

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Навчально-науковий інститут атомної та теплової енергетики

Кафедра інженерії програмного забезпечення в енергетиці

ДО ЗАХИСТУ ДОПУЩЕНО:

В. о. завідувача кафедри

____Олександр КОВАЛЬ

«_____»_202_р.

Дипломна робота

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Інженерія програмного забезпечення
інтелектуальних кібер-фізичних систем в енергетиці»**

спеціальності 121 Інженерія програмного забезпечення

**на тему: «Програмне забезпечення для розширення транспортних даних
шляхом моделювання дорожніх умов»**

Виконав:

Студент IV курсу, групи ТВ-22

Білогривий Владислав Юрійович

(прізвище, ім'я, по батькові)

(підпис)

Керівник:

Доцент, к.е.н., Гусєва Ірина Ігорівна

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Рецензент:

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____

Київ — 2026

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Навчально-науковий інститут атомної та теплової енергетики
Кафедра інженерії програмного забезпечення в енергетиці
Рівень вищої освіти перший (бакалаврський)
Спеціальності: 121 Інженерія програмного забезпечення
Освітньо-професійна програма: «Інженерія програмного забезпечення
інтелектуальних кібер-фізичних систем в енергетиці»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
Олександр КОВАЛЬ
(підпис)

«_____» 202_ р.

ЗАВДАННЯ

на дипломну роботу студенту

Білогривий Владислав Юрійович

(прізвище, ім'я, по батькові)

1. Тема роботи

Програмне забезпечення для розширення транспортних даних шляхом моделювання дорожніх умов

керівник роботи Гусєва Ірина Ігорівна, к.е.н.

(прізвище, ім'я, по батькові науковий ступінь, вчене звання)

затверджена наказом по університету від «28» травня 2026 р. № 1992-с

2. Строк подання студентом роботи 3 червня 2026 р.

3. Вихідні дані до роботи мова програмування JavaScript (ES2020+), середовище розробки VS Code, бібліотека візуалізації Chart.js, датасет T-Drive.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) розробити програмне забезпечення для розширення транспортних даних шляхом моделювання дорожніх умов, що включає аналіз існуючих методів симуляції транспортних потоків, проектування архітектури вебзастосунку з модулями завантаження та парсингу GPS-даних, візуалізації та порівняння реальних і синтетичних даних, а також генерації й експорту розширеного датасету у форматах CSV та JSON.

5. Перелік ілюстративного матеріалу Use-case діаграма програмного застосунку; діаграма взаємодії частин системи; конвеєр обробки та аналізу даних; послідовність взаємодії при завантаженні та обробці.
6. Дата видачі завдання «30» жовтня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Строки виконання етапів роботи	Примітки
1.	Отримання завдання	30.10.2025	Виконано
2.	Дослідження предметної області	31.10.2025 — 31.12.2025	Виконано
3.	Постановка вимог до проєктування системи	01.01.2026 — 15.02.2026	Виконано
4.	Дослідження існуючих рішень	16.02.2026 — 01.04.2026	Виконано
5.	Розробка програмного продукту	01.04.2026 — 05.05.2026	Виконано
6.	Тестування програмного продукту	05.05.2026 — 12.05.2026	Виконано
7.	Захист програмного продукту	12.05.2026	Виконано
8.	Оформлення дипломної роботи	12.05.2026 — 23.05.2026	Виконано
9.	Передзахист	03.06.2026	Виконано
10.	Захист	17.06.2026	Виконано

Студент

(підпис)

Владислав БІЛОГРИВИЙ

(ім'я, ПРІЗВИЩЕ)

Керівник роботи

(підпис)

Ірина ГУСЄВА

(ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Структура та обсяг дипломної роботи. Робота містить 61 сторінок, 18 рисунків, 2 додатки та 23 джерела.

Метою роботи є створення програмного забезпечення для розширення транспортних даних шляхом моделювання дорожніх умов — інструменту, що дозволяє генерувати синтетичні транспортні датасети з урахуванням динамічного впливу погодних явищ та стану дорожнього покриття на поведінку транспортних засобів.

Для досягнення поставленої мети було проведено аналіз існуючих моделей поведінки транспортних засобів та симуляційних платформ (SUMO, PTV Vissim, MovingPandas, Geotab), досліджено вплив дорожніх умов на кінематичні параметри руху, спроектовано архітектуру системи з модульним розділенням відповідальності між підсистемами завантаження, обробки, симуляції та візуалізації, реалізовано та протестовано повний конвеєр обробки реальних GPS-треків із подальшою генерацією розширеного датасету.

Розроблено вебзастосунок, що у клієнтському середовищі браузера виконує: завантаження та парсинг GPS-даних транспортних засобів, розрахунок кінематичних параметрів (швидкість, прискорення, кут повороту), класифікацію стилів водіння, симуляцію сценаріїв дорожніх умов (суха дорога, дощ, сніг, ожеледиця тощо), генерацію розширеного датасету та його експорт у форматах CSV і JSON.

Практичне значення одержаних результатів полягає у створенні інструменту, що забезпечує дослідників та інженерів датасетами для завдань інтелектуального аналізу транспортних даних — без необхідності дорогого збору реальних даних в екстремальних дорожніх умовах.

Ключові слова: *транспортні дані, GPS-треки, T-Drive, моделювання дорожніх умов, симуляція, класифікація стилю водіння, синтетичні дані, вебзастосунок, JavaScript, Chart.js.*

ABSTRACT

Structure and scope of the thesis. The work contains 61 pages, 18 figures, 2 appendices and 23 references.

The aim of the work is to create software for expanding transport data by modeling road conditions — a tool that allows generating synthetic transport datasets taking into account the dynamic impact of weather phenomena and road surface conditions on the behavior of vehicles.

To achieve this goal, an analysis of existing models of vehicle behavior and simulation platforms (SUMO, PTV Vissim, MovingPandas, Geotab) was conducted, the impact of road conditions on kinematic parameters of movement was studied, the system architecture was designed with a modular division of responsibility between the subsystems of loading, processing, simulation and visualization, and a complete pipeline for processing real GPS tracks with subsequent generation of an expanded dataset was implemented and tested.

A web application has been developed that performs the following tasks in the browser client environment: loading and parsing vehicle GPS data (T-Drive format), calculating kinematic parameters (speed, acceleration, steering angle), classifying driving styles, simulating road condition scenarios (dry road, rain, snow, ice, etc.), generating an extended dataset and exporting it in CSV and JSON formats for use in ML pipelines.

The practical significance of the results obtained lies in creating a tool that provides researchers and engineers with controlled, statistically representative, and fully labeled synthetic datasets for intelligent transport data analysis tasks — without the need for expensive field collection of real data in extreme road conditions.

Keywords: *transport data, GPS tracks, T-Drive, road condition modeling, simulation, driving style classification, synthetic data, web application, JavaScript, Chart.js.*

ЗМІСТ

1 ПОСТАНОВКА ЗАВДАННЯ	10
2 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ	12
2.1 Моделювання дорожніх умов	12
2.2 Огляд існуючих рішень	13
2.3 Огляд датасету T-Drive	18
2.4 Алгоритм класифікації стилю водіння	19
3 ЗАСОБИ РОЗРОБКИ	22
3.1 Вибір мови програмування та середовища виконання	22
3.2 Бібліотека візуалізації Chart.js	23
3.3 Стилiзація та формати даних	24
4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ	27
4.1 Архітектура застосунку	27
4.2 Модуль завантаження та парсингу даних	31
4.3 Модуль симуляції та розширення	33
4.4 Модуль візуалізації та порівняння	34
5 ОПИС РОБОТИ КОРИСТУВАЧА ІЗ СИСТЕМОЮ	38
5.1 Системні вимоги та налаштування	38
5.2 Завантаження даних	38
5.3 Перегляд статистики та графіків	39
5.4 Симуляція дорожніх умов	41
5.5 Порівняння реальних та симульованих даних	43
5.6 Генерація та експорт розширеного датасету	45
ВИСНОВКИ	47

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	49
ДОДАТОК А	52
ДОДАТОК Б	56

ВСТУП

Актуальність теми зумовлена сучасною проблематикою у сфері транспорту, яка пов'язана з нестачею якісних, різноманітних і деталізованих даних для навчання та тестування інтелектуальних систем. В умовах зростання вимог до безпеки транспортної галузі, а також потреби у точному моделюванні міської мобільності, особливої вагомості набуває наявність даних, що враховують вплив реальних дорожніх умов — таких як дощ, ожеледиця чи пошкоджене покриття. Використання синтетичних даних у цій сфері дозволяє перейти від обмежених реальних наборів до повноцінних, керованих і відтворюваних сценаріїв. Тобто технології моделювання та симуляції мають можливість зробити вагомий внесок у підвищення надійності та безпеки сучасних транспортних систем.

