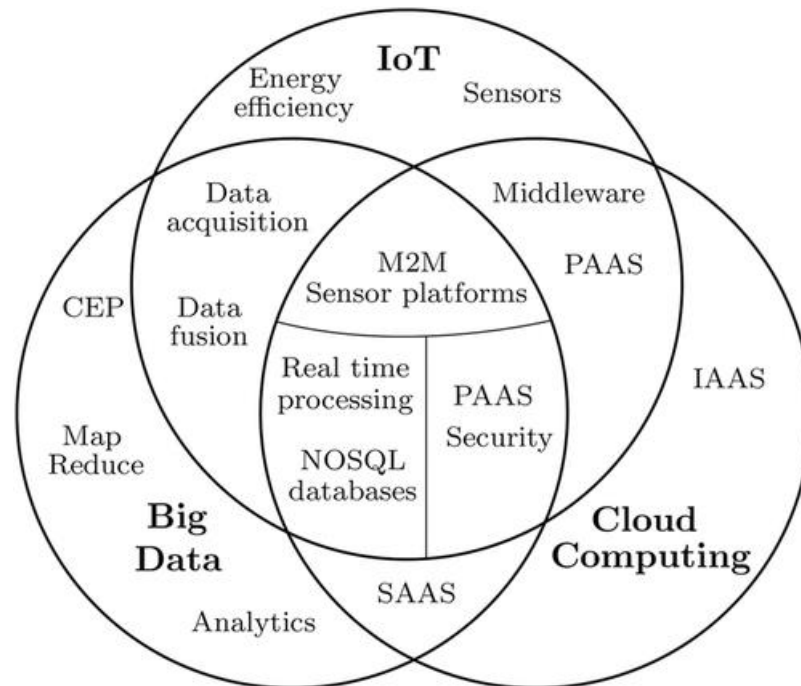


Рисунок 1.1 – Схематична діаграма, що показує перетин між парадигмами IoT, хмарні обчислення та великі дані

Джерело: [19].

У наведеному нижче списку детально описуються три найважливіші показники, які відрізняють Big Data від «звичайних» даних [10]:

1. **Обсяг:** чиста кількість зібраних, керованих і проаналізованих даних.
2. **Швидкість:** ця функція може стосуватися двох понять: (i) швидкість, з якою створюються екземпляри даних; та (ii) швидкість, з якою такі зразки даних отримані та оброблені (пропускна здатність). Наразі досліджувані сценарії ІТС вимагають затримки обробки в реальному часі порядку наносекунд, що є суворим обмеженням, яке передбачає важливі конструктивні рішення в обчислювальній інфраструктурі та розроблені моделі для аналізу даних.
3. **Різноманітність:** неоднорідність отриманих даних є результатом різноманітності оцифрованих доменів, пов'язаних із наявною програмою чи

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту послугою. Потреба в їх спільній обробці та аналізі також має глибокі наслідки з точки зору проєктування систем, щоб включити інтеграцію даних і функції злиття.

Наведений вище опис особливостей даних не обмежується жодною сферою інтересів, однак дослідження в [26] є хорошою відправною точкою для конкретизації їх у сферах, які розглядаються в цій статті: транспортування та мобільність.

Обсяг і охоплення отриманих транспортних даних зростає завдяки масовому розгортанню датчиків у транспортних засобах, мобільних телефонах та інших пристроях. У великих мегаполісах датчики в даний час розгортаються на дорогах, які раніше не були оснащені приладами і навіть не перевірялися візуально [29, с. 35-44]. Подібним чином, щільність сенсорних сіток на вже виявлених проїжджих частинах стає більш гранульованою, щоб краще характеризувати динаміку моніторингу руху, стан тротуарів та інші додатки, що представляють інтерес [5]. Менші спільноти з обмеженим або нульовим збором даних у минулому поступово розгортаються датчики та впроваджуються механізми збору даних, наприклад краудсорсинг [17]. Загалом, результатом у всіх цих випадках є помітне збільшення обсягу даних внаслідок більшої кількості датчиків.

У сфері транспортування та мобільності можна виділити три основні джерела великих даних: соціальні медіа (неструктуровані), дані датчиків (неструктуровані або напів-структуровані) і відкриті дані (неструктуровані – необроблений текст напів-структуровані – JSON/XML або семантично структуровані – пов’язані відкриті дані).

Соціальні медіа: сучасні платформи соціальних медіа зберігають величезні обсяги даних, які можуть виявити критичну інформацію після їх остаточного аналізу. Основна причина полягає в тому, що роль користувача змінилася від простого споживача до постачальника контенту. Соціальні медіа визначено Капланом та ін. як групу Інтернет-додатків, які будуються на ідеологічних і технологічних засадах Web 2.0 і дозволяють створювати та обмінюватися

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту контентом, створеним користувачами. Таким чином, соціальні медіа можна розглядати як контекстне багате джерело великих даних, для чого їх зазвичай називають соціальними великими даними. У контексті транспорту та мобільності це важливе джерело даних через те, що ця зібрана інформація зазвичай позначається геотегами, що дозволяє зробити висновок про геолокалізовані знання про мобільність людей, які створюють сліди даних. Іншими джерелами даних, які підпадають під цю категорію, є спільні програми та ініціативи, такі як краудсорсинг, тобто практика отримання інформації або введення в завдання чи проєкт шляхом залучення послуг великої кількості людей. Наприклад, робота в [20] ілюструє корисність краудсорсингових даних для управління мобільністю.

Щодо даних датчиків, поширення фізичних пристроїв із вбудованою електронікою, програмним забезпеченням, датчиками та приводами, викликане парадигмою Інтернету речей, різко прискорює темпи інновацій у транспортній галузі, особливо в тому, що стосується таких концепцій, як підключений автомобіль (також згадується як як автомобіль як датчик), що дозволяє обмінюватися отриманою інформацією не лише між транспортними засобами, але й між ними та інфраструктурою (наприклад, V2V, V2I, V2X). Крім того, очікується, що ланцюги постачання та логістичні операції отримають значну вигоду від нових способів доставки «останньої милі» за допомогою нових підключених засобів, таких як дрони або автономні транспортні засоби [13].

Відкриті дані: під «відкритими» мається на увазі, що будь-хто може вільно отримувати доступ, використовувати, змінювати та ділитися даними з будь-якою метою. Відкриті транспортні дані потенційно приносять користь як громадянам, так і державному сектору. Останнім часом все більше країн використовують цю можливість, публікуючи сховища даних про національні послуги громадського транспорту, але не про рівень завантаженості транзитної системи. Серед них варто згадати випадок Сполученого Королівства, яке однією з перших країн запровадило Національне сховище даних громадського транспорту [11]. Кожна поїздка



Кафедра інтелектуальних інформаційних систем

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту автобусом, потягом і міжміським автобусом, яка відбувалася щотижня по всій країні реєструється у цьому сховищі для аналізу того, в яких частинах країни автобусне сполучення відстає, серед інших другорядних цілей.

У зв'язку з цим спільнота стала свідком кількох інноваційних методів збору даних у контексті автобусного транспорту: для початку Чжао та ін. в [30] представили новий підхід, здатний збирати дані про швидкість, щільність і об'єм для коридорів безперервного потоку, що представляє прямий інтерес для моніторингу великих даних параметрів руху на великій території за допомогою одного датчика. У цьому ж напрямі [12] пропонує точний і надійний метод збору даних про траєкторію транспортного засобу на основі даних зображень, отриманих безпілотним літальним апаратом.

Іншим додатковим джерелом, з якого можна генерувати реалістичні дані про транспорт і мобільність, є використання симуляторів руху та платформ симуляції великих даних для великих територій, таких як система, представлена в [8]: реалістична імітаційна модель для транспортної мережі Торонто (Канада) була створено на основі даних, зібраних із кількох джерел у цьому регіоні, щоб сприяти проектуванню та розробці інтелектуальних додатків транспорту та мобільності в цьому місті.

## **1.2. Дослідження наявних вебресурсів та застосунків для покращення комфорту пасажирів автобусних перевезень**

Наша мета в цьому розділі – оцінити ефективність прикладів, подібних до нашого дослідження, які демонструють, як підходи на основі даних можуть бути використані для систем автобусних перевезень, прогнозування часу отримання та оптимізації розкладок. Громадський транспорт є важливою складовою з'єднаної та розумної громади. Тому громадяни потребують інформацію в режимі реального часу про прибуття та відправлення транспортних засобів. У багатьох містах світу впроваджуються інтелектуальні транспортні системи з послугами, які реагують на

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту попит, щоб подолати розрив між громадським транспортом і приватними автомобілями. У деяких ранніх дослідженнях аналітика даних використовувалася для оптимізації розкладу руху громадських автобусів та мінімізації часу очікування пасажирів.

Для прогнозування часу перебування автобусів можна використовувати різні технології, які можуть генерувати дані в реальному часі. Серед них найпопулярнішими є системи глобального позиціонування (GPS), автоматичні системи обліку пасажирів (APCS) та рішення Crowdsensing, в яких користувачі співпрацюють із системою через мобільний додаток [28].

Проблема прогнозування часу прибуття автобусів вивчалася шляхом розгляду різних моделей та різних важливих факторів. Основними факторами, які впливають на час прибуття автобуса, є джерело, пункт призначення, координати розташування автобуса, щільність руху, відстань між зупинками, робочий день тощо.

В дослідженні Р. Саджі та інших час прибуття автобусів прогнозували за допомогою моделі опорно-векторної регресії (SVR). У цьому дослідженні були використані дані Petaling Jaya City Bus, включаючи послідовність автобусних станцій, назви автобусних станцій, координати автобусних станцій, часові мітки та відстань, пройдену від попередньої станції до наступної. Вони також реалізували свою модель прогнозування з погодними даними і без них і показали, що додавання погодних параметрів до їхнього набору даних показує незначну різницю в помилці прогнозування [24].

Також у проєкті Пішгу була представлена система підтримки рішень у сфері громадського транспорту для короткострокового та довгострокового прогнозування часу будівництва автобусів. У цьому дослідженні були використані реальні історичні дані двох маршрутів автобусної системи Нешвілла. Підхід до цього дослідження поєднував кластерний аналіз і фільтри Калмана з моделлю спільного сегмента маршруту для отримання більш точних прогнозів часу, і на

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту основі їх результатів, відповідно до базової моделі прогнозування часу, їх система зменшила помилки прогнозування часу в середньому на 25% при прогнозуванні затримки прибуття на рік вперед і на 47% при прогнозуванні в межах 15-хвилинного майбутнього вікна [32].

Згідно з останніми дослідженнями, динамічні прикладні системи на основі даних (DDDAS), які набирають датчики в режимі реального часу та підтримують систему прийняття рішень на основі даних, можуть забезпечити онлайн-навчання моделей та аналітики в різних часових масштабах для підвищення інтелектуальних систем. У рамках свого дослідження автори проаналізували онлайн-систему прогнозування прибуття автобусів у Нешвіллі, використовуючи історичні та поточкові дані в реальному часі, які можуть бути упаковані у вигляді модульних, розподілених і стійких мікросервісів. Послуга довгострокового аналізу затримок включає шум від викидів в історичні дані, щоб виявити закономірності затримок, пов'язаних зі стандартними роками, днями та сезонами для конкретних часових точок і сегментів маршруту. Міські планувальники можуть використовувати дані зворотного зв'язку, згенеровані цими аналітичними сервісами, для покращення розкладу руху автобусів та підвищення рівня задоволеності пасажирів.

Аналіз веб-ресурсів і додатків, спрямованих на підвищення комфорту пасажирів автобусів також виявляє багатогранний ландшафт цифрових інструментів, призначених для вирішення різних проблем, пов'язаних з автобусними поїздками. Ці рішення охоплюють цілий ряд функціональних можливостей, кожна з яких сприяє покращенню загального досвіду для пасажирів.

Додатки для відстеження в реальному часі є одним з найпопулярніших рішень, що надає користувачам своєчасну інформацію про місцезнаходження автобусів, очікуваний час прибуття та будь-які відхилення від запланованих маршрутів. Пропонуючи прозорість і передбачуваність, ці додатки дають можливість пасажирам приймати обґрунтовані рішення щодо своїх планів

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту подорожей, тим самим зменшуючи невизначеність і мінімізуючи незручності, спричинені затримками або збоями в роботі.

Платформи планування подорожей пропонують ще одну цінну функцію, надаючи пасажиром комплексні варіанти маршрутів, пристосовані до їхніх уподобань і вимог. Ці платформи виходять за рамки базової карти маршруту, враховуючи такі фактори, як час у дорозі, варіанти пересадок і міркування доступності, щоб оптимізувати загальний досвід подорожі. Сприяючи ефективному вибору маршруту, ці додатки допомагають пасажиром мінімізувати час у дорозі та уникати перевантажених або ненадійних маршрутів, тим самим підвищуючи комфорт.

Індикатори наявності вільних місць є новим доповненням до набору функцій, пропонує деякими додатками для комфорту пасажирів автобусів. Використовуючи дані в режимі реального часу та інформацію від пасажирів, ці індикатори дозволяють користувачам визначати автобуси з вільними сидячими місцями, що дає їм змогу зробити більш комфортний вибір місця та уникнути переповнених транспортних засобів. Ця функція не лише покращує фізичний комфорт пасажирів, але й сприяє більш приємному та безстресовому досвіду подорожей. Крім того, інтеграція механізмів зворотного зв'язку в ці додатки сприяє налагодженню комунікації між пасажиром та транспортними органами. Надаючи пасажиром можливість повідомляти про проблеми, надавати пропозиції та висловлювати свої зауваження, ці механізми дозволяють транспортним агентствам своєчасно вирішувати проблеми та покращувати якість послуг. Цей ітеративний цикл зворотного зв'язку не лише підвищує комфорт поточних пасажирів, але й сприяє постійному вдосконаленню систем громадського транспорту.

Як приклад, огляд веб-ресурсів для покращення комфорту пасажирів автобусів

1. Moovit – це найкращий додаток для відстеження в режимі реального часу, створений для оптимізації подорожей автобусами. Об'єднуючи дані від

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту транспортних агентств, Moovit надає пасажирам точні прогнози щодо часу прибуття автобусів та оповіщає про перебої в роботі. Інтуїтивно зрозумілий інтерфейс та оновлення інформації про місцезнаходження автобусів у режимі реального часу підвищують довіру пасажирів та скорочують час очікування, що в кінцевому підсумку сприяє підвищенню комфорту під час автобусних поїздок.

2. Transit пропонує комплексне рішення для відстеження в режимі реального часу, доступне в багатьох містах світу. Завдяки таким функціям, як оновлення інформації про місцезнаходження автобусів у реальному часі та рекомендації щодо маршрутів, Transit дає можливість пасажирам легко орієнтуватися в мережах громадського транспорту. Зручний інтерфейс та достовірне надання інформації роблять його цінним інструментом для підвищення загальної надійності та зручності автобусних поїздок.

3. Google Maps революціонує планування поїздок завдяки інтеграції інформації про громадський транспорт у свій картографічний інтерфейс. Надаючи детальні варіанти маршрутів, розклади та приблизний час у дорозі, Google Maps спрощує планування поїздок для пасажирів автобусів. Доступність і широке використання роблять його першочерговою платформою для пасажирів, які шукають ефективні навігаційні рішення, тим самим підвищуючи комфорт автобусних поїздок.

4. Citymapper пропонує мультимодальний досвід планування поїздок, поєднуючи автобуси, метро та інші види транспорту для оптимізації маршрутів. Надаючи безперебійні навігаційні рішення, Citymapper мінімізує час у дорозі та спрощує складні транзитні мережі для пасажирів. Зручний інтерфейс та широкий вибір маршрутів підвищують загальний комфорт та зручність автобусних поїздок у міських умовах.

Веб-ресурси та додатки також сприяють комунікації між пасажирами та транспортними органами за допомогою механізмів зворотного зв'язку. Такі платформи, як Moovit та веб-сайти транспортних агентств, дозволяють пасажирам

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту надавати відгуки про свій досвід перевезень, що дає змогу транспортним органам визначати сфери для вдосконалення. Сприяючи відкритим каналам комунікації, ці платформи дають можливість пасажирам зробити свій внесок у покращення автобусних послуг, що в кінцевому підсумку підвищує рівень комфорту та задоволеності.

Отже, веб-ресурси відіграють ключову роль у підвищенні комфорту пасажирів автобусів, надаючи інформацію в режимі реального часу, сприяючи ефективному плануванню поїздок і забезпечуючи механізми зворотного зв'язку з пасажирами. Від додатків для відстеження в режимі реального часу до платформ планування поїздок і каналів зворотного зв'язку – ці веб-інструменти пропонують практичні рішення для вирішення поширених проблем, пов'язаних з автобусними перевезеннями. Оскільки технології продовжують розвиватися, подальше вдосконалення функціональності та доступності цих веб-ресурсів сприятиме більш комфортному та зручному користуванню автобусами для пасажирів.

Цей аналіз також підкреслює потенціал веб-ресурсів і додатків для значного підвищення комфорту пасажирів автобусів шляхом надання доступу до інформації в режимі реального часу, оптимізації вибору маршруту та полегшення комунікації між пасажирами і транспортними органами. Однак важливо визнати виклики, пов'язані з цифровою інклюзією та конфіденційністю даних, які можуть вплинути на доступність і впровадження цих технологій. Тим не менш, за умови постійних інвестицій у технології та підходів до дизайну, орієнтованих на користувача, ці цифрові рішення обіцяють докорінно змінити досвід поїздок на автобусах і сприяти стійкій міській мобільності.

### **1.3. Систематичний огляд літератури щодо використання машинного навчання у сфері громадських автобусних перевезень**

У сфері громадського транспорту, в тому числі автобусного, дані, зібрані з пристроїв транспортної мережі, можуть бути використані для моделювання рішень

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту на основі машинного навчання (МН). У цьому контексті МН використовується для побудови рішень для мінімізації проблем, пов'язаних з автобусними перевезеннями, підтримуючи розвиток інтелектуальних транспортних систем (ІТС). Наприклад, прогноз часу в дорозі може надати інформацію про тривалість автобусної поїздки, що дозволяє краще планувати поїздки. З іншого боку, рішення для прогнозування пасажиропотоку та підрахунку пасажирів можуть надати інформацію, яка допоможе компаніям в оперативному плануванні. Ці рішення намагаються покращити такі елементи транспортної послуги, як доступність інформації про час у дорозі, надійність і регулярність обслуговування, а також інші, що впливають на сприйняття користувачами при вимірюванні задоволеності наданою послугою.

Огляд різних рішень для громадських автобусних перевезень, запропонованих у літературі з використанням МН, а також того, як вони були змодельовані (наприклад, типи даних і методи МН), може розкрити можливості для науковців з даних і фахівців у галузі транспорту, а також висвітлити проблеми, які були малодослідженими. Крім того, аналітики даних, які тільки починають розробляти рішення для громадського транспорту, можуть отримати користь від картографічної інформації про цю сферу, наприклад, про пристрої, доступні в транспортній мережі, типи даних, які вони генерують, і які типи даних можуть бути використані для побудови рішення.

Для проведення дослідження ми використали методологію систематичного огляду літератури (СОЛ). Ми зібрали з літератури кілька проблем, на які вказують різні роботи, такі як відсутність більш реалістичної інформації для пасажирів, неадекватне операційне планування транспортних компаній, скупчення автобусів і так далі. Ми об'єднали ці проблеми в чотири основні групи: «процедура проведення конкурсу», «оперативне планування», «оперативний контроль» та «пасажирський попит». Ми класифікували ці рішення на дві основні групи: найбільш повторювані рішення, такі як прогнозування часу в дорозі і прогнозування пасажиропотоку і

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту менш повторювані, для яких було знайдено лише одну роботу. Щоб поглибити аналіз використання алгоритмів МН, ми також представили огляд джерел і типів даних, використаних при моделюванні цих розв'язків, і детально описали алгоритми і методи МН, що використовувалися для побудови цих рішень.

Як критерій включення розглядалися статті з журналів або конференцій, в яких пропонувалися рішення на основі МН для задач, пов'язаних з автобусними перевезеннями. На основі цих критеріїв були розглянуті всі роботи, в яких методи МН були центральною частиною пропозиції. Критеріями виключення були: роботи, які не стосувалися проблем міської мобільності в контексті автобусних перевезень; роботи, в яких запропоноване рішення не використовувало методи МН як основну частину рішення; роботи, в яких розглядалися тимчасові рішення для автобусних перевезень під час пандемії SARS-COVID-19; роботи, написані російською мовою; і роботи, недоступні в повному форматі.

Під час пошуку розглядалися лише роботи, опубліковані в період з 2020 по 2021 рік. Загалом було знайдено 549 робіт, які були збережені в Parsifal (інструмент для підтримки планування та проведення систематичних оглядів) [23]. За допомогою цього інструменту 203 роботи було видалено через дублювання. Згодом 192 роботи, що залишилися, були проаналізовані шляхом читання анотацій, і 154 були відкинуті відповідно до критеріїв включення і виключення цього огляду. Відкинуті роботи були об'єднані в чотири групи. До першої групи увійшли роботи, відхилені через те, що вони не мають прямого відношення до міської мобільності, як, наприклад, робота, яка використовувала методи ML і твіти для аналізу безпеки жінок у різних ситуаціях вразливості. Друга група включає роботи, відкинуті для вирішення інших питань, пов'язаних з міською мобільністю, як, наприклад, робота Кім, яка використовувала дані, зібрані за допомогою автоматичних систем збору автобусів, для визначення структури міст [15].

Третя група включає роботи, виключені через те, що вони стосуються проблем, пов'язаних з іншими транспортними системами (наприклад, велосипед,



Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту залізниця). Четверта група включає роботи, в яких розглядалися проблеми автобусних перевезень, але які були відкинуті, тому що в них не використовувалися методи ML або вторинно використовувалися методи ML для розв'язання. Наприклад, [27] представили мобільний додаток для надання пасажиром інформації про маршрут з використанням різних технологій (наприклад, Інтернету речей, Google Maps). Однак для надання цієї інформації пасажиром не було застосовано жодних методів МН. Автори лише вказують, що методи МН можуть бути застосовані до даних, зібраних таким додатком. Нарешті, з 48 робіт, які ми розглянули, вісім були відкинуті, оскільки вони не були доступні в повній версії. Таким чином, для повного ознайомлення було відібрано 40 робіт, і результати представлені нижче.

Транспортні компанії можуть укладати контракти на надання транспортних послуг у місті шляхом проведення тендерів. У цих процесах державні органи можуть використовувати резервну ціну як стратегію для того, щоб більш ефективні компанії брали участь у тендері, а інші компанії, які вважаються менш ефективними, були заблоковані. Однак визначення відповідної резервної ціни не є тривіальним завданням. В літературі Авеналі та інші запропонували дві моделі МН для прогнозування собівартості одиниці послуги громадського транспорту з урахуванням основних характеристик послуги, що надається. Ці моделі можуть бути цінним інструментом для державних установ для підтвердження резервних цін у тендерних процедурах [18].

Іншою проблемою, яка розглядалася в літературі, було прогнозування невиконання контрактів на надання послуг між державними та приватними підприємствами. Ці контракти мають на меті забезпечити кращі послуги громадського транспорту. Однак, контракти на надання послуг мають ризик невиконання, який, якщо його виявити на початковій стадії планування, може полегшити прийняття рішень державними організаціями. У цьому сценарії Ванг та інші зосередилися на розумінні факторів, які включають невиконання контракту.

Вони побудували модель МН для прогнозування кінцевого статусу контракту (успіх чи провал) [18].

Інша сфера досліджень – це оперативне планування, яке складається з чотирьох основних завдань у громадському автобусному транспорті, які виконуються послідовно: розробка маршруту, визначення розкладу руху, планування руху транспортних засобів та призначення екіпажу [21]. Розробка маршрутів включає в себе визначення ліній, маршрутів та зупиночних пунктів. При визначенні розкладу руху програмується час прибуття та відправлення автобусів на зупинках. Нарешті, в задачах планування транспортних засобів та призначення екіпажів, відповідно, визначаються транспортні засоби для кожного рейсу та водії, які будуть керувати цими транспортними засобами.

У літературі рішення зосереджені на визначенні часу в дорозі, складанні розкладу руху транспортних засобів та призначенні водіїв. У цих задачах розклад руху базується на кількості пасажирів, що здійснюють посадку на зупинках, часу в дорозі, що витрачається на кожну поїздку, та частоті, встановленій для надання послуг. Доступність цієї інформації для транспортних компаній є критично важливим фактором для того, щоб пропозиція послуг була ближчою до попиту.

Деякі рішення зосереджені на наданні інформації, яка допомагає транспортним компаніям в оперативному плануванні. Рішення для підрахунку пасажирів – прогнозування кількості пасажирів, які сіли на зупинках, і прогнозування пасажиропотоку або попиту – прогнозування кількості пасажирів, які сядуть у певному місці і в певний час – надають транспортним компаніям інформацію для розуміння мінливості пасажирського попиту (наприклад, часовий діапазон з вищим попитом, робочі дні та святкові дні).

На основі інформації, отриманої в результаті прогнозування пасажиропотоків або підрахунку пасажирів, можна дослідити, чи відповідає частота руху автобусів на певній лінії кількості пасажирів, які користуються цією послугою щодня. Транспортні компанії можуть скоригувати пропоновану послугу

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту (наприклад, зменшити або збільшити частоту руху, запровадити підкріплення), якщо це необхідно. Завдяки прогнозуванню часу в дорозі транспортні компанії можуть, за необхідності, скоригувати розклад руху, спостерігаючи за варіативністю часу, необхідного для проходження маршруту. Крім того, ця інформація може допомогти у програмуванні автопарку, призначенні екіпажів або діях з реструктуризації розроблених маршрутів.

В інших рішеннях застосовуються методи кластеризації в базах даних, зібраних з пристроїв транспортної мережі, для пошуку інформації для оперативного планування. Мендес-Морейра та інші застосували методи кластеризації, використовуючи дані про місцезнаходження автобусів, щоб зрозуміти, чи відповідають графіки обслуговування (наприклад, будні та свята) досліджуваній транспортній мережі [31].

Іншим рішенням, яке може сприяти оперативному плануванню, є виявлення зупинок на маршруті та інформація про час, проведений на цих зупинках. Ця інформація може допомогти у прогнозуванні часу в дорозі, особливо в таких містах, як Ахмедабад, Індія, де на час у дорозі може вплинути неврахування зупинок на нефіксованих пунктах. Нарешті, останнє рішення зосереджене на розробці стратегій для зниження експлуатаційних витрат електробусів, що все ще залишається проблемою для транспортних компаній. У цьому сенсі Нагешаро та інші запропонували модель прогнозування попиту на електроенергію для розробки стратегій електричної зарядки і, таким чином, зниження витрат [16].

GPS, автоматизовані системи збору даних (ADCS), цифрові тахографи, камери, BMS і датчики повітря (датчики MH-Z16 і Yes Plus) – це пристрої, які можуть бути доступні в автобусному парку. Транспортні компанії використовують GPS для відстеження транспортних засобів під час роботи. GPS використовує супутникові сигнали для визначення координат місцезнаходження транспортного засобу (широта, довгота). Однак супутникові сигнали можуть бути заблоковані високими будівлями, тунелями та віадукми. До ADCS належать системи AVL,

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту APC та AFC. Ці системи все частіше використовуються в громадському транспорті завдяки оперативним даним, які вони надають транспортним компаніям. AVL – це система відстеження, здатна записувати інформацію про місцезнаходження кожного транспортного засобу в режимі реального часу і передавати її в центр управління. GPS, наземне радіо та «мертвий розрахунок» – це деякі технології визначення місцезнаходження, які можуть бути використані в AVL. Можна комбінувати ці технології локації, наприклад, використовувати GPS з наземним радіо, причому останнє використовується, коли сигнали GPS недоступні [33].

APC має на меті підрахувати кількість пасажирів, які сідають і прибувають на кожну зупинку на маршруті. Ці системи використовуються разом з AVL для створення баз даних для планування послуг та операцій. AFC вже використовується для управління платежами в громадському транспорті. Основним елементом цієї системи є смарт-картки з інтегрованими мікрочіпами, здатні зберігати і обробляти дані при взаємодії користувачів з системою. Смарт-картки збирають дані про грошові транзакції (коли користувач додає кредити) та проїзні транзакції (при посадці та висадці). Дані зазвичай складаються з просторової (наприклад, місцезнаходження зупинок) і часової (наприклад, час початку кожної поїздки) інформації, яка може бути використана для визначення моделей міської мобільності.

Цифровий тахограф автоматично записує характеристики транспортного засобу на карту пам'яті або надсилає їх безпосередньо до хмари [4]. Цей пристрій надає інформацію про швидкість транспортного засобу, пройдено відстань, кількість обертів двигуна за хвилину, витрату палива, норму споживання палива та інші дані. Крім того, можна перевірити, чи відповідає поведінка водія очікуванням (наприклад, дозволена швидкість). Камера може бути встановлена на автобусах, щоб забезпечити запис зображень для моніторингу водіїв та пасажирів. BMS – це система, здатна керувати акумуляторною батареєю електробуса, відображаючи

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту інформацію про стан батареї (наприклад, напругу, струм, температуру, рівень заряду).

Датчики Yes Plus та MH-Z16 можуть використовуватися для вимірювання рівня концентрації чадного та вуглекислого газу в салоні автобусів через підключені до них адаптери. Індуктивні петльові детектори – це пристрої в транспортній мережі (наприклад, на дорогах і світлофорах). Серед типів даних, які можуть обробляти ці пристрої, – середня швидкість, об'єм і заповненість. Через складність збору реальних даних з індуктивних петлевих детекторів, в деяких роботах використовували SUMO для моделювання сценарію та збору даних з цих пристроїв. Крім згаданих вище пристроїв, для збору даних використовувалися й інші джерела. Хейдарі та інші розробили додаток для збору вимірювань з датчиків смартфонів (наприклад, акселерометра, гіроскопа) зареєстрованих користувачів під час подорожей.

Самарас та інші зібрали метеорологічні дані (наприклад, температуру і погоду) за допомогою API, наданого Dark Sky. Агарваль використовував дані, що характеризують територію навколо автобусних зупинок за чотирма типами (житлові, комерційні, культурні та офісні) [7].

У цілому, каталог рішень, знайдених в літературі, є дуже обмеженим і не вирішує низку проблем автобусних перевезень. Що стосується операційного планування, то ми не знайшли рішень для підтримки інших завдань планування (розробка маршрутів, складання розкладу руху транспортних засобів, призначення екіпажів). Тому рішення, орієнтовані на ці завдання, можуть допомогти вирішити такі проблеми, як визначення тимчасових маршрутів для відвідування заходів (наприклад, фестивалів), покриття транспортного обслуговування в регіоні (особливо в районах, де є лише один варіант маршруту), обслуговування автобусного парку та визначення пунктів зупинок.