Традиційні методи збору транспортних даних ґрунтуються на реальних спостереженнях, які не враховують рідкісні або екстремальні дорожні ситуації та не охоплюють усі можливі комбінації умов і станів інфраструктури. У результаті такі підходи зазнають певної втрати повноти та репрезентативності. Водночас методи симуляції дозволяють генерувати реалістичні, але штучно створені дані, що враховують вплив навіть найменших зовнішніх факторів.

Практична цінність розробки полягає у тому, що вона поєднує реальну дорожню інфраструктуру, моделювання руху та динамічне застосування дорожніх умов в єдину систему.

Метою роботи є створення програмного забезпечення для генерації синтетичних транспортних даних з використанням моделі руху та динамічного застосування дорожніх умов, що дозволить дослідникам та інженерам аналізувати вплив погоди та стану покриття на поведінку транспортних засобів, а також формувати розширені датасети для подальшого використання в системах машинного навчання.

Об'єктом дослідження є процес розширення транспортних даних шляхом моделювання дорожніх умов.

Предметом дослідження є програмне забезпечення для розширення транспортних даних шляхом моделювання дорожніх умов. Воно забезпечить можливість створення реалістичних сценаріїв руху та порівняння ефективності моделювання за різних зовнішніх факторів.

1 ПОСТАНОВКА ЗАВДАННЯ

Метою роботи є створення програмного забезпечення для розширення транспортних даних шляхом моделювання дорожніх умов — інструменту, що дозволяє генерувати синтетичні транспортні датасети з урахуванням динамічного впливу погодних явищ та стану дорожнього покриття на поведінку транспортних засобів:

- проаналізувати методи моделювання дорожніх умов;
- спроектувати модульну архітектуру клієнтського вебзастосунку;
- розробити алгоритм обчислення кінематичних параметрів;
- розробити алгоритм класифікації стилю водіння;
- розробити алгоритм моделювання дорожніх умов;
- розробити модуль генерації розширеного датасету у форматах JSON та CSV з метаданими умов руху;
- реалізувати інтерактивний вебінтерфейс;
- протестувати та перевірити працездатність, стабільність і обчислювальну точність алгоритмів аналізу.

Вхідними даними є текстові файли формату T-Drive зі структурою: ідентифікатор водія, мітка часу, довгота, широта.

Вихідними даними є: статистичні показники по водіях (середня швидкість, максимальна швидкість, середнє прискорення, частка різких маневрів, клас стилю водіння); графіки розподілу стилів водіння, розподілу швидкостей, умов руху та профілю прискорень; таблиця водіїв із сортуванням, пошуком та функцією видалення; панель порівняння реальних і симульованих даних; розширений датасет у форматах JSON і CSV.

Цільовою аудиторією застосунку є:

- Data Scientist-и та ML-інженери, яким необхідні підготовлені та збалансовані датасети;

- дослідники у сфері транспортної логістики, що вивчають патерни міської мобільності й вплив дорожніх умов на поведінку водіїв;
- розробники телематичних систем, які використовують застосунок для валідації власних алгоритмів класифікації.

Висновки до розділу 1

У першому розділі сформульовано мету та основні завдання дипломної роботи, що полягають у розширенні транспортних даних шляхом моделювання дорожніх умов. Визначено вхідні та вихідні дані системи, встановлено формат вхідних файлів T-Drive та склад вихідних статистичних показників, графіків і розширеного датасету.

Конкретизовано перелік завдань: аналіз методів моделювання дорожніх умов, проєктування модульної архітектури клієнтського вебзастосунку, розробка алгоритмів обчислення кінематичних параметрів і класифікації стилю водіння, симуляції дорожніх умов, генерації та експорту розширеного датасету, а також створення інтерактивного вебінтерфейсу. Окреслено цільову аудиторію продукту — фахівців з аналізу даних, дослідників транспортної логістики та розробників телематичних систем.

2 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

2.1 Моделювання дорожніх умов

Центральним алгоритмом обробки є обчислення кінематичних параметрів за послідовними GPS-точками. Для кожної пари сусідніх точок одного водія виконується конвеєр перетворень, який складається з трьох послідовних кроків: визначення відстані, обчислення швидкості та обчислення прискорення.

Обчислення відстані між двома географічними координатами виконується за формулою Гаверсина (Haversine formula), яка враховує сферичну форму Землі [9]. На відміну від евклідового наближення у декартових координатах, формула Гаверсина забезпечує точність порядку 0,5 % навіть на відстанях у сотні кілометрів, а для типових сегментів T-Drive завдовжки кілька сотень метрів похибка є практично нехтовною. Нехай (φ_1, λ_1) і (φ_2, λ_2) — географічні координати двох точок у радіанах, $\Delta\varphi = \varphi_2 - \varphi_1$, $\Delta\lambda = \lambda_2 - \lambda_1$, а $R = 6\,371\,000$ м — середній радіус Землі. Тоді відстань обчислюється за формулою (2.1):

$$d = 2R \arcsin \sqrt{\sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos\varphi_1 \cos\varphi_2 \sin^2\left(\frac{\Delta\lambda}{2}\right)} \quad (2.1)$$

Швидкість руху на сегменті обчислюється як відношення пройденої відстані до часового інтервалу між точками за формулою (2.2):

$$v = \frac{d}{\Delta t} \quad (2.2)$$

Отримане значення обмежується діапазоном $[0; 120]$ км/год, що відповідає реалістичним умовам міського руху та відсіює аномальні викиди, спричинені пропусками GPS-сигналу. Прискорення обчислюється як скінченна різниця швидкостей між поточним і попереднім сегментами за формулою (2.3):

$$a = \frac{v_i - v_{i-1}}{\Delta t_i} \quad (2.3)$$

Результат обмежується фізично досяжним для легкового автомобіля діапазоном $[-5; +5]$ м/с² [11]. Додатково для кожної трійки послідовних точок визначається кут повороту між вектором поточного та вектором наступного сегмента; результат нормується до діапазону $[0^\circ; 180^\circ]$, а значення понад 45° кваліфікується як різкий поворот. Сукупність цих трьох похідних величин — швидкості, прискорення та кута повороту — формує базовий вектор кінематичних ознак [15], на якому ґрунтується подальша класифікація стилю водіння.

Симуляція змінених умов (функція `runSimulation()`) обчислює комбінований коефіцієнт впливу на основі параметрів погоди, стану покриття, часу доби та положень інтерактивних повзунків. Базові коефіцієнти погоди визначено на основі публікацій про вплив погодних умов на швидкість транспортного потоку [10, 19]: ясно — 1,00/1,00; туман — 0,75/0,85; дощ — 0,85/0,90; сніг — 0,70/0,75; заметіль — 0,55/0,65 (у форматі «коефіцієнт швидкості/коефіцієнт прискорення»). Коефіцієнт стану покриття множиться на погодний: суха дорога — 1,00; мокра — 0,92; ожеледь — 0,60. Часові коефіцієнти відображають добові зміни інтенсивності трафіку: день — 1,00; світанок — 0,95; сутінки — 0,90; ніч — 1,05. Повзунки рівня опадів і видимості вносять додаткові лінійні поправки у межах $\pm 10\%$. Для запобігання артефактам точного лінійного масштабування до кожної модифікованої величини додається гаусівський шум, що відтворює природні флуктуації реальних телематичних вимірювань [8].

2.2 Огляд існуючих рішень

Для обґрунтування актуальності розробки доцільно розглянути існуючі рішення у двох суміжних напрямках: моделювання та симуляція дорожнього руху, а також аналіз поведінки водія за даними телематики. Нижче наведено огляд чотирьох найбільш представницьких аналогів із зазначенням їхніх ключових переваг і недоліків стосовно завдань цієї роботи.