Щодо даних, які використовуються у моделях, ми помітили, що геолокація використовується для побудови кількох рішень (наприклад, підрахунок пасажирів

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту для отримання інформації для операційного планування, прогнозування пасажиропотоку та виявлення скупчення автобусів), і що цей тип даних можна вважати стратегічним для побудови рішень. Для побудови рішень використовуються також дані, отримані за допомогою пристроїв на автобусах, від найпоширеніших, таких як тахографи і камери, до найдорожчих пристроїв (наприклад, AFC, APC). Однак доступність цих даних може ускладнювати просування досліджень у цій сфері. Як правило, зібрані дані зберігаються транспортними компаніями, які можуть не пропонувати прямих засобів (наприклад, API, портали відкритих даних) для збору даних, що генеруються пристроями.

Збір даних також може бути складним, оскільки передбачає захоплення смартфонів пасажирів. Часто місцезнаходження користувачів та інша більш конфіденційна інформація може бути отримана за допомогою додатків. Для цього пасажир повинні погодитися на встановлення додатків і політику конфіденційності, що може стати проблемою для рішень, які залежать від збору цих даних. У цьому випадку можна дослідити інші типи даних або запропонувати альтернативи для заміни даних користувачів.

Щодо персональних даних, зокрема, у прогнозуванні часу в дорозі, ми помітили лише те, що часові атрибути (наприклад, час відправлення, день тижня) використовувалися при побудові моделей прогнозування. Однак на час у дорозі може впливати кілька факторів (наприклад, погода, аварії, події, роботи та кількість смуг руху), і атрибути, пов'язані з цими факторами, можуть сприяти побудові більш точних моделей. Таким чином, побудова цих моделей може досліджувати інші типи даних, а їх збір та інтеграція є складним завданням.

Ще один момент полягає в тому, що більшість зусиль зосереджено на підготовці даних для побудови рішень, спрямованих на автобусні перевезення. Наприклад, більш складні методи (наприклад, зіставлення з картою) можуть бути використані для побудови цільового атрибуту, якщо він відсутній у зібраній базі

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту даних. Наразі не існує інструменту, який би допоміг аналітикам даних підготувати дані, зібрані з пристроїв транспортної мережі. Якби такий інструмент був доступний, він міг би прискорити процес моделювання рішень.

Що стосується алгоритмів МН, які використовуються для побудови найбільш рекурентних рішень, алгоритми GBDT, SVR та нейронних мереж (MLP, RNN, FNN), як правило, отримують найкращі результати при прогнозуванні часу в дорозі. Алгоритми RF, MLP та SVM показали хороші результати при прогнозуванні попиту на електроенергію. В інших рішеннях неможливо було зробити огляд ефективності алгоритмів, оскільки деякі дослідники використовували гібридні рішення або не проводили порівняльного аналізу ефективності.

### **Висновки до розділу 1**

Встановлено, що інформатизація громадського транспорту активно розвивається, зокрема через впровадження систем моніторингу та відстеження руху автобусів у режимі реального часу. Це сприяє покращенню комфорту пасажирів, забезпечуючи точну та своєчасну інформацію про маршрути і розклади.

Від додатків для відстеження до платформ планування і комунікаційних каналів – ці інструменти розв'язують поширені проблеми автобусних перевезень. Аналіз літератури також показує, що прогнозування часу в дорозі часто використовує часові атрибути, такі як час відправлення і день тижня. Однак, точність моделей можна покращити, враховуючи додаткові фактори, наприклад, погоду, аварії та кількість смуг руху. Збір та інтеграція таких даних є складним завданням.

Підготовка даних для автобусних перевезень потребує вдосконалення. Відсутність інструментів для підготовки даних з транспортних мереж уповільнює процес моделювання. Наявність такого інструменту могла б значно прискорити цей процес.

Алгоритми GBDT, SVR, MLP, RNN та FNN показали найкращі результати в прогнозуванні часу в дорозі, тоді як RF, MLP та SVM ефективні при прогнозуванні

Кафедра інтелектуальних інформаційних систем

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту попиту на електроенергію. Огляд ефективності інших алгоритмів ускладнений через використання гібридних рішень або відсутність порівняльного аналізу.



## **2 АНАЛІЗ МЕТОДІВ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ОПТИМІЗАЦІЇ КОМФОРТУ АВТОБУСНИХ ПЕРЕВЕЗЕНЬ**

### **2.1. Визначення математичних моделей для оптимізації комфорту пасажирів**

Надійне передбачення часу прибуття автобусів набуває все більшої ваги в розробці ефективного планування поїздок у постійно зростаючих дорожніх умовах із збільшенням обчислювальної техніки та засобів зв'язку. Наявність інформації про місцезнаходження транспортного засобу (VLI) як частини AdvancedTravel Information (ATIS) допомагає підвищити точність передбачення часу прибуття. На час прибуття автобуса можуть впливати різні фактори. Такий список можна легко доповнити багатьма факторами, такими як умови руху, погодні умови, кількість пасажирів, сигналізація, дорожньо-транспортні пригоди та час руху.

1. Моделі, засновані на історичних даних. Цей тип моделі прогнозування дає поточний і майбутній час у дорозі автобуса на основі історичного часу в дорозі попередніх поїздок за той самий період часу. Передбачається, що поточні умови руху залишаються незмінними. Williams and Hoel зазначив, що явище, коли умови дорожнього руху слідуєть номінально послідовним щоденним і тижневим моделям, призводить до очікування, що історичні середні умови в певний час і день тижня забезпечать розумний прогноз майбутніх умов в той самий час доби і день тижня. Таким чином, ці моделі є надійними лише тоді, коли структура руху в досліджуваній області є відносно стабільною, наприклад, у сільській місцевості.

2. Використання середнього часу в дорозі. Ці моделі використовують історичний середній час у дорозі безпосередньо або в поєднанні з іншими вхідними даними певним чином, щоб визначити час прибуття автобуса. У більшості досліджень вони були розроблені з метою порівняння. І майже в усіх цих дослідженнях вони були перевершені відповідними запропонованими основними алгоритмами [25]. Проте було також показано, що вони перевершують деякі

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту моделі, наприклад моделі багатолінійної регресії. Ченг та інші розробили статистичну модель очікуваного часу прибуття (ETA) з використанням пояснювальних змінних [6]. Запропонована модель передбачила час прибуття на основі вхідних даних двох категорій: історичних даних за останні кілька днів і робочих умов поточного дня. Операційна стратегія була додатково включена в модель, щоб зменшити ризик того, що завищений час прибуття може призвести до пропуску автобуса. Під час розробки моделі найбільш помітним обмеженням був розмір історичних даних для калібрування моделі. На відміну від транзитних транспортних засобів, шкільні автобуси мають один курсовий маршрут на день, а їхні розклади переглядаються кожного навчального року. Це зумовило необхідність, щоб модель ETA базувалася на методі, застосовному до відносно невеликого обсягу історичних даних. Оскільки автобуси зазвичай курсували за фіксованими маршрутами та зупинками та відповідно до опублікованих розкладів, їхня модель ETA припускала, що час у дорозі між двома зупинками можна пояснити історичними тенденціями часу в дорозі автобуса та іншими незалежними корельованими змінними. Що стосується умов експлуатації, дослідження включало дотримання розкладу та погодні умови, але не враховував час перебування через стабільний попит. Автори оцінили продуктивність моделі, використовуючи дані, зібрані в реальних умовах експлуатації шкільних автобусів, на яких була встановлена система автоматичного визначення місця розташування автомобіля (AVL) на основі глобальної системи позиціонування. Було зазначено, що запропонована модель постійно демонструє нижчі рівні похибки прогнозування, ніж ковзне середнє та регресійний підходи. З операційною стратегією модель забезпечила досить надійну послугу, в якій приблизно 99 часу подорожі також можна поєднати з інформацією про час подорожі в реальному часі для розробки алгоритму динамічного прогнозування часу подорожі.

3. Використання середньої швидкості. Цей тип моделей використовує середню швидкість транспортних засобів на певних ланках для прогнозування часу

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту подорожі. Вони особливо популярні для прогнозування часу подорожі за допомогою даних, зібраних технологією GPS, оскільки відстань, пройдену за посыланнями, можна обчислити за допомогою інформації про місцезнаходження. Зазвичай ці моделі використовують методи зіставлення карт, які можна встановити в програмному забезпеченні Геоінформаційної системи (ГІС) для оцінки розташування транспортного засобу та часу у дорозі. Вейганг та інші розробили модель для оцінки часу прибуття автобуса на автобусні зупинки за допомогою інформації GPS [9]. Модель складалася з основного алгоритму, який обчислює очікуваний час прибуття, і двох додаткових алгоритмів для визначення положення та швидкості автобуса на дорозі. По-перше, маршрут автобуса ділиться на кілька коротких прямих ліній, підмаршрутів, потім ці лінії моделюється як рівняння першого ступеня на площині. Коли GPS обладнання в автобусі передає своє положення, швидкість та іншу пов'язану інформацію в центр управління, малоімовірно, що це положення збігатиметься з будь-якою точкою на прямолінійних графіках. Таким чином, фактичне положення було зіставлено з точкою на графіку, щоб отримати положення автобуса. По-друге, при використанні інформації про швидкість від GPS час прибуття до точки зупинки буде нескінченним, якщо транспортний засіб стоїть на місці. Щоб вирішити цю проблему, вони використали історичну швидкість руху автобуса на ділянці маршруту та поточну швидкість автобуса, отриману з даних GPS. Завдяки вдосконаленому методу та емпіричному калібруванню результати розробленої моделі були визнані задовільними під час впровадження та експерименту. Вони виявили, що середня похибка між результатами, виведеними системою, і фактичним положенням автобуса становить менше 8 та зменшується за рахунок збільшення кількості ліній, що представляють автобусний маршрут. Пізніше така ж модель була розроблена Саном та іншими з невеликими модифікаціями.

Запропонований алгоритм прогнозування об'єднав дані про місцезнаходження в реальному часі від приймачів системи глобального

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту позиціонування із середніми швидкостями окремих сегментів маршруту, враховуючи історичну швидкість подорожі, а також часові та просторові варіації умов руху. Згадуючи, що розрахункова середня швидкість до станції, запропонована Вейгангом та іншими залежатиме головним чином від її історичної середньої швидкості на маршруті, оскільки автобус знаходиться далеко від станції, Сун та інші стверджували, що поточна швидкість автобуса зазвичай важливіша. Фактор, що впливає на те, як швидко автобус проїде відстань до цікавої станції [9]. Їхній алгоритм в основному складався з двох компонентів. Перший компонент складався з моделі відстеження автобусів у реальному часі з метою обробки даних GPS, проєктування їх на електронну карту, а потім отримання відстані до кожної автобусної станції. Другим компонентом була модель прогнозування часу прибуття автобуса, яка використовувалася для оцінки часу до автобусної станції в режимі реального часу на основі вихідних даних першого компонента та інших факторів. Система була реалізована як кінцевий автомат для забезпечення її регулярності, стабільності та міцності у широкому діапазоні умов експлуатації. Було проведено приклад реального автобусного маршруту, щоб оцінити ефективність запропонованої системи з точки зору точності прогнозування. Результати показали, що запропонована система здатна досягти задовільної точності у прогнозуванні часу прибуття автобуса та ідеальної продуктивності прогнозування напрямку руху. Однак було помічено, що їхня модель менш ефективна в годину пік, ніж у години пік. Це пов'язано зі зміною умов дорожнього руху, тому швидкість збільшується зі збільшенням рівня заторів. Продуктивність алгоритму також порівнювали з алгоритмом, запропонованим Вейгангом та іншими щодо точності передбачення [9].

Як правило, моделі на основі історичних даних вимагають великого набору історичних даних, які можуть бути недоступними на практиці, особливо коли структура трафіку суттєво змінюється. Ці моделі не підходять для великих міст, де як час у дорозі, так і час перебування зазнають великих коливань. Їхня точність

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту значною мірою залежить від подібності між моделями трафіку в реальному часі та історичними.

4. Статистичні моделі. Час прибуття автобуса залежить від кількох факторів, включаючи поведінку водія, ширину колії, перехрестя, сигнали тощо. Ці фактори використовуються як незалежні змінні в багатьох дослідженнях. Точність цих методів залежить від того, що всі залежні змінні розпізнаються та включені в модель, що є складною процедурою. Більшість літератури про модель часових рядів і регресійну модель було опубліковано до 1990-х років. Проте останнім часом дослідники працювали над поєднанням цих моделей з іншими

Моделі часових рядів залежать від даних, отриманих з історичних періодів часу, і прогнозують майбутні періоди часу. У цій моделі припускається, що закономірність або суміш шаблонів виникає час від часу, і ці закономірності можуть бути забезпечені математичними функціями, а для цієї мети можна використовувати історичні дані. Моделі часових рядів припускають, що історичні моделі трафіку залишаються незмінними в майбутньому. У моделях часових рядів його точність сильно залежить від функції відповідності між моделями трафіку в реальному часі та історичними. Варіації в історичних даних або зміни у зв'язку між історичними даними та даними в реальному часі можуть значно спричинити неточність результатів прогнозування, і проблема в цих методах зазвичай повертається до короткої затримки часу, якщо модель прогнозування працює в реальному часі. Д'Анджело використовував модель нелінійного часового ряду для прогнозування часу проходження коридору на шосе. Він порівняв два випадки: перша модель використовувала лише дані про швидкість як змінну, тоді як друга модель використовувала дані про швидкість, заповнюваність і обсяг, щоб передбачити час подорожі. Було виявлено, що модель з однією змінною, що використовує швидкість, була кращою, ніж модель прогнозування з багатьма змінними [2].

5. Регресійні моделі. Регресійні моделі прогнозують і пояснюють залежну змінну за допомогою математичної функції, утвореної набором незалежних змінних. На відміну від моделей прогнозування на основі історичних даних, вони здатні задовільно працювати за нестабільних умов руху. Регресійні моделі зазвичай вимірюють одночасний вплив різних незалежних факторів, що впливають на залежну змінну. Патнаїк та інші запропонували набір багатолінійних регресійних моделей для оцінки часу прибуття автобуса, використовуючи дані, зібрані автоматичним лічильником пасажирів (АРС) [22]. Вони використовували відстань, кількість зупинок, час перебування, посадку та висадку пасажирів і дескриптори погоди як незалежні змінні. Вони вказали, що моделі можна використовувати для оцінки часу прибуття автобуса на зупинки. Ставроула та інші також розробили багатолінійні регресійні моделі з використанням різних наборів вхідних даних [14]. Обидва дослідження показали, що регресійні моделі перевершують інші моделі. Одна велика перевага мультилінійної регресійної моделі полягає в тому, що вона показує, які вхідні дані є менш чи більш важливими для прогнозування. Наприклад, Патнаїк та інші виявили, що погода не є важливим вхідним фактором для їх моделі. Загалом, використання регресійної моделі обмежене, оскільки змінні в транспортних системах сильно взаємозалежні.