SUMO (Simulation of Urban MObility) — відкритий мікроскопічний мультимодальний пакет симуляції дорожнього руху, що розробляється Німецьким аерокосмічним центром (DLR) та спільнотою у межах проєкту Eclipse [3]. Кожен транспортний засіб моделюється явно, має власний маршрут і рухається мережею за моделями слідування за лідером (car-following) [13]. Пакет містить набір окремих застосунків (netconvert, duarouter, sumo-gui тощо) для імпорту мережі, генерації попиту та візуалізації [4]. Інтерфейс симуляції руху SUMO зображено на рисунку 2.1.

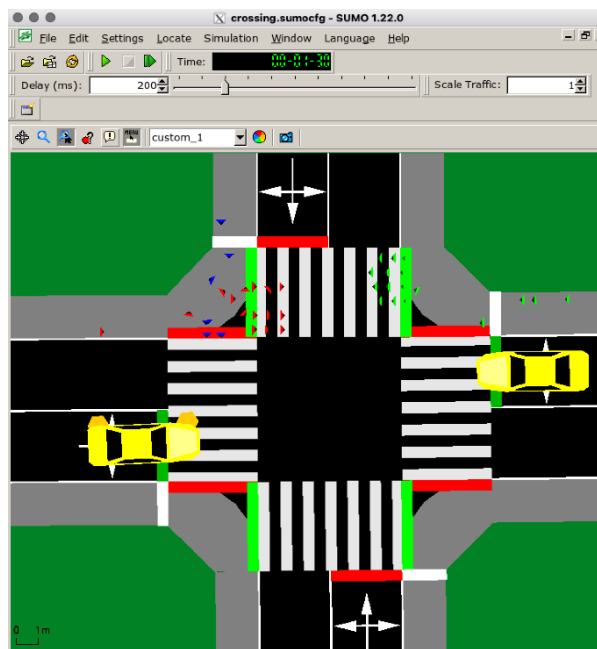


Рисунок 2.1 — Інтерфейс симуляції руху SUMO (sumo-gui)

Переваги: відкритий код (EPL 2.0); масштабованість до мереж із десятками тисяч ребер; імпорт реальних мереж (OpenStreetMap, OpenDRIVE, VISUM); підтримка керування симуляцією у реальному часі через інтерфейс TraCI; велика спільнота та потужна наукова база.

Недоліки: високий поріг входження через конфігурацію у форматі XML та роботу з окремими утилітами замість цілісного середовища; орієнтація на синтетичний світ, а не на розширення реальних GPS-даних; відсутність вбудованої класифікації стилю водіння; потреба у ретельному ручному калібруванні для отримання реалістичних патернів трафіку.

PTV Vissim — комерційний пакет мікроскопічного моделювання мультимодального руху, що широко застосовується транспортними інженерами для оцінки пропускної здатності, проектування перехресть та моделювання роботи світлофорних об'єктів [5]. Реалістичність руху забезпечується психофізичною моделлю поведінки водія Відемана (Wiedemann) [14], яка враховує сприйняття дистанції та швидкості. Робоче середовище PTV Vissim зображено на рисунку 2.2.

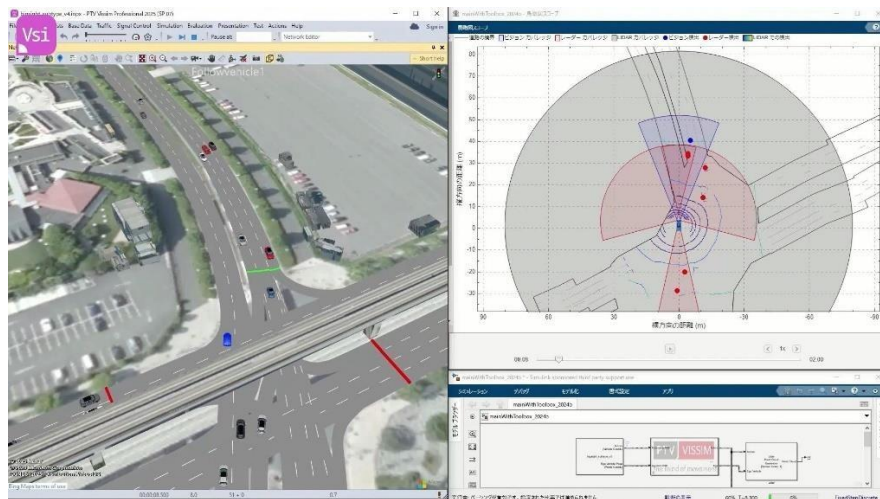


Рисунок 2.2 — Робоче середовище PTV Vissim

Переваги: висока деталізація поведінки водія завдяки психофізичній моделі; професійні інструменти аналізу та звітності; зручний графічний редактор сцен; підтримка складних мультимодальних сценаріїв (автомобілі, громадський транспорт, пішоходи).

Недоліки: пропрієтарне платне рішення із закритими алгоритмами; висока вартість ліцензії; орієнтація на проектування інфраструктури, а не на обробку наявних GPS-треків; відсутність генерації розширеного датасету.

MovingPandas — відкрита бібліотека мовою Python для аналізу траєкторій рухомих об'єктів, побудована на основі GeoPandas [6]. Вона надає структури даних і методи для очищення, сегментації, узагальнення та візуалізації GPS-треків, що дозволяє обчислювати кінематичні характеристики руху та виявляти патерни поведінки. Візуалізацію траєкторій засобами MovingPandas зображено на рисунку 2.3.

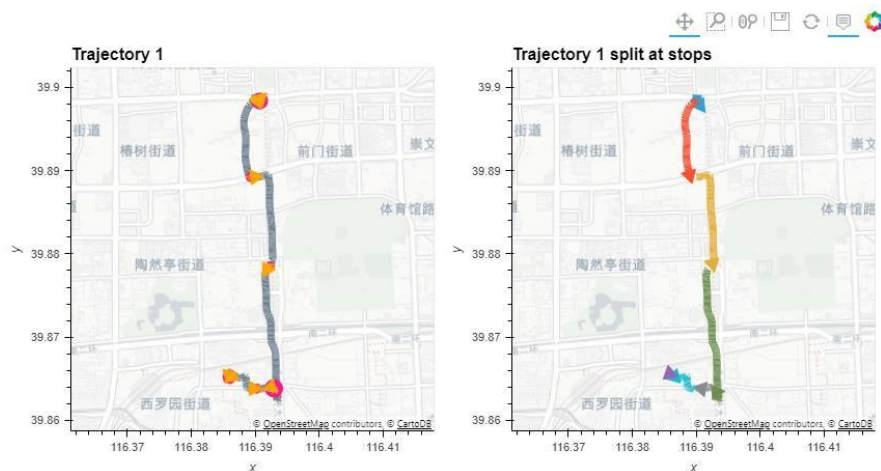


Рисунок 2.3 — Візуалізація траєкторій засобами MovingPandas

Переваги: відкритий код і безкоштовність; гнучка програмна обробка реальних GPS-даних; інтеграція з екосистемою наукового Python (pandas, GeoPandas, scikit-learn); придатність для дослідницького аналізу траєкторій.

Недоліки: орієнтація на розробників і вимога навичок програмування; відсутність готового вебінтерфейсу для швидкого аналізу; немає вбудованої класифікації стилю водіння; не передбачає моделювання впливу дорожніх умов та генерації синтетичних даних.

Geotab — комерційна телематична платформа для управління автопарками, що збирає дані з бортових пристроїв (OBD-II) та оцінює поведінку водіїв за подіями різкого гальмування, прискорення, перевищення швидкості й агресивного маневрування [7]. Результати подаються у вигляді інтерактивних інформаційних панелей та звітів про безпеку. Інформаційну панель аналізу поведінки водія Geotab зображено на рисунку 2.4.