6. Модель фільтрації Калмана. Фільтри Калмана широко використовуються для прогнозування часу прибуття автобуса. Основною функцією моделі є надання оцінок поточного стану системи. Але модель фільтрації також служить основою для прогнозування майбутніх значень або для покращення оцінок змінних у попередні періоди, тобто вона має здатність фільтрувати шум. Уолл і Дейл представили короткостроковий алгоритм прогнозування часу прибуття транзитних транспортних засобів шляхом поєднання даних AVL у реальному часі з джерелом історичних даних у Сіетлі, Вашингтон [31]. Їх алгоритм складається з двох компонентів: відстеження та прогнозування. Вони використовували модель фільтра Калмана для відстеження місця розташування транспортного засобу та

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту статистичного оцінювання для прогнозування часу прибуття автобуса. Як уже було зазначено вище, модель спиралася на дані про місцезнаходження в реальному часі та історичну статистику часу, що залишився до прибуття. Тобто припускалося, що інші змінні, які, можливо, впливають на час прибуття, неявно включені в статистику. Тому вони явно не розглядали час перебування як незалежну змінну. У літературі згадується, що деякі емпіричні результати показали, що запропонований алгоритм був досить гнучким, щоб працювати в несприятливих умовах. Було виявлено, що вони можуть передбачити час прибуття автобуса з помилкою менше ніж 12 відсотків. Алгоритм був реалізований як веб-додаток, щоб нарешті надати користувачам прогнозований час прибуття.

Однак, через складну природу даних, як правило, окремий метод або алгоритм не досягає жодних надійних і реальних результатів. Проте існує зростаюча тенденція до використання гібридних алгоритмів для підвищення точності прогнозування. З цього огляду можна зробити висновок, що жоден метод не може створити надійні прогнози через характер середовища та суб'єктів. У той час як алгоритми машинного навчання дають кращі результати та стабільність, завдяки застосованим тут методам, вони узагальнюють великий набір даних, що знижує точність у разі коротких періодів часу від фактичного прогнозованого часу. На майбутнє ми пропонуємо модель, який класифікує умови як незліченну кількість станів. Наприклад, поєднання дорожньо-транспортної пригоди з дощовою погодою дасть кращий прогноз, якщо ми обмежимо наш набір даних подібними умовами. У разі недостатньої інформації про умови деякі позитивні дані можуть бути використані найбільш репрезентативним набором даних нейронної мережі. У таких обмежених фрагментах регресія або часовий ряд подальше застосування фільтрів Калмана виключатиме будь-які інші динамічні фактори, підвищуючи при цьому точність аналізу та моделей загалом.

## **2.2. Аналіз методів та інформаційних технологій для реалізації вебресурсу**

Збільшення попиту на ефективні та надійні системи громадського транспорту призвело до розробки веб-сайтів із прогнозуванням руху автобусів, які надають інформацію в режимі реального часу про розклад руху автобусів, місце розташування та час прибуття. Цей аналіз досліджує різні методи та інформаційні технології, необхідні для створення ефективного веб-сайту з прогнозуванням руху автобусів

1. Глобальна система позиціонування (GPS) і автоматичне визначення місця розташування транспортного засобу (AVL). Глобальна система позиціонування (GPS) – це супутникова навігація система, яка надає інформацію про місцезнаходження та час за будь-яких погодних умов, будь-де на Землі або поблизу неї, де є безперешкодна лінія видимості для чотирьох або більше супутників GPS. Спочатку розроблений Міністерством оборони Сполучених Штатів, GPS згодом був прийнятий для багатьох цивільних цілей, зокрема для управління транспортом.

Компоненти GPS:

1. Супутники: щонайменше 24 супутники, що обертаються навколо Землі, кожен з яких транслює унікальний сигнал.
2. Наземні станції керування: вони контролюють супутники, забезпечуючи їх належне функціонування та точність.
3. Приймачі: пристрої, встановлені в шинах, які приймають сигнали від кількох супутників до визначити точне місцезнаходження автомобіля.

Супутники GPS безперервно передають сигнали, що містять інформацію про їх місцезнаходження та точний час відправлення сигналу. GPS-приймач в автобусі приймає сигнали щонайменше з чотирьох супутників. Обчислюючи різницю в часі між моментом надсилання сигналів і моментом їх отримання, приймач може визначити відстань до кожного супутника. Використовуючи відстань від кількох супутників, GPS-приймач обчислює точне положення автобуса (широту, довготу



Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту та висоту) за допомогою процесу, який називається тріангуляцією. Таким чином, GPS надає поточне місцезнаходження автобусів, що є важливим для моніторингу їхнього руху та надання пасажиром оновлень у реальному часі та гарантує, що автобуси слідують за визначеними маршрутами та розкладом. Крім того, GPS надає історичні дані про автобусні маршрути та розклад, корисні для підвищення ефективності обслуговування та планування інфраструктури.

Автоматичне визначення місцезнаходження транспортного засобу (AVL) – це технологія, яка використовує дані GPS для автоматичного визначення та передачі географічного розташування автомобіля. Системи AVL є невід’ємною частиною управління автопарком у громадському транспорті, надаючи дані в реальному часі про положення транспортних засобів і допомагаючи передбачити час прибуття.

Компоненти AVL:

1. GPS-приймач: встановлюється в автобусі для збору даних про місцезнаходження.
2. Бортовий комп’ютер, який обробляє дані GPS і зв’язується з центральною системою.
3. Система зв’язку, яка передає дані про місцезнаходження з автобуса до центрального центру керування за допомогою стільникового, радіо- чи супутникового зв’язку.
4. Центральна система керування, що отримує, обробляє та відображає дані в центральному інтерфейсі карти, часто інтегрованому з програмним забезпеченням для диспетчеризації та керування.

Багато систем громадського транспорту також пропонують API, які надають дані про місцезнаходження та розклад автобусів у реальному часі. API (інтерфейси прикладного програмування), надані транспортними органами, є потужними інструментами, які дозволяють розробникам отримувати доступ у режимі реального часу до даних про системи громадського транспорту, зокрема

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту розташування автобусів, розклади та інше відповідну інформацію. Ці API служать як міст між системами даних органів транспорту та зовнішніми програмами або веб-сайтами.

Крім того, у контексті прогнозування автобусів існує поняття краудсорсінгу, яке передбачає збір даних у реальному часі від користувачів, які надають інформацію про свій досвід громадського транспорту. Ці створені користувачами дані доповнюють офіційні джерела даних, підвищуючи загальну точність і надійність системи. Пасажири повідомляють такі дані, як час прибуття автобусів, затримки та інциденти, використовуючи мобільні програми або веб-платформи. Зібрані дані агрегуються та обробляються для забезпечення точності та відповідності. Петлі зворотного зв'язку дозволяють користувачам побачити вплив їхніх внесків і заохочують до подальшої участі.\*

Бекенд-технології: бекенд-технології є важливими компонентами для розробки веб-сайту з прогнозуванням автобусів, обробляючи важливі функції, такі як реалізація логіки, взаємодія з базою даних і обробка даних у реальному часі. Ці технології охоплюють серверні мови, інфраструктури, системи керування базами даних і методи обробки даних у реальному часі.

Серверні мови: серверні мови відіграють вирішальну роль у обробці логіки та взаємодії з базами даних у мережі. Вони відповідають за обробку запитів від клієнтів, виконання бізнес-логіки та взаємодію з базою даних для отримання або зберігання даних. Наприклад, Node.js, побудований на движку Chrome V8 JavaScript, дозволяє розробникам використовувати JavaScript для сценаріїв на стороні сервера, що робить його популярним вибором для програм реального часу. Python широко використовується для веб-розробки завдяки своїй великій бібліотеці плагінів та простоті інтеграції. Ruby, відомий своєю простотою та продуктивністю, часто використовується у веб-розробці, особливо з фреймворком Ruby on Rails.

Фреймворки своєю чергою надають розробникам структуроване середовище та готові компоненти для оптимізації серверної розробки. Express.js, мінімальний і

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту гнучкий фреймворк веб-додатків Node.js, спрощує створення веб-додатків і API за допомогою таких функцій, як проміжне програмне забезпечення та маршрутизація. Django, високорівневий веб-фреймворк Python, наголошує на швидкій розробці та включає вбудовані інструменти для взаємодії з базою даних, автентифікації та безпеки. Ruby on Rails, платформа веб-додатків на стороні сервера, написана на Ruby, сприяє конвенції над конфігурацією та принципом DRY («не повторюйся»), що дозволяє розробникам створювати веб-додатки швидко й ефективно.

Керування базою даних: системи керування базами даних зберігають і керують ними дані, які використовуються веб-додатками. Реляційні бази даних, такі як MySQL і PostgreSQL, підходять для зберігання структурованих даних, таких як графіки та історичні записи. Вони пропонують такі функції, як відповідність ACID, запити SQL та індекси для ефективного керування даними. Бази даних NoSQL, такі як MongoDB і Cassandra, з іншого боку, розроблені для обробки неструктурованих даних і підтримки масштабованості. MongoDB зберігає дані в гнучких документах, схожих на JSON, тоді як Cassandra – це розподілена база даних NoSQL із високою масштабованістю, придатна для обробки великих обсягів даних на кількох серверах. Обробка даних у реальному часі: обробка даних у реальному часі має вирішальне значення для надання миттєвих оновлень і підтримання швидкодії веб-програми. WebSockets забезпечують двонаправлений зв'язок між клієнтами та серверами, полегшуючи оновлення та взаємодію в реальному часі.

Фреймворки обробки потоків, такі як Apache Kafka та Apache Flink, використовуються для обробки та аналізу потоків даних у реальному часі. Apache Kafka – це розподілена потокова платформа, відома своєю високою пропускнуною спроможністю та відмовостійкістю. Apache Flink забезпечує точну обробку даних і аналітику в реальному часі з такими функціями, як обробка часу подій і обчислення зі збереженням стану.

Таким чином, серверні технології формують основу веб-сайту прогнозування автобусів, що дозволяє реалізувати критичні функції, такі як логічне виконання, керування базами даних і обробка даних у режимі реального часу. Використовуючи серверні мови, фреймворки, системи керування базами даних і методи обробки даних у реальному часі, розробники можуть створювати надійні та ефективні веб-додатки, які надають своєчасну та точну інформацію користувачам.

Хмарні послуги та хостинг: такі хмарні платформи, як AWS (Amazon Web Services), Azure (Microsoft Azure) і Google Cloud, пропонують ряд послуг для розміщення та керування веб-додатками. Ці платформи забезпечують масштабовану інфраструктуру, сховище даних і обчислювальні ресурси, що дозволяє розробникам розгортати та масштабувати програми за потреби без накладних витрат на керування фізичними серверами. Завдяки таким функціям, як автоматичне масштабування та моделі ціноутворення з оплатою за використання, хмарні платформи пропонують гнучкість і економічність для розміщення веб-сайтів із прогнозуванням руху автобусів. Мережі доставки вмісту (CDN) відіграють вирішальну роль у покращенні продуктивності веб-сайтів шляхом географічного розподілу вмісту. Кешуючи вміст на серверах, розташованих ближче до кінцевих користувачів, CDN зменшують затримку та покращують час завантаження сторінки. Це особливо важливо для веб-сайтів із прогнозуванням руху автобусів, де оновлення в режимі реального часу та оперативність є важливими для надання точної інформації користувачам. CDN також допомагають пом'якшити стрибки трафіку та забезпечити стабільну роботу в різних регіонах.

Transit API, такі як Google Maps Transit і NextBus, надають доступ до даних про громадський транспорт, зокрема розклади автобусів, маршрути та інформацію про місцезнаходження в реальному часі. Інтеграція цих API у веб-сайти прогнозування автобусів дозволяє користувачам отримувати доступ до актуальної інформації про прибуття, відправлення та маршрути автобусів. Використовуючи

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту транзитні API, розробники можуть розширити функціональність своїх веб-сайтів і надати користувачам цінну інформацію для ефективного планування подорожей.

Зауваження щодо впровадження: масштабованість є критично важливим фактором для успішного впровадження системи прогнозування автобусів. Оскільки кількість користувачів і точок даних з часом зростає, система повинна мати можливість обробляти збільшений трафік і робоче навантаження без погіршення продуктивності. Масштабованість гарантує, що система залишається чутливою та ефективною навіть у періоди пікового використання. Цього можна досягти шляхом використання масштабованої інфраструктури, такої як хмарні платформи, і проєктування архітектури системи, яка має горизонтальне масштабування, що дозволяє додавати додаткові ресурси за потреби. Надійність є ще одним важливим аспектом реалізації системи. Користувачі покладаються на систему прогнозування руху автобусів, щоб надати точну й актуальну інформацію про розклад руху автобусів, маршрути та прибуття. Забезпечення тривалої роботи та доступності має вирішальне значення для збереження довіри та задоволення користувачів. Це вимагає надійної інфраструктури, заходів із резервування, а також проактивного моніторингу та обслуговування практики для виявлення та вирішення потенційних проблем до того, як вони вплинуть на доступність послуг. Впровадження механізмів резервного копіювання та відновлення після збоїв може ще більше підвищити надійність шляхом мінімізації простоїв у разі збоїв апаратного чи програмного забезпечення. Безпека має першорядне значення при розробці системи прогнозування автобусів, оскільки передбачає обробку конфіденційних даних користувача, таких як інформація про місцезнаходження та платіжні деталі. Захист даних користувача та безпека каналів зв'язку є важливими для запобігання несанкціонованому доступу, витоку даних та іншим загрозам безпеці. Цього можна досягти за допомогою шифрування, механізмів автентифікації, контролю доступу, а також регулярних перевірок і оцінок безпеки для виявлення та усунення вразливостей.

Відповідність галузевим стандартам і нормам, таким як Загальний регламент захисту даних (GDPR), гарантує, що дані користувачів обробляються відповідально та відповідно до правових вимог. Конфіденційність також є критично важливою мірою, особливо під час обробки персональних даних, таких як місцезнаходження користувача. Відповідність нормам захисту даних, таким як GDPR, необхідна для захисту конфіденційності користувачів і забезпечення того, що особисті дані збираються, обробляються та зберігаються прозорим і законним способом. Це включає отримання згоди користувача на збір і обробку даних, надання чіткої та лаконічної політики конфіденційності та впровадження заходів для захисту конфіденційності користувачів, таких як анонімізація та мінімізація даних. Віддаючи пріоритет конфіденційності та дотриманню правил захисту даних, розробники можуть створити довіру користувачів і продемонструвати відданість захисту їхніх прав на конфіденційність.

Таким чином, створення веб-сайту з прогнозуванням автобусів вимагає поєднання передових методів збору даних у реальному часі та прогнозної аналітики, а також надійних інформаційних технологій для розробки інтерфейсу та серверної частини, обробки в режимі реального часу та дизайну взаємодії з користувачем. Ефективно інтегрувавши ці компоненти, розробники можуть надати надійну та ефективну послугу, яка покращить враження від громадського транспорту.

## **Висновок до розділу 2**

Отже, у цьому розділі було визначено та проаналізовано математичні моделі, що застосовуються для оптимізації комфорту пасажирів у автобусних перевезеннях. Зокрема, розглянуто моделі розподілу пасажиропотоку, оптимізації розкладів руху, а також моделі прогнозування попиту на транспортні послуги. Також було проаналізовано сучасні методи та інформаційні технології, які можуть бути використані для розробки вебресурсу з метою покращення комфорту пасажирів автобусних перевезень. Виявлено, що технології веброботки, такі як

Кафедра інтелектуальних інформаційних систем

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту HTML, CSS, JavaScript, а також фреймворки для розробки інтерфейсів користувача (React, Angular) та бекенд-технології (Node.js, Django), дозволяють створити ефективні та зручні у користуванні вебресурси на основі Transit API, такі як Google Maps Transit і NextBus.

Кафедра інтелектуальних інформаційних систем  
Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту

### **3 МОДЕЛЮВАННЯ ТА ТЕХНІЧНЕ ПРОЄКТУВАННЯ ВЕБРЕСУРСУ ДЛЯ ОПТИМІЗАЦІЇ КОМФОРТУ АВТОБУСНИХ ПЕРЕВЕЗЕНЬ**

#### **3.1. Визначення бази даних та алгоритмів обробки інформації про маршрути та розклади**

Набір даних є критичним компонентом кожної системи прогнозування руху автобусного транспорту. Це досліджене використовує UK Open Bus service. Цей набір даних було обрано через його багаті властивості та доступність онлайн. Кожен запис у цьому наборі даних охоплює інформацію у форматі 17 полів:

- Longitude;
- VehicleLocation.Latitud;
- DestinationLong;
- DestinationLat;
- OriginLong, OriginLat;
- RecordedAtTime;
- ArrivalTime;
- ScheduledArrivalTime;
- DistanceFromStop;
- OriginName;
- DestinationName;
- PublishedLineName;
- NextStopPointName;
- ArrivalProximityText;
- VehicleRef;
- DirectionRef.

Перші 6 полів – це поточне місце розташування автобуса, координати місця призначення та відправлення. Опис інших полів наведено нижче.



`RecordedAtTime` – це час на контрольній точці, в якому фіксується поточне відвідування автобуса.

`ArrivalTime` – ця година, коли автобус прибуває на наступну зупинку.

`ScheduledArrivalTime` автоматично береться з опублікованого розкладу руху автобусів і вказує на запланований час прибуття автобуса на наступну зупинку.

`DistanceFromStop` – відстань автобуса від наступної зупинки в момент спостереження.

`OriginName` та `DestinationName` – це місце відправлення та місце призначення автобусів.

`PublishedLineName` вказує, на якій лінії працює автобус.

`NextStopPointName` – назва наступної зупинки.

`ArrivalProximityText` показує поточний статус автобуса у вигляді тексту, включаючи зупинку, наближення та відстань до автобуса.

`VehicleRef` – це ідентифікаційний номер для кожного автобуса, сайт якого відстежується.

Поле `DirectionRef` вказує напрямок вхідної або вихідної шини.

Різні показники можуть вимірювати якість послуг в інфраструктурі громадського транспорту. Вчасне прибуття на зупинку є фактором. Різниця в часі між запланованим і фактичним прибуттям автобусів була обрана як основна причина, через яку люди створюють систему громадського транспорту в багатьох містах (рис. 3.1).

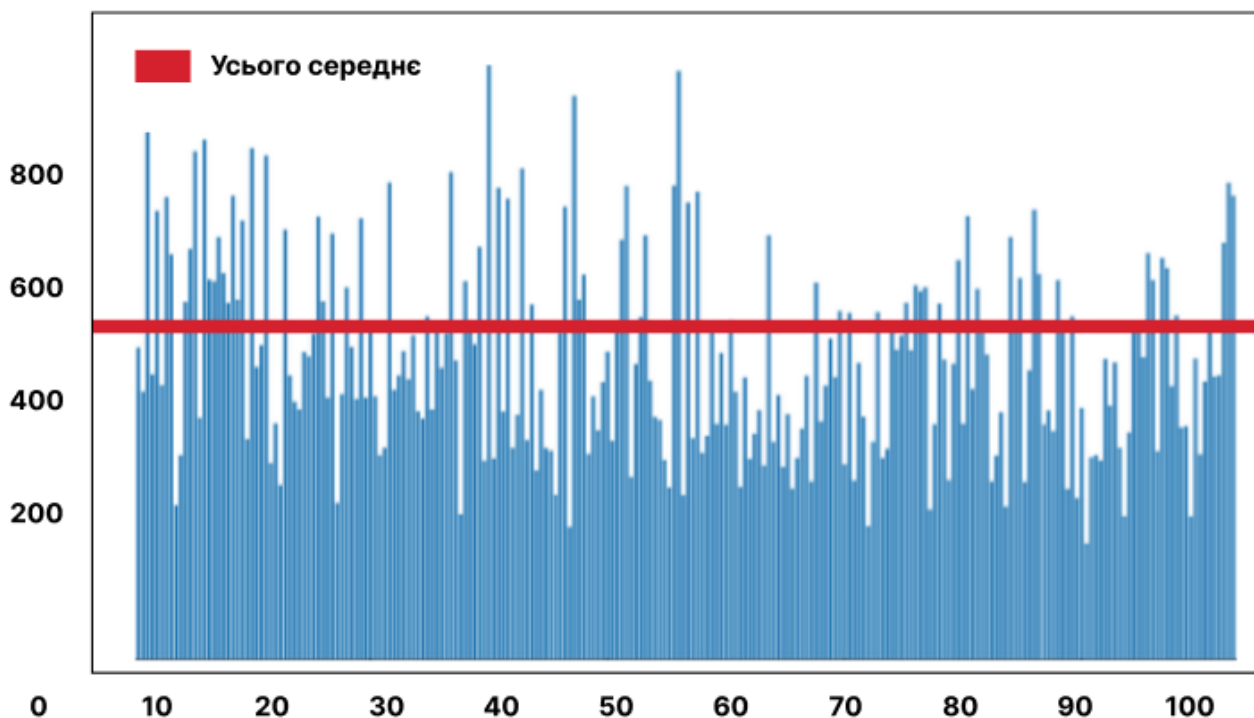


Рисунок 3.1 – Середня підтримка на всіх автобусних лініях

Джерело: [27].

Початковий аналіз записів у наборі даних показує, що середня підтримка на всіх автобусних лініях становить 491 секунду. Отже, спочатку аналізуються дані щодо розбіжностей між запланованим часом прибуття та фактичним часом прибуття. Невідповідність часу – це будь-яка різниця між часом прибуття автобуса за розкладом та фактичним часом прибуття. Середній час запізнення та розбіжності на всіх лініях цього набору даних становлять близько 8 хвилин і 6 хвилин відповідно [27].

Таким чином, на основі наведеної вище бази даних, використання моделей машинного навчання та регресійного аналізу для прогнозування часу прибуття може значно зменшити непередбачуваність та покращити задоволення пасажирів. Більш точне прогнозування часу прибуття дозволить пасажирам планувати свої поїздки з меншою невизначеністю та стресом. використання зазначених алгоритмів

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту дозволить більш точно прогнозувати час прибуття, що підвищить загальну якість послуг громадського транспорту та задоволення пасажирів.

### **3.2. Огляд моделей машинного навчання з метою прогнозування часу руху автобусів**

Для створення моделі машинного навчання, яка здатна прогнозувати час прибуття автобусів на основі визначеної вище бази даних, можна використовувати бібліотеки Python, такі як Pandas для обробки даних та Scikit-learn для побудови моделі.

На першому етапі підготовка даних час у форматі дати й часу (RecordedAtTime, ScheduledArrivalTime, ArrivalTime) необхідно конвертувати до числового формату, що полегшує аналіз. Для цього кожен з цих часових показників перетворюється на кількість секунд від початку доби. Це дозволяє моделі працювати з числовими значеннями, що є важливим для алгоритмів машинного навчання.

Наступним кроком є створення набору ознак (features) та міток (labels). Ознаки включають координати (Longitude, VehicleLocation.Latitude, DestinationLong, DestinationLat, OriginLong, OriginLat) та запланований час прибуття у секундах (ScheduledArrivalTime\_seconds). Міткою виступає фактичний час прибуття у секундах (ArrivalTime\_seconds). Після цього дані розділяються на тренувальний та тестовий набори.

Побудова моделі включає тренування на тренувальних даних та оцінку її на тестових даних.

Найпростіший метод – це використання лінійної регресії з бібліотеки Scikit-learn. Лінійна регресія підходить для цієї задачі, оскільки вона добре працює з числовими даними та є простою у реалізації. Модель навчається на тренувальних даних, що означає, що вона аналізує взаємозв'язки між ознаками та мітками й

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту намагається знайти оптимальні вагові коефіцієнти, які мінімізують різницю між передбаченими та фактичними значеннями.

Після тренування моделі за допомогою алгоритму лінійної регресії на тренувальних даних, стає можливим провести оцінку її точності на тестових даних. Оцінка здійснюється за допомогою середньоквадратичної похибки (MSE), яка обчислює квадратичну відмінність між фактичними та передбаченими значеннями. Чим менше значення MSE, тим краще модель адаптується до нових даних і точніше прогнозує час прибуття автобуса.

Після успішної оцінки моделі вона зберігається у файлі для подальшого використання. Це дозволяє використовувати навчену модель безпосередньо в реальному веб-сервісі для прогнозування часу прибуття.

Інтеграція моделі стає можливим за допомогою Flask – мінімалістичного фреймворку для створення вебдодатків на Python. За допомогою Flask можливо створити API, яке може обробляти POST-запити з даними про автобус і повертати передбачення часу його прибуття.

Під час інтеграції вебресурсу з моделлю машинного навчання, Flask забезпечує легкий доступ до моделі через мінімальну кількість коду і простий механізм маршрутизації запитів. Кожен POST-запит сприймається як дані для прогнозування часу прибуття автобуса. Після отримання вхідних даних модель використовує збережені знання для генерації прогнозу та повертає його у форматі, зрозумілому для клієнта.

Таким чином, подальша інтеграція моделі у вебресурс через Flask відкриває можливості для широкого спектру застосувань, від моніторингу транспорту до створення персоналізованих сервісів для користувачів з різних сфер життя.

Іншим варіант передбачає побудову моделі на основі методу Gradient Boosted Trees (GBT), а саме регресор Gradient Boosted Trees (GBTRegressor) у PySpark, який собою потужний підхід ансамблевого навчання та ґрунтується на деревах рішень для створення високоточних моделей прогнозування. Ця модель особливо

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту ефективна як для класифікації, використовуючи сильні сторони дерев рішень, пом'якшуючи їх слабкі сторони шляхом ітеративного уточнення (Додаток Б).

GBTRegressor вимагає набір даних, відформатований як RDD об'єкти LabeledPoint для задач регресії. Кожна LabeledPoint містить мітку (цільове значення) та вектор ознак, що представляє вхідні ознаки. Для задач класифікації мітки є двійковими значеннями (0 або 1), тоді як для задач регресії мітки є дійсними числами.

При роботі з категоріальними ознаками модель дозволяє вказати артикуляцію категоріальних ознак за допомогою карти. Це має вирішальне значення для правильної інтерпретації категоріальних змінних, особливо коли вони закодовані як цілі числа.

Модель надає декілька гіперпараметрів для налаштування, зокрема:

1. Функція втрат, яка використовується для мінімізації помилок під час навчання.
2. Швидкість навчання, яка контролює розмір кроку на кожній ітерації під час руху до мінімуму функції втрат. Має бути в діапазоні від 0 до 1.
3. Максимальна глибина дерев, що визначає максимальну глибину окремих дерев в ансамблі. Глибші дерева можуть охоплювати складніші патерни, але можуть призвести до надмірної підгонки.
4. Максимальна кількість відсіків – використовується для розбиття ознак, особливо важливо для категорійних ознак з великою кількістю унікальних значень.

Процес навчання передбачає завантаження підготовленого набору даних до GBTRegressor, вказуючи кількість ітерацій (дерев, які потрібно додати до ансамблю). Кожна ітерація додає нове дерево, яке намагається виправити помилки, допущені попередньо навченими деревами. Модель ітеративно покращує свої прогнози до збіжності або до досягнення заданої кількості ітерацій.

Після навчання GBTRegressor може прогнозувати результати для нових точок даних. Прогнози робляться шляхом підсумовування прогнозів всіх дерев в

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту ансамблі, зважених за їхньою важливістю. Продуктивність моделі можна оцінити за допомогою різних метрик, таких як середньоквадратична помилка (RMSE) для задач регресії, яка вимірює різницю між прогнозованими і фактичними результатами.

Отже, приклад використання GBTRegressor у PySpark включає завантаження набору даних, його підготовку шляхом індексування категоріальних ознак, розділення на навчальні та тестові набори, навчання моделі, створення прогнозів та оцінку ефективності моделі. Цей процес підкреслює гнучкість і потужність GBTRegressor в обробці складних наборів даних і створенні точних прогнозних моделей.

### **Висновок до розділу 3**

Набір даних є критичним компонентом систем прогнозування руху автобусного транспорту. У цій роботі використовувався набір даних UK Open Bus service, обраний через його багаті властивості та доступність онлайн. Записи містять 17 полів, включаючи поточне місце розташування автобуса, координати місця призначення та відправлення, час на контрольній точці, запланований і фактичний час прибуття, відстань до наступної зупинки, назви маршрутів та ідентифікаційний номер автобуса.

Використання моделей машинного навчання та регресійного аналізу для прогнозування часу прибуття може значно зменшити непередбачуваність та підвищити задоволення пасажирів. Точніше прогнозування дозволить пасажирам планувати свої поїздки з меншою невизначеністю та стресом, покращуючи загальну якість послуг громадського транспорту.

## **4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА РОЗРОБКА ДОКУМЕНТАЦІЇ ВЕБРЕСУРСУ ДЛЯ ОПТИМІЗАЦІЇ КОМФОРТУ АВТОБУСНИХ ПЕРЕВЕЗЕНЬ**

### **4.1. Вибір мов програмування, фреймворків та інших програмних інструментів**

Завдяки використанню сучасних інструментів, таких як Prefect та Streamlit, наша модель може бути доступним для широкого кола фахівців з різним рівнем кваліфікації. Крім того, проект можна реалізувати безкоштовно за допомогою пробного періоду Google Cloud, що робить його чудовою можливістю для навчання для початківців.

Управління хмарними ресурсами відбувається за допомогою Terraform, оркестрування функцій ETL (у керовані завдання та потоки) відбувається за допомогою Prefect. Блоки Prefect своєю чергою використовуються для обробки автентифікації та взаємодії з ресурсами Google Cloud. А створення образу Docker – для середовища виконання сценаріїв. Безпосередньо візуалізація даних відбувається за допомогою Streamlit.

Нижче наведено детальний опис переваг кожного інструмента:

#### **1. Bus Open Data Service:**

1.1. Безкоштовний доступ до великого обсягу даних про автобусні перевезення у Великобританії.

1.2. Дані оновлюються в реальному часі, що дозволяє отримувати актуальну інформацію.

1.3. Доступ до даних через API, що спрощує інтеграцію та обробку.

#### **2. Black:**

2.1. Автоматичне форматування коду дозволяє зосередитися на написанні коду, а не на його стилістичному оформленні.

2.2. Забезпечує єдиний стиль коду у всьому проєкті, що важливо при спільній роботі.

2.3. Швидке форматування коду завдяки оптимізованим алгоритмам.

3. Flake8:

3.1. Допомогає знайти синтаксичні та стилістичні помилки, що покращує якість коду.

3.2. Можливість налаштування правил відповідно до потреб проєкту.

3.3. Легка інтеграція з іншими інструментами для перевірки коду.

4. pre-commit:

4.1. Автоматичний запуск перевірок перед комітами запобігає внесенню некоректного коду в репозиторій.

4.2. Підтримка різних плагінів для різних перевірок, що розширює функціонал.

4.3. Проста конфігурація за допомогою YAML-файлів.

5. Terraform:

5.1. Дозволяє керувати інфраструктурою за допомогою кодів, що спрощує її відтворюваність та масштабування.