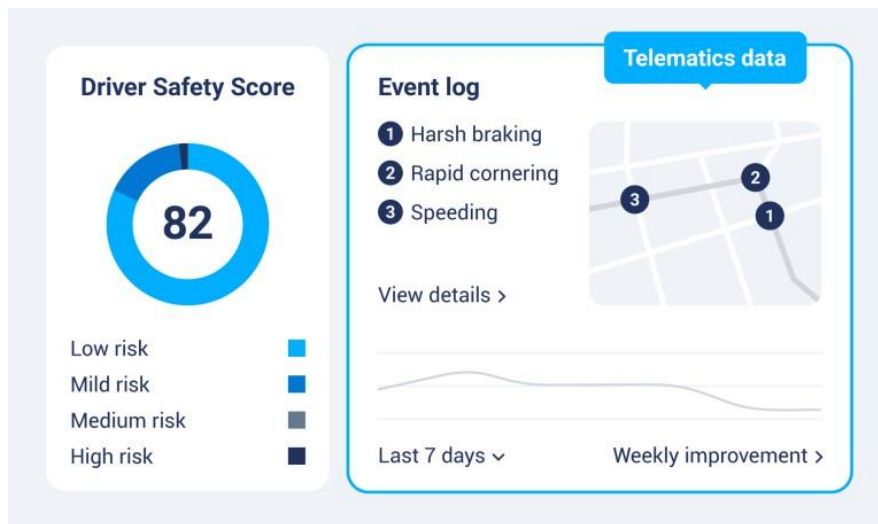


Рисунок 2.4 — Інформаційна панель аналізу поведінки водія Geotab

Переваги: аналіз поведінки водія у реальному часі; зручні інформаційні панелі та звіти; точність завдяки сигналам з апаратних сенсорів (акселерометр, OBD); зрілий промисловий продукт із технічною підтримкою.

Недоліки: платне закрите SaaS-рішення; потреба у спеціалізованому обладнанні у кожному транспортному засобі; недоступність первинних алгоритмів класифікації, що унеможливорює їх контроль в академічних дослідженнях; неможливість роботи з уже наявними історичними GPS-датасетами без власних пристроїв.

Узагальнюючи, у напрямі моделювання (SUMO, PTV Vissim) інструменти зосереджені на генерації синтетичного руху «з нуля» та потребують або складного калібрування, або дорогих ліцензій, а в напрямі аналізу поведінки (MovingPandas, Geotab) рішення є або низькорівневими бібліотеками без вебінтерфейсу, або закритими комерційними платформами з прив'язкою до обладнання. Таким чином, у ніші доступних, готових до використання вебінструментів, що поєднують обробку реальних GPS-даних, класифікацію стилю водіння та генерацію розширеного датасету, наявна прогалина, яку й заповнює розроблений застосунок.

2.3 Огляд датасету T-Drive

Як джерело реальних транспортних даних у роботі використано відкритий набір T-Drive — один із найвідоміших еталонних датасетів GPS-траєкторій у сфері транспортних досліджень [12, 21]. Набір був зібраний компанією Microsoft Research у Пекіні впродовж тижневого періоду спостережень і містить записи переміщень понад 10 000 таксі [1]. Загальний обсяг датасету перевищує 15 мільйонів GPS-точок, що відповідає приблизно 9 мільйонам кілометрів сукупного пробігу транспортних засобів [1].

Кожен рядок файлу даних має чітко визначену структуру з чотирьох полів: ідентифікатор таксі, мітка часу у форматі ISO 8601, довгота та широта у десяткових градусах. Така мінімалістична схема є водночас перевагою (компактність, простота парсингу) та обмеженням, оскільки не містить додаткових сенсорних показників. Середня дискретизація записів становить приблизно 177 секунд, однак фактичний інтервал між сусідніми точками суттєво варіюється — від кількох секунд до декількох хвилин — залежно від умов прийому супутникового сигналу, щільності міської забудови та поточного стану транспортного засобу [2].

Датасет T-Drive охоплює як щільну міську забудову центральної частини Пекіна, так і менш завантажені приміські райони, що забезпечує природне різноманіття умов руху — від повільного руху у заторах до вільного руху магістралями. Велика кількість унікальних водіїв у наборі дозволяє статистично достовірно вивчати індивідуальні особливості стилю керування, а відкрита ліцензія Microsoft Research надає можливість використовувати дані для академічних і некомерційних досліджень без додаткових обмежень [2].

Разом із тим набір має низку обмежень, які безпосередньо вплинули на проєктні рішення цієї роботи. Відсутність OBD-даних (оберти двигуна, миттєва витрата палива, положення дросельної заслінки) унеможливорює пряме вимірювання енергетичних показників руху, тому відповідні характеристики доводиться оцінювати опосередковано через кінематику. Нерівномірна

дискретизація ускладнює коректне обчислення похідних величин, насамперед прискорення, і вимагає додаткової фільтрації аномальних сегментів. Нарешті, географічна прив'язка виключно до Пекіна обмежує кліматичне різноманіття вибірки — саме цю прогалину покликаний компенсувати модуль симуляції дорожніх умов, що штучно відтворює сценарії дощу, снігу та ожеледиці, відсутні у вихідних даних [1].

2.4 Алгоритм класифікації стилю водіння

Аналіз стилю водіння є активним напрямом досліджень, що поєднує цифрову обробку сигналів, машинне навчання та транспортну інженерію [11]. Традиційно розрізняють три основні підходи до розв'язання цієї задачі: на основі порогових правил, статистичний та заснований на машинному навчанні.

Підхід на основі порогових правил полягає у визначенні екстремальних подій — різкого гальмування (прискорення менше -2 м/с^2), різкого прискорення (понад $+2 \text{ м/с}^2$) та різкого повороту (кут понад 45°) — і подальшому підрахунку частоти їх виникнення. Такий підхід простий у реалізації та інтерпретації, проте не враховує контекст: наприклад, різке гальмування у заторі має іншу природу, ніж аналогічне гальмування на вільній магістралі [11]. Статистичний підхід ґрунтується на розрахунку агрегованих характеристик поїздки — середньої швидкості, коефіцієнта варіації швидкості як міри нерівномірності руху та інтенсивності маневрів; ці ознаки нормалізуються та порівнюються між водіями або між поїздками одного водія за різних умов [11]. Підхід на основі машинного навчання передбачає навчання класифікатора на розмічених даних і здатний враховувати складні нелінійні залежності [16–18], проте вимагає значного обсягу попередньо розмічених прикладів, дефіцит яких і покликана компенсувати ця робота.

У розробленому застосунку класифікацію стилю водіння реалізовано як мультикритеріальну систему оцінювання. Для кожного водія обчислюється вектор ознак, на основі якого за сукупністю порогових умов визначається клас.

Система ознак включає такі показники: частка різкого гальмування (`sharpBrakingRate`) — відсоток сегментів із прискоренням менше -2 м/с^2 ; частка різкого розгону (`hardAccelRate`) — відсоток сегментів із прискоренням понад $+2 \text{ м/с}^2$; частка різких поворотів (`sharpTurnRate`) — відсоток сегментів із кутом повороту понад 45° ; коефіцієнт варіації швидкості (`speedCV`) — відношення стандартного відхилення швидкості до середньої швидкості у відсотках; частка руху на високій швидкості (`highSpeedRate`) — відсоток сегментів зі швидкістю понад 60 км/год .

Правило класифікації ґрунтується на підрахунку кількості сигналів агресивного та еко-водіння. Водія віднесено до агресивного класу, якщо виконуються три або більше з умов: `sharpBrakingRate` понад 5% , `hardAccelRate` понад 5% , `sharpTurnRate` понад 10% , `speedCV` понад 40% , `highSpeedRate` понад 30% . Водія віднесено до еко-класу, якщо одночасно `sharpBrakingRate` менше 2% , `hardAccelRate` менше 2% , `sharpTurnRate` менше 3% , `speedCV` менше 20% та `highSpeedRate` менше 10% . За наявності однієї або двох агресивних умов та середньої швидкості понад 15 км/год водія класифіковано як нормального, інакше — як обережного. Пороги екстремальних маневрів ($\pm 2 \text{ м/с}^2$, 45°) узгоджено з типовими значеннями, які наводяться у дослідженнях стилю керування [11, 20].

Висновки до розділу 2

У другому розділі досліджено предметну область роботи. Проаналізовано відкритий датасет T-Drive як джерело реальних GPS-траєкторій, описано його структуру, обсяг та обмеження, зокрема відсутність OBD-показників і нерівномірність дискретизації, що обґрунтовує потребу у модулі симуляції дорожніх умов.

Виконано порівняльний огляд чотирьох представницьких аналогів — SUMO, PTV Vissim, MovingPandas та Geotab — і встановлено, що жоден із них не поєднує одночасно обробку реальних GPS-даних, класифікацію стилю

водіння та генерацію розширеного датасету у доступному вебінтерфейсі, що підтверджує актуальність розробки.

Формалізовано алгоритм обробки GPS-даних: наведено формули обчислення відстані за Гаверсином, швидкості та прискорення з відповідними фізичними обмеженнями, а також правило визначення кута повороту. Описано систему класифікації стилю водіння за чотирма класами та модель застосування коефіцієнтів дорожніх умов.