5.2. Підтримка різних хмарних провайдерів, що забезпечує гнучкість вибору інфраструктури.

5.3. Автоматизація процесу створення та налаштування ресурсів, що знижує ймовірність помилок.

6. Prefect:

6.1. Забезпечує зручний спосіб керування ETL-процесами за допомогою завдань та потоків.

6.2. Вбудовані інструменти для моніторингу та відлагодження процесів.

6.3. Підтримка масштабування процесів відповідно до потреб проєкту.

7. Google Cloud Services (GCS):

7.1. Висока надійність та доступність хмарних сервісів від Google.



7.2. Підтримка різних сервісів, таких як обробка даних, зберігання, аналітика та багато іншого

7.3. Легка інтеграція з іншими сервісами та інструментами.

8. Visual Studio Code:

8.1. Інтуїтивно зрозумілий інтерфейс та висока продуктивність.

8.2. Підтримка великої кількості плагінів для розширення функціоналу.

8.3. Вбудовані інструменти для відлагодження, управління версіями тощо.

9. Docker:

9.1. Контейнеризація дозволяє переносити додатки між різними середовищами без змін.

9.2. Забезпечує ізольоване середовище для виконання додатків, що підвищує безпеку та стабільність.

9.3. Підтримка горизонтального масштабування додатків.

10. Streamlit:\*\*

10.1. Легке створення веб-додатків для візуалізації даних за допомогою Python.

10.2. Можливість створення інтерактивних інтерфейсів для користувачів.

10.3. Швидке створення та розгортання додатків без необхідності у великій кількості коду.

Python був обраний як основна мова програмування для цього проєкту з кількох ключових причин. По-перше, Python відзначається простотою та читабельністю, що робить його доступним для новачків та сприяє швидкому розвитку проєктів. Його лаконічний синтаксис полегшує розуміння та супровід коду, що особливо важливо при командній розробці.

По-друге, Python має багату екосистему бібліотек і фреймворків, які охоплюють усі аспекти обробки даних, машинного навчання, веб-розробки та візуалізації. Це включає такі популярні бібліотеки, як pandas і NumPy для обробки даних, TensorFlow і scikit-learn для машинного навчання, Django і Flask для веб-

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту розробки, а також `matplotlib` і `Plotly` для візуалізації. Така різноманітність інструментів дозволяє ефективно реалізувати всі етапи розробки і обробки даних в межах одного проєкту.

Крім того, Python має велику та активну спільноту розробників, яка забезпечує підтримку, регулярне оновлення та вдосконалення мови та її бібліотек. Це створює надійне середовище для розвитку та вирішення технічних проблем, а також доступ до численних навчальних ресурсів і документації.

Ще однією важливою перевагою Python є його здатність легко інтегруватися з іншими інструментами та сервісами, що робить його універсальним вибором для розробки комплексних рішень. Наприклад, Python легко інтегрується з `Google Cloud Services`, `Docker`, `Terraform`, та іншими інструментами, що використовуються у проєкті.

Завдяки цим перевагам Python став оптимальним вибором для розробки і реалізації даного проєкту, забезпечуючи ефективність, масштабованість та простоту в користуванні.

Ці інструменти у комплексі забезпечують ефективну, надійну та масштабовану платформу для обробки даних та надання користувачам актуальної інформації в реальному часі.

#### **4.2. Детальний опис програмного коду та функціональності вебресурсу**

Налаштування надійного середовища розробки має вирішальне значення для успіху будь-якого проєкту. Для цього дослідження використовуємо `Visual Studio Code (VS Code)` як редактор коду. Нижче наведено деякі основні інструменти та бібліотеки, які прискорили процес розробки: `Влас`.

`Black` як форматер коду Python, який забезпечує єдиний стиль коду, що особливо корисно, коли над одним проєктом працює декілька розробників.

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту Використовуючи Black, стає можливим зосередитися на написанні коду, а не на його форматуванні.

Щоб встановити Black, можна використати наступну команду pip: «black {source\_file\_or\_directory}». Або можливо налаштувати Black для автоматичного форматування коду при збереженні безпосередньо у VS Code (рис. 4.1).

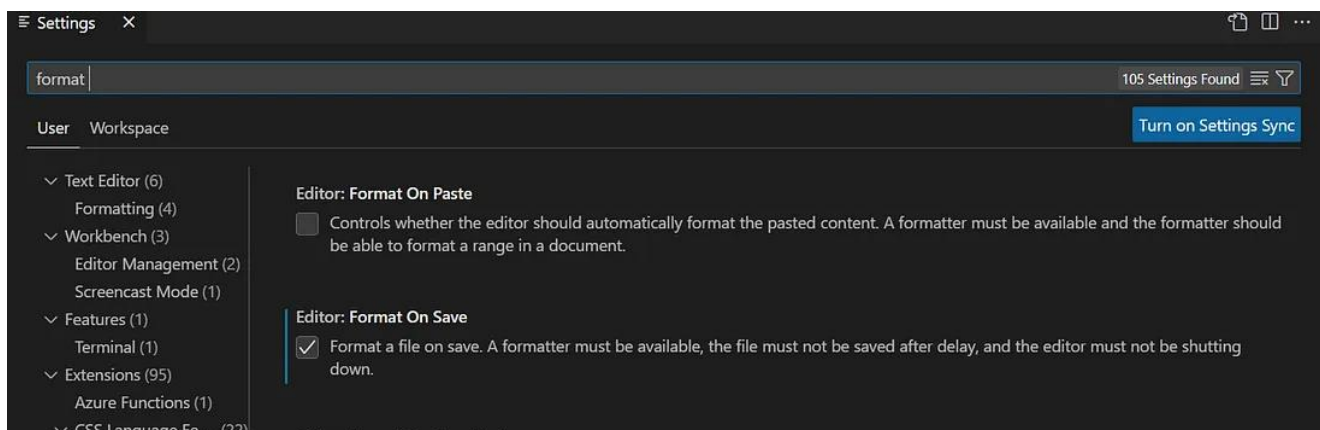


Рисунок 4.1. Встановлення Black у VS Code

Для цього необхідно просто перейти до налаштувань VS Code і позначити пункт «Editor». Після встановлення ми можемо запустити Black на конкретному файлі Python або на цілому каталозі.

Після цього встановлюємо Flake8, літнер коду, який перевіряє наш код на відповідність PEP, керівництва по стилю Python. Він допомагає виявити синтаксичні помилки, невикористані змінні та інші поширені проблеми, які можуть призвести до помилок або невідповідностей у нашому коді.

Між Black та Flake8 можуть виникати конфлікти через різницю у застосуванні стилів. Щоб вирішити ці конфлікти, було створено файл setup.cfg у каталозі проєкту, щоб налаштувати параметри Flake8 для сумісності з Black (рис. 4.2).

```
[flake8]
ignore = E203, E266, E501, W503
max-line-length = 88
max-complexity = 18
select = B,C,E,F,W,T4
exclude =
    venv
    .venv
    .git,
    __pycache__,
    docs/source/conf.py,
    old,
    build,
    dist
```

Рисунок 4.2. setup.cfg файл для Flake8

`max-line-length = 88`: встановлює максимальну довжину рядка у 88 символів, що є значенням за замовчуванням для Black.

`extend-ignore = E203, W503`: ігнорує певні правила PEP, які конфліктують з форматуванням Black. Таким чином, ми зможемо забезпечити впорядкований і послідовний робочий процес розробки.

Далі необхідно розробити фреймворк, який дозволяє запускати скрипти перед фіксацією нашого коду. Якщо скрипт не спрацює, то фіксація не відбудеться. Ми можемо запустити Black і flake8 як хуки `pre-commit`, так що кожного разу, коли код буде зафіксовано, Black автоматично відформатує його, а flake8 перевірить його на відповідність вимогам. Якщо flake8 не спрацює, то у фіксації буде відмовлено.

Після встановлення `pre-commit` також потрібно буде визначити наш конфігураційний файл `pre-commit`. Для цього створюємо `.pre-commit-config.yaml`, який визначить джерела, звідки будуть братися хуки.

```

repos:
- repo: https://github.com/psf/black
  rev: stable
  hooks:
  - id: black
    language_version: python3.6
- repo: https://gitlab.com/pycqa/flake8
  rev: 3.7.9
  hooks:
  - id: flake8
  
```

Рисунок 4.3. Конфігураційний файл pre-commit.yaml

Далі ми створюємо файл requirements.in, який містить лише ваші прямі залежності проєкту, а потім виконуєте `pip-compile .\requirements.in` буде згенеровано файл requirements.txt, який міститиме модулі, перелічені у нашому файлі requirements.in, а також транзитивні залежності цих модулів.

Використання `pip-compile` з `pip-tools` є корисним при створенні файлу requirements.txt. Це забезпечить передбачуваність та детермінованість збірки.

Подальше налаштування та розгортання проєкту передбачає створення облікового запису Google Cloud. Для цього проєкту ми будемо використовувати наступні сервіси Google Cloud:

1. Artifact Registry для зберігання нашого образу Docker.
2. Cloud Run для запуску наших скриптів на Python
3. Compute Engine для запуску нашого агента Prefect.
4. Хмарне сховище для зберігання наших даних
5. BigQuery для нашого без серверного сховища даних

Спочатку створюємо новий проєкт з унікальним ідентифікатором проєкту (рис. 4.4).

```
export GOOGLE_APPLICATION_CREDENTIALS="your_key.json"
```

Рисунок 4.3. Створення нового проєкту Google Cloud

У цьому ж проєкті створюється новий обліковий запис IAM. Обліковий запис можна використовувати для керування доступом до сервісів Google Cloud.

У новому обліковому записі служби натискаємо на «Actions», а потім «Manage keys» (рис. 4.5).

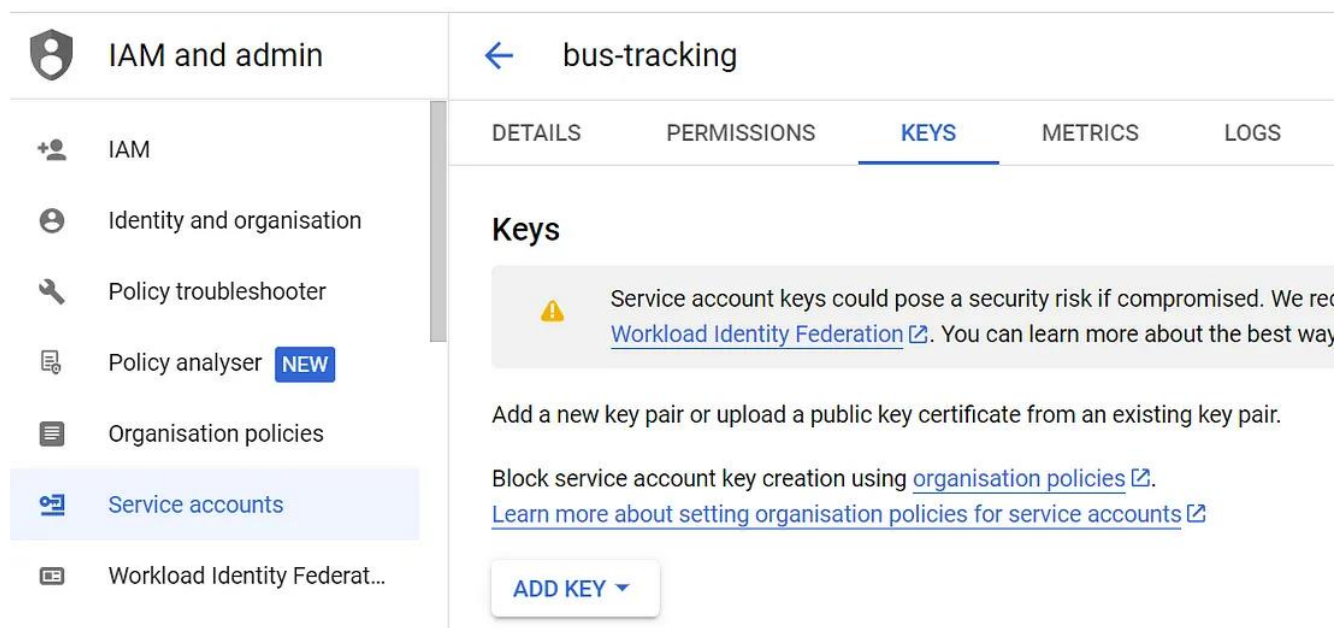


Рисунок 4.5. Створення нового ключу Google Cloud IAM

Новий ключ буде завантажено у вигляді JSON-файлу. Після цього встановлюємо Google Cloud SDK та змінну середовища наших облікових даних на місце розташування JSON-файлу ключа, який перенаправить нас на сторінку входу в браузер для входу у наш обліковий запис Google. Після підтвердження ми будемо автентифіковані й зможемо локально взаємодіяти з нашим хмарним середовищем Google.

Після цього нам потрібно додати дозволи до облікового запису служби, щоб ми могли взаємодіяти з різними службами, які нам потрібні. Для цього перейдемо до IAM, а потім поруч з обліковим записом служби натиснемо на «Edit» головного адміністратора і призначимо наступні ролі (рис. 4.6).

Name ↑	Role
bus-tracking	BigQuery Admin
	BigQuery Job User
	BigQuery User
	Cloud Run Admin
	Secret Manager Admin
	Service Account User
	Storage Admin
	Storage Object Admin
	Viewer

Рисунок 4.6. Розподіл ролей у Google Cloud IAM

Після цього потрібно увімкнути API для сервісів, які ми збираємося використовувати, щоб ми могли взаємодіяти з ними.

Після цього переходимо до налаштування Terraform. Спочатку ми визначаємо мінімально необхідну версію Terraform і серверну частину, яка буде використовуватися (де зберігається знімок стану).

Далі ми визначаємо нашого постачальника послуг (в даному випадку Google). Це схоже на імпорт бібліотеки з модулями, бо Terraform буде використовувати бібліотеку «Google» для імпорту попередньо визначених типів ресурсів, таких як Google Cloud Storage Bucket. Ми також додаємо ідентифікатор проєкту GCS, який ми створили раніше, регіон і, за бажанням, можемо вказати облікові дані (в нашому випадку це не потрібно, оскільки ми вже пройшли авторизацію в GCS).