3 ЗАСОБИ РОЗРОБКИ

3.1 Вибір мови програмування та середовища виконання

Мовою програмування для реалізації клієнтської частини системи обрано JavaScript стандарту ES2020+, який використовується без залучення додаткових фреймворків чи бібліотек. Такий вибір зумовлений фундаментальною особливістю мови: JavaScript є єдиною мовою програмування високого рівня, що нативно підтримується сучасними веббраузерами та виконується безпосередньо у середовищі користувача без попередньої компіляції або встановлення додаткових плагінів.

Середовищем виконання застосунку є рушій JavaScript сучасного веббраузера (V8 у Google Chrome, SpiderMonkey у Mozilla Firefox тощо), що працює повністю на стороні клієнта. Уся обробка даних — від зчитування файлів до генерації розширеного датасету — відбувається в оперативній пам'яті браузера, без передавання інформації на віддалений сервер, що гарантує конфіденційність даних та працездатність застосунку навіть за відсутності мережевого з'єднання. Середовищем розробки обрано редактор Visual Studio Code, який забезпечує підсвічування синтаксису, статичний аналіз коду, інтегрований термінал та зручну роботу з системою контролю версій. Запуск і налагодження застосунку виконуються безпосередньо у браузері із застосуванням вбудованих інструментів розробника (DevTools).

Інтеграція JavaScript зі стандартизованими браузерними API — File API, Blob API та URL API — формує потужний інструментарій для локальної роботи з даними. File API дозволяє безпечно отримувати доступ до файлів користувача через елемент `<input type="file">` або механізм drag-and-drop, зчитуючи текстові дані безпосередньо в пам'ять браузера. Blob API надає можливість конструювати бінарні об'єкти довільної структури, що є основою для формування експортованих файлів у форматах CSV або JSON. URL API, зокрема метод `URL.createObjectURL()`, генерує тимчасові посилання на ці об'єкти та ініціює

завантаження файлу на диск користувача через програмний клік по динамічно створеному елементу `<a>`. Така комбінація технологій реалізує повний цикл обробки даних виключно на стороні клієнта [22].

Використання сучасного стандарту ECMAScript 2020 (ES11) та новіших специфікацій надає доступ до синтаксичних конструкцій, які суттєво підвищують надійність і читабельність коду при обробці складних структур даних [22]. Оператор опціонального ланцюжка (`?.`) та оператор нульового злиття (`??`), запроваджені саме у редакції ECMAScript 2020, активно застосовуються для безпечної навігації вкладеними об'єктами та масивами, запобігаючи аварійному завершенню скрипта у разі відсутності очікуваних властивостей у некоректних або пошкоджених GPS-записах [22]. Деструктуризація масивів спрощує розпакування рядків датасету, дозволяючи стисло присвоювати змінним значення координат, часових міток та ідентифікаторів без громіздких індексних звернень. Функціональні методи масивів `.map()`, `.filter()` та `.reduce()` використовуються для декларативної трансформації, фільтрації та агрегації траєкторних даних, що робить код виразнішим і менш схильним до побічних ефектів порівняно з імперативними циклами. Нарешті, конструкція `async/await` забезпечує управління асинхронними операціями, зокрема зчитуванням великих файлів через `FileReader`, перетворюючи складну логіку зворотних викликів на лінійний, легко супроводжуваний потік виконання, що є критично важливим для підтримання чутливості інтерфейсу при обробці десятків тисяч записів.

3.2 Бібліотека візуалізації Chart.js

Для візуалізації аналітичних даних у застосунку обрано бібліотеку `Chart.js`, яка підключається безпосередньо через CDN без необхідності локального встановлення або збірки залежностей [23]. Такий підхід забезпечує миттєву доступність інструментарію візуалізації при завантаженні сторінки, мінімізує початковий розмір застосунку та спрощує розгортання. Архітектура `Chart.js` базується на декларативному підході: конфігурація кожного графіка описується

єдиним об'єктом JavaScript, що містить визначення типу діаграми, набори даних, параметри осей, легенди, підказок та анімацій [23].

Динамічне оновлення візуалізації в реальному часі виконується шляхом модифікації відповідних властивостей об'єкта `data` або `options` з подальшим викликом методу `chart.update()`, який ефективно перемальовує лише змінені елементи полотна `<canvas>` без повного знищення та повторного створення екземпляра. Це забезпечує плавну анімацію переходів, високу продуктивність навіть при частій зміні даних та економію обчислювальних ресурсів клієнтського пристрою [23].

У межах застосунку реалізовано чотири специфічні типи діаграм, кожен з яких обрано відповідно до семантики відображуваних даних. Кільцева (doughnut) діаграма використовується для візуалізації розподілу стилів водіння серед ідентифікованих учасників руху, де кожен сектор відповідає частці водіїв певного класу («еко», «обережний», «нормальний», «агресивний»). Стовпчаста (bar) діаграма застосовується у двох контекстах: для побудови гістограми розподілу миттєвих швидкостей із дванадцятьма бінами в діапазоні 0–120 км/год та для порівняльної панелі, де зіставляються агреговані метрики реальних і симульованих даних. Радарна діаграма відображає шість нормованих кінематичних показників руху, що дозволяє наочно порівнювати реальний та симульований профілі. Точкова (scatter) діаграма розсіювання використовується для аналізу двовимірного розподілу пар значень «швидкість — прискорення», де колір маркера кодує приналежність точки до певного класу водіння; така візуалізація розкриває кореляційні зв'язки між кінематичними величинами та дозволяє візуально підтвердити коректність класифікації.

3.3 Стилзація та формати даних

Архітектуру стилізації інтерфейсу реалізовано засобами нативного CSS без залучення зовнішніх препроцесорів (Sass, Less) чи компонентних фреймворків (Bootstrap, Tailwind). Усі тематичні кольори, тіні, радіуси

скруглення та типографічні параметри централізовано у псевдокласі `:root` через механізм `CSS Custom Properties`. Такий підхід забезпечує декларативне керування темою оформлення: перехід між світлою та темною темами реалізується шляхом динамічної зміни значень змінних у кореновому контексті документа, що усуває потребу в додаткових інструментах збірки чи дублюванні стилів. Адаптивна верстка побудована на сучасних модулях `CSS Grid` та `Flexbox`, які дозволяють гнучко керувати розкладкою елементів.

Критичною точкою перелому встановлено ширину `768 px`: при звуженні видимої області нижче цього значення сітка автоматично перебудовується з багатостовпцевої на одноколонкову, інтерактивні елементи збільшують області дотику, а навігація трансформується у компактний мобільний формат, що гарантує коректне відображення та зручну експлуатацію на смартфонах і планшетах.

Проміжні результати обчислень та стан застосунку зберігаються безпосередньо в оперативній пам'яті браузера у вигляді нативних об'єктів і масивів `JavaScript`. Таке архітектурне рішення свідомо уникає зайвих циклів серіалізації та десеріалізації (наприклад, перетворення даних у `JSON`-рядки чи збереження у `localStorage` під час активної обробки), що суттєво знижує навантаження на процесор і мінімізує затримки введення-виведення. Серіалізація застосовується лише на фінальному етапі — під час експорту готового датасету у формати `CSV` та `JSON`.

Висновки до розділу 3

У третьому розділі обґрунтовано вибір засобів розробки. Як мову програмування обрано `JavaScript` стандарту `ES2020+`, що нативно виконується у браузері без серверної частини; визначено середовище виконання (рушій браузера) та середовище розробки (`Visual Studio Code`). Показано, що сучасні синтаксичні конструкції `ECMAScript 2020`, зокрема оператори опціонального

ланцюжка та нульового злиття, підвищують стійкість коду до некоректних вхідних даних.

Обґрунтовано вибір бібліотеки Chart.js для візуалізації та описано чотири типи діаграм — кільцеву, стовпчасту, радарну й точкову, — кожен з яких узгоджено із семантикою відповідних даних і з реалізованим кодом. Описано підхід до стилізації інтерфейсу засобами нативного CSS з адаптивною версткою, а також стратегію роботи з даними в оперативній пам'яті із серіалізацією лише на етапі експорту.

4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

4.1 Архітектура застосунку

Архітектуру застосунку організовано за принципом модульного розділення, реалізованого засобами функціонального програмування на JavaScript та централізованого управління станом.