Потім ми можемо визначити різні ресурси, які ми хочемо створити. Реєстр Terraform містить шаблони для створення ресурсів у різних хмарних провайдерів (рис. 4.7).

```
1 terraform {
2   required_version = ">=1.0"
3   backend "local" {}
4   required_providers {
5     google = {
6       source = "hashicorp/google"
7     }
8   }
9 }
10
11 provider "google" {
12   project = var.project_id
13   region = var.region
14   //credentials = file(var.credentials) #use this if you don't want to set env-var GOOGLE_APPLICATION
15 }
16
17 # GCS storage bucket
18 resource "google_storage_bucket" "bus_data_lake" {
19   name      = var.bucket_name
20   location  = var.region
21
22   # Optional, but recommended settings:
23   storage_class = var.storage_class
24   uniform_bucket_level_access = true
25
26   versioning {
27     enabled = true
28   }
29
30   force_destroy = true
31 }
32
33 # Artifact registry for containers
34 resource "google_artifact_registry_repository" "bus-container-registry" {
35   location      = var.region
36   repository_id = var.registry_id
37   format        = "DOCKER"
38 }
```

Рисунок 4.7. Створення сховища Terraform

Для нашого сховища нижче ми визначаємо його назву і регіон, а для реєстру артефактів ми визначаємо його ідентифікатор, регіон і формат (у нашому випадку Docker). Ці змінні мають префікс var, а фактичні значення для них оголошуються в нашому окремому файлі tfvars.

variables.tf охопить оголошення наших змінних. Кожна вхідна змінна оголошується за допомогою блоку змінних (рис. 4.8).



```
1 variable "project_id" {
2   type= string
3   description = "GCS project ID"
4 }
5
6 variable "bucket_name" {
7   type =string
8   description= "Name of Google Storage Bucket to create"
9 }
10
11 variable "region" {
12   type = string
13   description = "Region for GCP resources"
14   default = "europe-west6"
15 }
16
17 variable "storage_class" {
18   type = string
19   description = "Storage class type for bucket."
20   default = "STANDARD"
21 }
22
23
24 variable "registry_id" {
25   type = string
26   description = "Name of artifact registry repository."
27 }
```

Рисунок 4.8. variables.tf

Він містить назву змінної, тип, короткий опис і значення за замовчуванням, якщо потрібно. Ми можемо встановити значення наших змінних у цьому файлі, але для зручності краще виокремити їх в інший файл .tfvars.

Як вже зазначалось у попередніх розділах, у цьому проєкті використовуються дані з британського сервісу відкритих даних Bus Open Data Service. Для цього створюємо bus\_live\_locations.py, який витягує актуальні місця розташування автобусів з потоку Open Bus Data GTFS для області, визначеної координатами обмежувальної рамки (наприклад, місто Лідс). Дані зберігаються у вигляді Parquet файлу і завантажуються до сховища Google.

bus\_timetables.py отримує останні дані розкладу автобусів у форматі GTFS для відповідного регіону, трансформує їх і завантажує Parquet файл у сховище.

compare\_bus\_times.py отримує файли розташування автобусів у реальному часі та розкладів руху автобусів, трансформує їх та обчислює, які автобуси наразі

Вебресурс для покращення комфортного перевезення пасажирів автобусного транспорту запізнюються на наступній зупинці. Результатом є невеликий csv-файл, що містить список автобусів, які запізнюються, та пов'язану з ними інформацію про маршрут. Файл csv завантажується у кошик.

`write_to_bq.py` отримує csv-файл із запізненням автобусів і додає дані до нашої таблиці BigQuery.

Вище описані відносно прості кроки вилучення та перетворення даних, і скрипти можна запускати окремо в локальному середовищі, але наша мета полягає в тому, щоб вони виконувалися за розкладом у хмарному середовищі. Саме тут на допомогу приходить наш Prefect.

У Prefect потоки і завдання – це будівельні блоки нашого робочого процесу з даними. Потік як функція, яка може отримувати вхідні дані і створювати вихідні. Завдання – це, як правило, окремі, дискретні частини роботи в потоці. Наприклад, наша функція завантаження даних розкладу руху автобусів GCS є окремим завданням. Потоки можуть викликати завдання або інші потоки (відомі як підпотоки).

Чудовою особливістю Prefect є те, що ми можемо зробити наші наявні функції завданням або потоком, просто скориставшись декоратором завдань або потоків. У Додатку А можна спостерігати, що функція `load_live_locations_to_gcs` має декоратор `@task()`, а функція `get_live_bus_locations` визначена як потік за допомогою `@flow()` – цей потік викликає завдання `load_live_locations_to_gcs` і `get_live_gtfs`. При цьому Prefect автоматично додає лог-файл, щоб ми могли переглянути час виконання та кінцевий стан потоку (рис. 4.9).

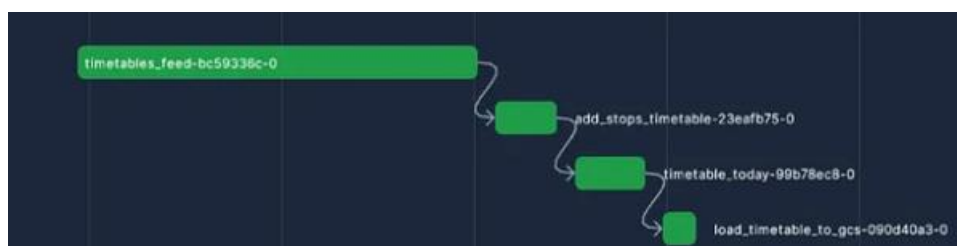


Рисунок 4.9. Лог файл у Perfect

Розгортання Prefect дозволяє нам запускати і планувати наші потоки замість того, щоб запускати їх вручну. Збираємо файл розгортання за допомогою `prefect deployment build`. У результаті буде створено `yaml`-файл розгортання. У файлі розгортання можна вказати значення параметрів за замовчуванням, які потім використовуватимуться потоками, пов'язаними з цим розгортанням.

Для будь-якого потокового розгортання можна встановити розклад. Розклад можна визначити у файлі розгортання або у самому інтерфейсі Prefect.

Щоб оновити потік Prefect, включивши до нього прогнозування регресора GBT, необхідно буде просто створити нове завдання в потоці Prefect, яке запустить прогнозування регресора GBT. Це завдання завантажить необхідні дані, запустить прогноз, а потім збереже результати або в базі даних, або у файлі CSV.

Далі необхідно визначити з методом запуску нашого вебресурсу у хмарному середовищі. Ми будемо використовувати Docker та Google Cloud Run. Наш образ Docker буде містити наше базове середовище для виконання, а ми будемо використовувати Cloud Run для запуску контейнера. Важливо пам'ятати про різницю між джерелом потоку (який є фактичним кодом потоку, наприклад, наш скрипт `bus_live_locations.py`) і залежностями потоку. Залежності потоку зібрані в нашому файлі `requirements.txt` і встановлені в нашому базовому середовищі в образі Docker.

Останнє завдання налаштування передбачає використання локальної системи для запуску агента Prefect. Агент відповідає за перевірку готових до запуску потоків і запуск їх виконання. Для того, щоб ми могли повністю запустити проєкт у хмарі, нам потрібно налаштувати нашого агента на екземплярі Google Compute Engine:

- 1) створити обчислювальний екземпляр;
- 2) під'єднатися по SSH до віртуальної машини. Можемо скористатися кнопкою прямого SSH-підключення в консолі Google;

3) створити новий командний скрипт для встановлення необхідних пакунок в(Python та Prefect, а потім увійти до Prefect за допомогою нашого ключа Prefect API);

4) зберігаємо файл та запускаємо скрипт.

Для подальшого запуску агента використовуємо tmux, щоб агент продовжував працювати після того, як ми розірвемо SSH-з'єднання.

Для фінального розгортання проєкту використовуємо Streamlit (бібліотека з відкритим вихідним кодом, яка дозволяє швидко і легко створювати веб-додатки для роботи з даними на чистому Python) (рис. 4.10).

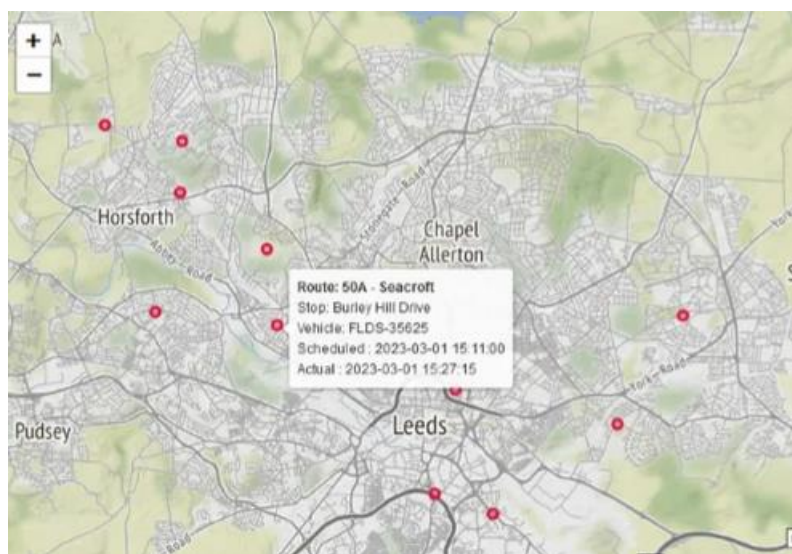


Рисунок 4.10. Streamlit додаток з використанням Folium

При цьому можливо розмістити свій додаток на власному хостингу або скористатися хмарним сервісом Streamlit для безкоштовного розгортання. Для створення карти автобусів було використано бібліотеку Streamlit-Folium, яка інтегрує Streamlit з картографічною бібліотекою Folium.

### **Висновок до розділу 3**

Отже, наше дослідження забезпечує надійну основу для вилучення цінності з сучасних даних та інфраструктурних інструментів, що дозволяє створювати ефективний вебресурс.

Налаштування надійного середовища розробки має вирішальне значення для успіху будь-якого проєкту. У цьому дослідженні ми використовуємо Visual Studio Code (VS Code) як редактор коду. Основні інструменти та бібліотеки, які прискорили процес розробки, включають:

1. Black як форматер коду Python, що забезпечує єдиний стиль коду, особливо корисний для командної роботи. Використовуючи Black, ми можемо зосередитися на написанні коду, а не на його форматуванні.

2. Flake8, лінтер коду, що перевіряє код на відповідність PEP і виявляє синтаксичні помилки, невикористані змінні та інші поширені проблеми. Конфлікти між Black та Flake8 вирішуються шляхом налаштування параметрів Flake8 у файлі `setup.cfg`.

3. Pre-commit hooks забезпечують автоматичне форматування коду Black і перевірку Flake8 перед фіксацією, що сприяє впорядкованому та послідовному робочому процесу.

Для управління залежностями проєкту використовувалися `pip-compile` і `pip-tools`, що забезпечують передбачуваність та детермінованість збірки.

Продемонстрований робочий процес слугує цінним зразком для фахівців з даних, які прагнуть надавати інформацію в режимі реального часу, а його адаптивність забезпечує його актуальність у різних сферах і додатках.

## ВИСНОВКИ

Отже, дане дослідження продемонструвало ефективне використання сучасних інформаційних технологій та машинного навчання для підвищення комфорту пасажирів під час автобусних перевезень шляхом розробки спеціалізованого вебресурсу. Основні висновки та досягнення дослідження можна узагальнити наступним чином:

1. Проведено детальний огляд поточного стану інформаційного сектору в галузі автобусних перевезень. Це включало вивчення наявних вебресурсів та додатків, спрямованих на підвищення комфорту пасажирів. Дослідження виявило прогалини та можливості для вдосконалення цих технологій за допомогою більш досконалих методів.

2. Дослідження включало визначення та огляд моделей та методів прогнозування на основі машинного навчання для кращого прогнозування руху автобусного транспорту та підвищення комфорту пасажирів.

3. Було розглянуто моделі машинного навчання для аналізу параметрів пересування автобусного транспорту, зокрема модель на основі методу Gradient Boosted Trees в PySpark, яка пропонує надійний і гнучкий підхід до задач регресії та класифікації, здатний досягти високої точності шляхом ітеративного вдосконалення ансамблю дерев рішень. Її конфігураційна природа дозволяє тонко налаштовувати її відповідно до конкретних вимог задачі, що робить її цінним інструментом у контексті розробки вебресурсу для прогнозування руху автобусного транспорту.

4. В рамках проекту було окреслено технічний дизайн веб-ресурсу, включаючи визначення структури бази даних та розробку алгоритмів для обробки маршрутів і розкладу. Для забезпечення надійної реалізації веб-ресурсу були обрані відповідні мови програмування, фреймворки та інструменти. Цей вибір був зумовлений потребою в ефективності, масштабованості та простоті обслуговування.

5. Реалізація була розроблена з урахуванням можливості масштабування, що дозволяє адаптувати та розширювати ресурс для ширшого застосування в системах громадського транспорту.

У дослідженні також були розглянуті аспекти охорони праці, що гарантує, що розробка і розгортання вебресурсу відповідає відповідним стандартам і практикам безпеки.

Подальше вдосконалення та тестування моделі машинного навчання на більшому наборі даних для підвищення її точності та надійності. Розширення вебресурсу для включення додаткових функцій, таких як зворотний зв'язок з пасажирями в режимі реального часу та інтеграція з іншими системами громадського транспорту.

Таким чином, це дослідження забезпечує комплексну основу для використання сучасних технологій для підвищення комфорту пасажирів під час автобусних перевезень. Розроблений вебресурс слугує цінним інструментом як для перевізників, так і для пасажирів, сприяючи підвищенню комфорту та ефективності користування громадським транспортом.

### СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Компромiс Парето мiж критерiями ефективностi процесу формування розкладу навчальних занять. Репозиторiй ОНТУ. URL: <https://card-file.ontu.edu.ua/items/cb5d4bad-f661-4c7d-924a-72928fc6eaa> (дата звернення: 10.06.2024).
2. Нейромереживо. Аналіз та прогнозування часових рядів. Друкарня. URL: <https://drukarnia.com.ua/articles/analiz-ta-prognozuvannya-chasovikh-ryadiv-H6gg0> (дата звернення: 10.06.2024).
3. Разом до сталої і розумної мобільності: Що таке Стратегія сталої і розумної мобільності ЄС та як її можна застосовувати в Україні. Екодія. URL: <https://ecoaction.org.ua/razom-do-staloi-mobilnosti.html> (дата звернення: 10.06.2024).
4. Смарт-тахографи 2-го покоління. Website for transport and logistics. URL: <https://trans.info/ua/smart-tahografy-2-go-pokolinnya-vse-shho-potribno-znaty-do-21-serpnya-358766> (дата звернення: 10.06.2024).
5. A Deep Learning Approach for Estimating Traffic Density Using Data Obtained from Connected and Autonomous Probes / D. Nam et al. Sensors. 2020. Vol. 20, no. 17. P. 4824. URL: <https://doi.org/10.3390/s20174824> (date of access: 10.06.2024).
6. A Spatiotemporal Neural Network Model for Estimated-Time-of-Arrival Prediction of Flights in a Terminal Maneuvering Area / Y. Ma et al. IEEE Intelligent Transportation Systems Magazine. 2022. P. 2–15. URL: <https://doi.org/10.1109/mits.2021.3132766> (date of access: 10.06.2024).
7. Agarwal V. Making Historical Weather Data for any place using Dark Sky API calls. Medium. URL: <https://medium.com/analytics-vidhya/making-historical-weather-data-for-any-place-using-dark-sky-api-calls-d5876de0ec01> (date of access: 10.06.2024).
8. Alsaleh N., Farooq B. Interpretable data-driven demand modelling for on-demand transit services. Transportation Research Part A: Policy and Practice. 2021.



Vol. 154. P. 1–22. URL: <https://doi.org/10.1016/j.tra.2021.10.001> (date of access: 10.06.2024).

9. Bie Y., Wang D., Qi H. Prediction Model of Bus Arrival Time at Signalized Intersection Using GPS Data. *Journal of Transportation Engineering*. 2012. Vol. 138, no. 1. P. 12–20. URL: [https://doi.org/10.1061/\(asce\)te.1943-5436.0000310](https://doi.org/10.1061/(asce)te.1943-5436.0000310) (date of access: 10.06.2024).

10. 19. Big data defined: examples and benefits. Google Cloud. URL: <https://cloud.google.com/learn/what-is-big-data> (date of access: 10.06.2024).

11. Bus Open Data Service. Bus Open Data Service. URL: <https://www.bus-data.dft.gov.uk/> (date of access: 10.06.2024).

12. Cao H., Wu Z., Yu W. Drone for Intelligent Traffic Monitoring: Current Status and Future Trends. *Computational and Experimental Simulations in Engineering*. Cham, 2024. P. 1133–1150. URL: [https://doi.org/10.1007/978-3-031-44947-5\\_88](https://doi.org/10.1007/978-3-031-44947-5_88) (date of access: 10.06.2024).

13. Could autonomous vehicles put last-mile delivery on the fast track?. WEF. URL: <https://www.weforum.org/agenda/2021/12/last-mile-delivery-autonomous-vehicles/> (date of access: 10.06.2024).

14. Dimitriadou S., Nikolakopoulos K. G. Multiple Linear Regression Models with Limited Data for the Prediction of Reference Evapotranspiration of the Peloponnese, Greece. *Hydrology*. 2022. Vol. 9, no. 7. P. 124. URL: <https://doi.org/10.3390/hydrology9070124> (date of access: 10.06.2024).

15. Estimating Transit Ridership Patterns Through Automated Data Collection Technology. Westernite. URL: [https://www.westernite.org/annualmeetings/17\\_San\\_Diego/Presentations/5A-Kim.pdf](https://www.westernite.org/annualmeetings/17_San_Diego/Presentations/5A-Kim.pdf).

16. Head-to-head comparison of clustering methods for heterogeneous data: a simulation-driven benchmark / G. Preud'homme et al. *Scientific Reports*. 2021. Vol. 11, no. 1. URL: <https://doi.org/10.1038/s41598-021-83340-8> (date of access: 10.06.2024).

17. Lee H., Lee J., Chung Y. Traffic density estimation using vehicle sensor data. *Journal of Intelligent Transportation Systems*. 2021. P. 1–15. URL: <https://doi.org/10.1080/15472450.2021.1966626> (date of access: 10.06.2024).
18. Machine Learning Applied to Public Transportation by Bus: A Systematic Literature Review / T. Alexandre et al. *Transportation Research Record: Journal of the Transportation Research Board*. 2023. P. 036119812311551. URL: <https://doi.org/10.1177/03611981231155189> (date of access: 10.06.2024).
19. Madakam S. Internet of things: smart things. *International journal of future computer and communication*. 2015. Vol. 4, no. 4. P. 250–253. URL: <https://doi.org/10.7763/ijfcc.2015.v4.395> (date of access: 10.06.2024).
20. Niu H., Silva E. A. Crowdsourced Data Mining for Urban Activity: Review of Data Sources, Applications, and Methods. *Journal of Urban Planning and Development*. 2020. Vol. 146, no. 2. P. 04020007. URL: [https://doi.org/10.1061/\(asce\)up.1943-5444.0000566](https://doi.org/10.1061/(asce)up.1943-5444.0000566) (date of access: 10.06.2024).
21. Optimizing the bus operation plan Based on Deep Learning / Y. Lv et al. *Microprocessors and Microsystems*. 2021. P. 104042. URL: <https://doi.org/10.1016/j.micpro.2021.104042> (date of access: 10.06.2024).
22. Patnaik J., Chien S., Bladikas A. Estimation of Bus Arrival Times Using APC Data. *Journal of Public Transportation*. 2004. Vol. 7, no. 1. P. 1–20. URL: <https://doi.org/10.5038/2375-0901.7.1.1> (date of access: 10.06.2024).
23. Perform Systematic Literature Reviews. Parsifal. URL: <https://parsif.al/> (date of access: 10.06.2024).
24. Predict arrival time by using machine learning algorithm to promote utilization of urban smart bus. ResearchGate. URL: [https://www.researchgate.net/publication/339333121\\_Predict\\_Arrival\\_Time\\_by\\_Using\\_Machine\\_Learning\\_Algorithm\\_to\\_Promote\\_Utilization\\_of\\_Urban\\_Smart\\_Bus](https://www.researchgate.net/publication/339333121_Predict_Arrival_Time_by_Using_Machine_Learning_Algorithm_to_Promote_Utilization_of_Urban_Smart_Bus) (date of access: 10.06.2024).

25. Prediction of arrival times and human resources allocation for container terminal / G. Fancello et al. *Maritime Economics & Logistics*. 2011. Vol. 13, no. 2. P. 142–173. URL: <https://doi.org/10.1057/mel.2011.3> (date of access: 10.06.2024).
26. Predictive analytics on open big data for supporting smart transportation services / P. P. F. Balbin et al. *Procedia computer science*. 2020. Vol. 176. P. 3009–3018. URL: <https://doi.org/10.1016/j.procs.2020.09.202> (date of access: 10.06.2024).
27. Real-Time bus arrival prediction: a deep learning approach for enhanced urban mobility. *Arxiv*. URL: <https://arxiv.org/html/2303.15495v3>.
28. Skhosana M., Ezugwu A. E. A Real-Time Machine Learning-Based Public Transport Bus-Passenger Information System. *International Journal of Sustainable Development and Planning*. 2021. Vol. 16, no. 7. P. 1221–1238. URL: <https://doi.org/10.18280/ijstdp.160703> (date of access: 10.06.2024).
29. Suhag D., Jha V. A COMPREHENSIVE SURVEY ON MOBILE CROWDSENSING SYSTEMS. *Journal of Systems Architecture*. 2023. P. 102952. URL: <https://doi.org/10.1016/j.sysarc.2023.102952> (date of access: 10.06.2024).
30. Systematic Review of the Literature on Big Data in the Transportation Domain: Concepts and Applications / A. Neilson et al. *Big Data Research*. 2019. Vol. 17. P. 35–44. URL: <https://doi.org/10.1016/j.bdr.2019.03.001> (date of access: 10.06.2024).
31. Temporal Hierarchical Graph Attention Network for Traffic Prediction / L. Huang et al. *ACM Transactions on Intelligent Systems and Technology*. 2021. Vol. 12, no. 6. P. 1–21. URL: <https://doi.org/10.1145/3446430> (date of access: 10.06.2024).
32. Towards a systematic approach to design new ensemble learning algorithms. *Arxiv*. URL: <https://arxiv.org/abs/2402.06818>. Transit Vehicle Arrival Prediction: Algorithm and Large-Scale Implementation / D. J. Dailey et al. *Transportation Research Record: Journal of the Transportation Research Board*. 2001. Vol. 1771, no. 1. P. 46–51. URL: <https://doi.org/10.3141/1771-06> (date of access: 10.06.2024).

33. Understanding Kalman Filters. MathWorks. URL: <https://www.mathworks.com/videos/series/understanding-kalman-filters.html> (date of access: 10.06.2024).

34. What Is Automatic Vehicle Location (AVL) And How Does It Work?. Track Your Truck. URL: <https://www.trackyourtruck.com/blog/what-is-automatic-vehicle-location/> (date of access: 10.06.2024).

## ДОДАТКИ

### ДОДАТОК А

#### Функція «load\_live\_locations\_to\_gcs»

```
@task()
def load_live_locations_to_gcs(
    pref_gcs_block_name: str, from_path: str, to_path: str
) -> None:
    """Load the live bus locations to Google Bucket"""

    gcs_block = GcsBucket.load(pref_gcs_block_name)
    gcs_block.upload_from_path(from_path=from_path, to_path=to_path)

    os.remove(from_path)
    return None

@flow()
def get_live_bus_locations(
    area_coords: dict = {
        "min_lat": 53.725,
        "max_lat": 53.938,
        "min_long": -1.712,
        "max_long": -1.296,
    },
    pref_gcs_block_name: str = "bus-tracker-gcs-bucket",
    live_locations_filename: str = "live_location",
):
```

```
get_live_gtfs(  
    area_coords.get("min_lat"),  
    area_coords.get("max_lat"),  
    area_coords.get("min_long"),  
    area_coords.get("max_long"),  
    filename=live_locations_filename,  
)  
  
load_live_locations_to_gcs(  
    wait_for=[get_live_gtfs],  
    pref_gcs_block_name=pref_gcs_block_name,  
    from_path=f"{live_locations_filename}.parquet.gzip",  
    to_path=f"live_location/{live_locations_filename}.parquet.gzip",  
)  
  
if __name__ == "__main__":  
  
    get_live_bus_locations()
```

## ДОДАТОК Б

Модель прогнозування руху автобусного транспорту за допомогою PySpark

```
from pyspark.sql import SparkSession
from pyspark.sql import functions as F
from pyspark.sql.window import Window
from pyspark.sql.types import LongType, StringType
from math import radians, cos, sin, asin, sqrt
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.regression import LinearRegression, DecisionTreeRegressor,
GBTRRegressor
from pyspark.ml.evaluation import RegressionEvaluator

# Initialize Spark session
spark = SparkSession.builder.appName("BusDataProcessing").getOrCreate()

# Read JSON data
total_df = spark.read.json('busFile path')

# Filter data where atStop == True
total_df = total_df.filter(total_df.atStop == True)

# Select relevant columns and rename them
total_df = total_df.select('_id', 'atStop', 'busStop', 'congestion', 'delay', 'justLeftStop',
'justStopped',
                           'latitude', 'longitude', 'vehicleId', 'vehicleSpeed', 'actualDelay',
'timestamp',
                           'journeyPatternId') \
    .withColumnRenamed('_id', 'id') \
```

```
.withColumnRenamed('justLeftStop', 'just_left') \  
.withColumnRenamed('justStopped', 'just_stopped') \  
.withColumnRenamed('congestion', 'conges')  
  
# Define UDFs for data transformation  
def get_weekday(s):  
    return s.weekday()  
  
def get_hour(s):  
    return s.hour  
  
def get_is_weekend(s):  
    return 1 if s in [6, 7] else 0  
  
def get_id(s):  
    return str(s['$oid'])  
  
def get_month(s):  
    return s.month  
  
def get_only_date(s):  
    return str(s.date())  
  
def get_flag(value):  
    return 1 if value else 0  
  
def get_timestamp(s):  
    return int(s['$numberLong'][::-3])
```



```
# Register UDFs
```

```
spark.udf.register("get_timestamp", get_timestamp, LongType())  
spark.udf.register("get_weekday", get_weekday, LongType())  
spark.udf.register("get_hour", get_hour, LongType())  
spark.udf.register("get_is_weekend", get_is_weekend, LongType())  
spark.udf.register("get_id", get_id, StringType())  
spark.udf.register("get_month", get_month, LongType())  
spark.udf.register("get_only_date", get_only_date, StringType())  
spark.udf.register("get_flag", get_flag, LongType())
```

```
# Apply UDFs for data transformation
```

```
total_df = total_df.withColumn('date', F.from_unixtime(F.col('timestamp') /  
1000).cast('timestamp'))  
total_df = total_df.withColumn('weekday', F.expr("get_weekday(date)"))  
total_df = total_df.withColumn('month', F.expr("get_month(date)"))  
total_df = total_df.withColumn('only_date', F.expr("get_only_date(date)"))  
total_df = total_df.withColumn('is_weekend', F.expr("get_is_weekend(weekday)"))  
total_df = total_df.withColumn('hour', F.expr("get_hour(date)"))  
total_df = total_df.withColumn('id', F.expr("get_id(id)"))  
total_df = total_df.withColumn('just_left', F.expr("get_flag(just_left)"))  
total_df = total_df.withColumn('just_stopped', F.expr("get_flag(just_stopped)"))  
total_df = total_df.withColumn('conges', F.expr("get_flag(conges)"))
```

```
# Drop unnecessary columns
```

```
total_df = total_df.drop('atStop', 'justLeftStop', 'justStopped', 'congestion', 'timestamp')
```

```
# Define distance calculation UDFs
```

```
@F.udf("float")
```

```
def calculate_distance(longit_a, latit_a, longit_b, latit_b):
    if None in [longit_a, latit_a, longit_b, latit_b]:
        return 9999

    longit_a, latit_a, longit_b, latit_b = map(radians, [longit_a, latit_a, longit_b,
latit_b])

    dist_longit = longit_b - longit_a
    dist_latit = latit_b - latit_a
    area = sin(dist_latit / 2) ** 2 + cos(latit_a) * cos(latit_b) * sin(dist_longit / 2) ** 2
    central_angle = 2 * asin(sqrt(area))
    radius = 6371 # Radius of Earth in km
    distance = central_angle * radius
    return distance * 1000
```

```
@F.udf("float")
```

```
def distance_from_center(longit_a, latit_a):
    city_center_coords = (53.3422665, -6.2554468) # City center coordinates
    if None in [longit_a, latit_a]:
        return 9999

    longit_a, latit_a = map(radians, [longit_a, latit_a])
    longit_b, latit_b = map(radians, city_center_coords)
    dist_longit = longit_b - longit_a
    dist_latit = latit_b - latit_a
    area = sin(dist_latit / 2) ** 2 + cos(latit_a) * cos(latit_b) * sin(dist_longit / 2) ** 2
    central_angle = 2 * asin(sqrt(area))
    radius = 6371 # Radius of Earth in km
    distance = central_angle * radius
    return distance * 1000
```

```
# Apply distance UDFs and filter out extreme values
total_df = total_df.withColumn('distance_between', calculate_distance(F.col('prev_lat'),
F.col('prev_lon'), F.col('latitude'), F.col('longitude')))
total_df = total_df.withColumn('lat_between', (total_df.prev_lat + total_df.latitude) / 2)
total_df = total_df.withColumn('lon_between', (total_df.prev_lon + total_df.longitude) /
2)
total_df = total_df.withColumn('distance_from_center',
distance_from_center(F.col('lat_between'), F.col('lon_between')))
total_df = total_df.filter((total_df.distance_between < 100000) &
(total_df.distance_from_center < 100000))

# Calculate speed in segment
total_df = total_df.withColumn('speed_in_segment', total_df.distance_between /
total_df.time_to_reach_next)
total_df = total_df.filter(total_df.speed_in_segment < 55) # Filter out segments with
speed over 200 km/h

# Count specific attributes from origin to destination
cols = ['journeyPatternId', 'origin', 'dest']
w = Window.partitionBy(cols)
total_df = total_df.withColumn('specific_count', F.count('dest').over(w))
total_df = total_df.withColumn('avg_timestamp', F.mean('date').over(w))

# Count total attributes from origin
cols = ['journeyPatternId', 'origin']
w = Window.partitionBy(cols)
total_df = total_df.withColumn('total_count', F.count('origin').over(w))
```

```
# Calculate percentage of each option 'origin -> dest'
total_df = total_df.withColumn('percentage', F.col('specific_count') /
F.col('total_count'))

# Filter for common transitions (over 25%)
total_df = total_df.filter(total_df.percentage > 0.25)

# Find most relevant timestamps for each journeyPatternId and origin
total_df = total_df.withColumn('max_percentage',
F.max('percentage').over(Window.partitionBy('journeyPatternId', 'origin')))
total_df = total_df.filter(total_df.percentage == total_df.max_percentage)
total_df = total_df.drop('max_percentage')

# Read accidents and events data
accidents_df = spark.read.csv("/FileStore/tables/accidents.csv", header=True)
events_df = spark.read.csv("/FileStore/tables/events.csv", header=True)

# Additional processing of accidents_df
accidents_df = accidents_df.withColumnRenamed('date', 'only_date')
accidents_df = accidents_df.withColumn('date_time',
accidents_df.date_time.cast('timestamp'))
accidents_df = accidents_df.withColumn('lat', accidents_df.lat.cast('float'))
accidents_df = accidents_df.withColumn('lon', accidents_df.lon.cast('float'))

# Stop the SparkSession
spark.stop()
```