Компонентну архітектуру застосунку, що відображає взаємозв'язки між п'ятьма модулями та глобальним станом, наведено на рисунку 4.1.

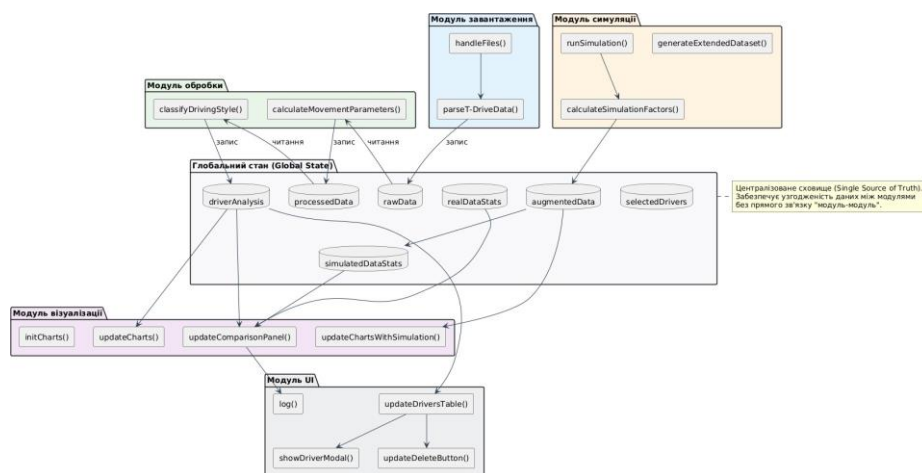


Рисунок 4.1 — Компонентна архітектура застосунку

Глобальний стан застосунку представлено набором змінних верхнього рівня, структурованих за семантикою та етапом обробки даних. Масив `rawData` акумулює всі розпарсені GPS-точки у вигляді об'єктів `{taxiId, timestamp, longitude, latitude}`, зберігаючи вихідну телеметрію без трансформацій. Масив `processedData` містить результати попередньої обробки — сегменти між послідовними точками з обчисленими кінематичними параметрами `{taxiId, speedKmh, speedMs, acceleration, turnAngle, distance, timeDiff}`. Агреговані показники по кожному учаснику руху зберігаються в об'єкті `driverAnalysis`, реалізованому як словник `{taxiId: {avgSpeed, avgAccel, drivingStyle, ...}}` для швидкого доступу за унікальним ідентифікатором. Для ізоляції експериментальних даних від оригіналу симульовані записи накопичуються в окремому масиві `augmentedData`, що гарантує незмінність реальних вимірювань

під час багаторазових запусків симуляції. Множина `selectedDrivers` відстежує ідентифікатори водіїв, відмічених користувачем для видалення, а об'єкти `realDataStats` і `simulatedDataStats` містять порівняльну агрегацію метрик для аналітичної панелі.

Функціональну структуру застосунку систематизовано у п'ять взаємопов'язаних модулів, кожен з яких інкапсулює визначений набір операцій та має чітко визначені точки входу й виходу:

- перший модуль — модуль завантаження та імпорту — реалізує функції `handleFiles()` і `parseTDriveData()`, що відповідають за ініціалізацію `FileReader` API, асинхронне зчитування текстових файлів, валідацію структури рядків та перетворення їх у типізовані об'єкти телеметрії;
- другий модуль — модуль кінематичної обробки — включає функції `calculateMovementParameters()` і `classifyDrivingStyle()`: перша обчислює прискорювальні характеристики та кути повороту на основі часових і просторових інтервалів, друга застосовує порогові правила для присвоєння кожному водієві категорії стилю керування;
- третій модуль — модуль симуляції та розширення датасету — містить функції `runSimulation()`, `calculateSimulationFactors()` і `generateExtendedDataset()` та відповідає за сценарне моделювання зовнішніх умов, розрахунок коригувальних коефіцієнтів, генерацію варіацій кінематики й формування збалансованого розширеного набору даних;
- четвертий модуль — модуль візуалізації — об'єднує функції `initCharts()`, `updateCharts()`, `updateChartsWithSimulation()` і `updateComparisonPanel()`, що забезпечують ініціалізацію графічних компонентів, реактивне оновлення діаграм, накладення симульованих серій на реальні дані та побудову порівняльних метрик;
- п'ятий модуль — модуль інтерфейсу користувача (UI) — реалізує функції `updateDriversTable()`, `showDriverModal()`, `updateDeleteButton()` і `log()` та відповідає за відображення табличних даних, керування модальними

вікнами детальної статистики, активацію елементів керування й виведення системних повідомлень про прогрес операцій.

Взаємодію користувача з описаними архітектурними компонентами формалізовано у вигляді діаграми варіантів використання (Use-case), наведеної на рисунку 4.2. Діаграма відображає ключові сценарії експлуатації застосунку: від імпорту сирих логів, конфігурування параметрів симуляції та запуску обчислювальних конвеєрів до інтерактивного порівняння результатів, управління вибіркою водіїв та експорту фінальних датасетів.

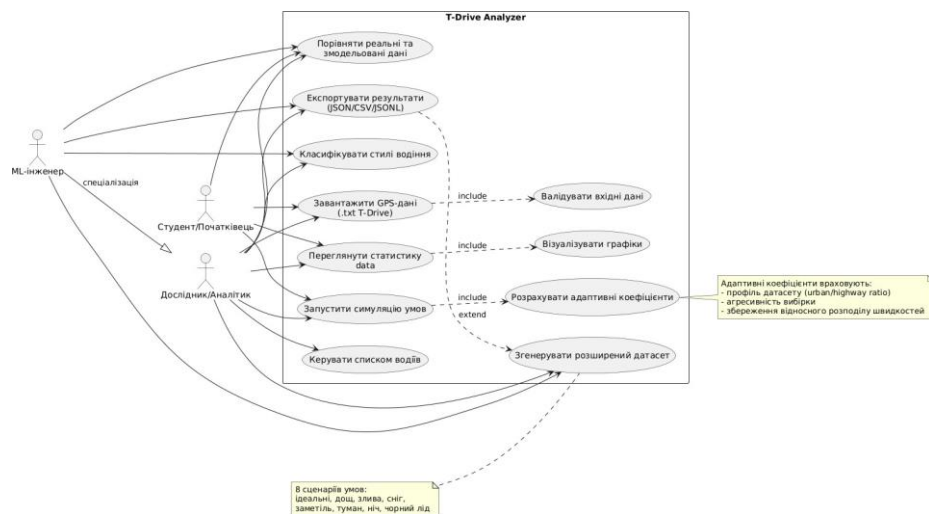


Рисунок 4.2 — Use-case діаграма застосунку

Послідовність перетворення даних від завантаження файлу до візуалізації та експорту унаочнює конвеєр обробки й аналізу даних, наведений на рисунку 4.3.

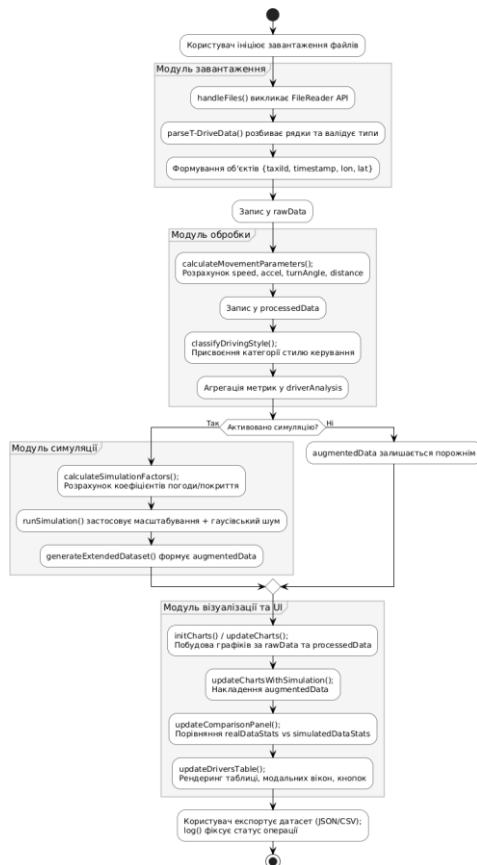


Рисунок 4.3 — Конвеєр обробки та аналізу даних

Динаміку викликів між користувачем, модулями та глобальним станом під час типового сценарію завантаження й обробки відображає діаграма послідовності, наведена на рисунку 4.4.

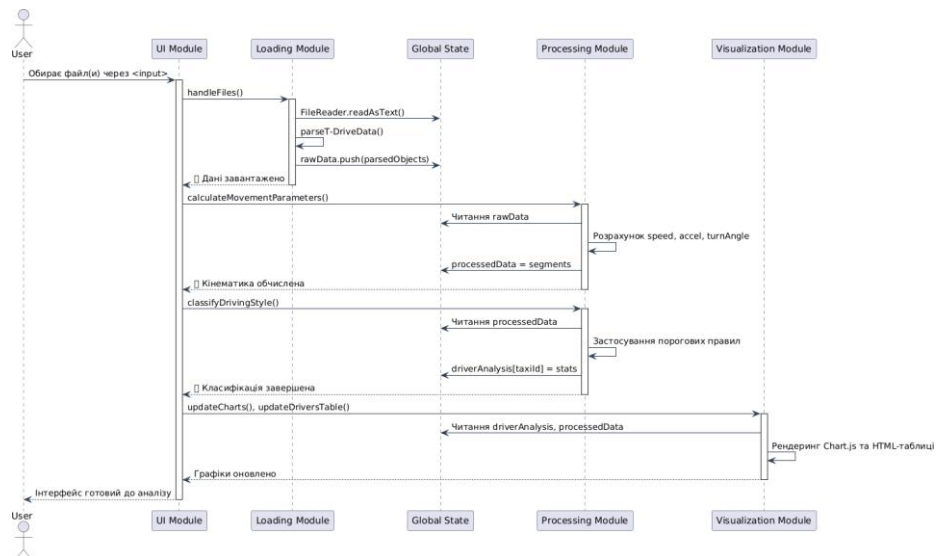


Рисунок 4.4 — Послідовність взаємодії при завантаженні та обробці (Sequence Diagram)

4.2 Модуль завантаження та парсингу даних

Завантаження файлів у системі реалізовано за допомогою двох взаємодоповнюючих механізмів, що забезпечують гнучкість інтерфейсу та зручність користувача. Перший підхід базується на класичній взаємодії з нативним елементом `<input type="file">`, прихованим у DOM-дереві та активованим при кліку по спеціальній зоні завантаження; атрибут `multiple` дозволяє одночасно обирати кілька файлів. Другий механізм реалізує технологію `drag-and-drop`, яка перехоплює події `dragover` та `drop` на тому ж контейнері: під час перетягування система блокує стандартну поведінку браузера (`preventDefault`), а при скиданні файлів витягує масив об'єктів `event.dataTransfer.files` та передає його в єдиний конвеєр обробки. Така архітектура гарантує, що незалежно від обраного способу введення дані надходять на наступний етап у стандартизованому форматі.

Безпосереднє зчитування вмісту файлів здійснюється у функції `handleFiles()`, яка ітерує по кожному файлу та ініціює асинхронне читання через `FileReader.readAsText()` з явним вказанням кодування UTF-8. Після успішного завершення читання викликається подія `onload`, де результат передається модулю `parseTDriveData()`. Архітектурною особливістю є підтримка накопичення даних: замість перезапису масиву `rawData` результат парсингу кожного нового файлу додається до попереднього вмісту за допомогою `spread`-оператора (...). Це дозволяє користувачеві послідовно завантажувати лог-файли різних водіїв протягом однієї сесії, формуючи єдиний об'єднаний датасет без втрати раніше обробленої інформації.

Парсинг сирих текстових даних реалізовано з акцентом на валідацію та стійкість до структурних аномалій. Кожен рядок проходить попередню обробку — обрізку зайвих пробілів і перевірку на порожність. Після розбиття за символом коми система перевіряє наявність щонайменше чотирьох полів, що відповідає очікуваній структурі запису. Далі виконується явне перетворення типів: `parseInt` для ідентифікатора водія, конструктор `new Date()` для часової

мітки та `parseFloat` для координат; коректність перетворень контролюється перевіркою на `NaN`, а валідність дати — викликом `isNaN(date.getTime())`. У разі виявлення будь-якої невідповідності (рядок-заголовок, порожній чи пошкоджений запис, некоректний формат координат) система ігнорує поточний рядок і переходить до наступного, не перериваючи процес імпорту. Узагальнену структуру модуля завантаження та парсингу даних наведено на рисунку 4.5.



Рисунок 4.5 — Схема модуля завантаження та парсингу даних

Як показано на схемі, процес опрацювання вхідних даних організовано у вигляді послідовного конвеєра: зчитування файлу, попередня обробка та розбиття рядків, валідація структури запису, перетворення типів і відсіювання некоректних значень. Лише записи, що успішно пройшли всі етапи перевірки, потрапляють до внутрішніх структур даних застосунку й використовуються на подальших кроках обчислень. Такий підхід забезпечує стійкість модуля до пошкоджених та неповних вхідних даних і гарантує цілісність набору, на основі якого виконується подальше моделювання дорожніх умов.

4.3 Модуль симуляції та розширення

Модуль симуляції транспортних даних розроблено з урахуванням двох фундаментальних сценаріїв моделювання. Перший сценарій передбачає симуляцію змінених умов для вже наявних GPS-треків: базова траєкторія зберігає просторову структуру, проте кінематичні параметри модифікуються відповідно до заданих коефіцієнтів, що імітує вплив зовнішніх факторів на динаміку руху. Другий сценарій орієнтований на генерацію повністю нових синтетичних треків «з нуля», що забезпечує масштабування вибірки та моделювання рідкісних або екстремальних сценаріїв поведінки, відсутніх в оригінальному датасеті.

Алгоритмічне ядро модуля реалізує модифікацію кожної точки траєкторії через функцію симуляції, яка застосовує мультиплікативне масштабування та адитивний шум до кінематичних величин. Швидкість руху множиться на коефіцієнт `speedFactor`, після чого до результату додається гаусівський шум зі стандартним відхиленням $\sigma = 2$ км/год; аналогічно прискорення масштабується через `accelFactor` та підсилюється шумом із $\sigma = 0,1$ м/с². Впровадження випадкової складової імітує природні флуктуації реальних телематичних вимірювань і запобігає виникненню артефактів точного лінійного масштабування, що могли б спотворити статистичні розподіли.

Узагальнену структуру модуля симуляції та розширення наведено на рисунку 4.6.

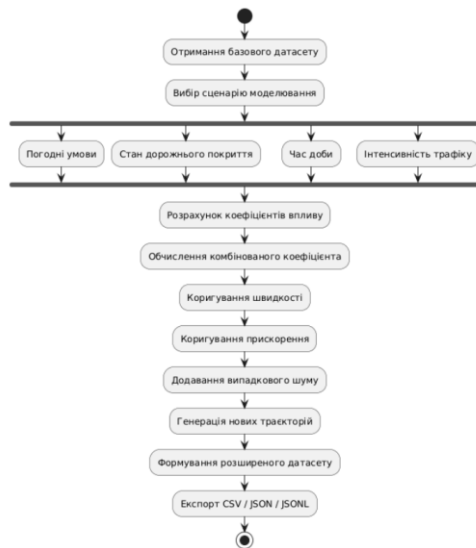


Рисунок 4.6 — Схема модуля симуляції та розширення

Функція `generateExtendedDataset()` інтегрує реальні та симульовані записи у єдину структуру, збагачуючи кожен запис контекстними метаданими зовнішніх умов: `weather` (погодні умови), `surface` (тип покриття), `timeOfDay` (час доби), `precipitation` (інтенсивність опадів), `trafficLoad` (завантаженість трафіку) та `visibility` (видимість). Паралельно виконується інженерія ознак: обчислюється нормована швидкість ($\text{speedNorm} = \text{speedKmh} / 120$) та нормоване прискорення ($\text{accelNorm} = (\text{acceleration} + 5) / 10$), а також формуються бінарні індикатори `sharpBraking` і `sharpAcceleration`, що фіксують екстремальні маневри. Контроль обсягу вихідного датасету реалізовано через інтерактивний повзунок у діапазоні від 1000 до 10 000 записів; за нестачі даних система застосовує рівномірну вибірку з повторенням (`sampling with replacement`) зі збереженням відносної частки реальних і синтетичних записів.

4.4 Модуль візуалізації та порівняння

Візуалізацію реалізовано через чотири графіки, що оновлюються після кожної операції завантаження або симуляції. Кругова діаграма стилів водіння відображає розподіл водіїв за класами «агресивний / нормальний / обережний / еко»; зберігається одна інстанція `Chart.js`, що оновлюється через присвоєння `chart.data.datasets[0].data` та виклик `chart.update()` без повторного створення, чим

уникається мерехтіння. Стовпчаста діаграма розподілу швидкостей будується як гістограма з дванадцятьма бінами від 0 до 120 км/год із кроком 10 км/год; підрахунок сегментів у кожному біні виконується через `Math.floor(speedKmh / 10)`, а при симуляції на той самий графік накладається другий, напівпрозорий, датасет для порівняння.

Радарна діаграма умов руху відображає шість показників — середню швидкість, максимальну швидкість, середнє прискорення, частку різких гальмувань, частку різких прискорень та частку різких поворотів, — нормованих до діапазону $[0; 1]$ для забезпечення порівнянності; при симуляції відображаються два набори даних (реальний та симульований). Точкова діаграма профілю прискорень будує пари «швидкість — прискорення» для вибірових точок (максимум 500 для продуктивності), де колір точки кодує клас стилю водіння, що дозволяє візуально підтвердити коректність класифікації.

Панель порівняння реальних і симульованих даних реалізована у двох режимах перегляду, між якими відбувається перемикання без перезавантаження: режим «паралельно» відображає дві колонки числових показників та блок відхилень, а режим «накладання» виводить три додаткові графіки (розподіл швидкостей, профіль прискорень та радарну діаграму) з двома наборами даних на кожному. Таблиця водіїв оновлюється функцією `updateDriversTable()` після кожної операції та підтримує текстовий пошук за ідентифікатором, чекбокси групового вибору, кнопку деталей (відкриває модальне вікно) і кнопку видалення окремого водія з повним перерахунком статистики. Узагальнену структуру модуля візуалізації та порівняння наведено на рисунку 4.7.

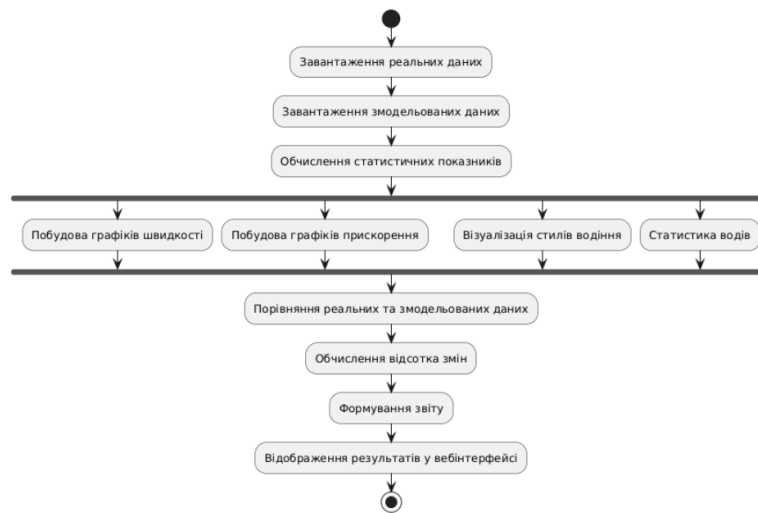


Рисунок 4.7 — Схема модуля візуалізації та порівняння

Потік даних організовано лінійно: від завантаження файлів T-Drive через парсинг до розрахунку кінематичних параметрів, класифікації стилів, адаптивної симуляції та фінального експорту. Управління станом здійснюється через централізовані глобальні змінні модуля (`rawData`, `processedData`, `augmentedData`, `driverAnalysis`), що забезпечує синхронізацію між компонентами. Для оптимізації продуктивності реалізовано потоковий парсинг з індикацією прогресу, обмеження кількості точок для точкових графіків, пакетне оновлення DOM через `DocumentFragment` та уникнення повторного створення екземплярів `Chart.js` шляхом оновлення лише даних.

Висновки до розділу 4

У четвертому розділі описано програмну реалізацію застосунку. Подано організацію глобального стану через набір типізованих змінних верхнього рівня та обґрунтовано модульне розділення відповідальності між п'ятьма взаємопов'язаними модулями — завантаження та імпорту, кінематичної обробки, симуляції та розширення, візуалізації й інтерфейсу користувача. Наведене компонентну діаграму, Use-case діаграму, конвеєр обробки даних та діаграму послідовності.

Детально розглянуто реалізацію модуля завантаження та парсингу даних з механізмом введення та стійким до аномалій парсингом; модуля симуляції та розширення; а також модуля візуалізації та порівняння з чотирма типами діаграм і двома режимами порівняльної панелі.

5 ОПИС РОБОТИ КОРИСТУВАЧА ІЗ СИСТЕМОЮ

5.1 Системні вимоги та налаштування

Застосунок реалізовано як вебдодаток. Для роботи достатньо сучасного веббраузера з підтримкою стандарту ECMAScript 2020 та увімкненим виконанням JavaScript — Google Chrome версії 90 і новіших, Mozilla Firefox версії 88 і новіших, Microsoft Edge або іншого браузера на основі рушія Chromium. Уся обробка даних відбувається локально в оперативній пам'яті браузера.

Апаратні вимоги визначаються переважно обсягом вхідного датасету. Для комфортної роботи з файлами обсягом до кількох сотень тисяч GPS-точок достатньо персонального комп'ютера або ноутбука з 4 ГБ оперативної пам'яті та двоядерним процесором; для обробки великих наборів даних рекомендовано 8 ГБ оперативної пам'яті та більше. Окремого графічного прискорювача не потрібно, оскільки візуалізація засобами Chart.js ефективно працює на інтегрованих графіці.

Налаштування застосунку зводиться до відкриття HTML-файлу у браузері. Підключення до мережі Інтернет потрібне лише під час першого завантаження сторінки для отримання бібліотеки Chart.js з CDN; після кешування бібліотеки браузером застосунок здатний працювати автономно. Усі параметри симуляції (погодні умови, стан покриття, час доби, рівень опадів та видимість) задаються безпосередньо в інтерфейсі та не вимагають редагування конфігураційних файлів.

5.2 Завантаження даних

Робота із застосунком розпочинається із завантаження вхідних даних формату T-Drive. Користувач може скористатися одним із двох способів: натиснути на зону завантаження та обрати один чи кілька файлів у системному

діалозі або перетягнути файли безпосередньо у цю зону за допомогою механізму drag-and-drop. Обидва способи рівноцінні та передають дані в єдиний конвеєр обробки.

Приклад інтерфейсу завантаження файлів датасету наведено на рисунку 5.1.

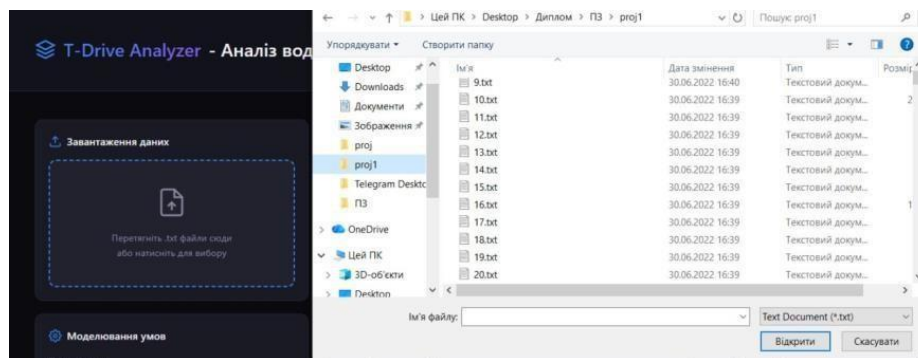


Рисунок 5.1 — Завантаження даних

Після завантаження файлів застосунок автоматично виконує парсинг даних, обчислення кінематичних параметрів та класифікацію стилів водіння. Лог-панель у нижній частині інтерфейсу відображає хід обробки: кількість успішно зчитаних точок, кількість виявлених водіїв та сегментів траєкторій, а також підсумковий звіт. Некоректні рядки (заголовки, порожні чи пошкоджені записи) ігноруються без переривання обробки. Якщо користувач завантажує додаткові файли після початкового завантаження, нові дані додаються до вже наявних, формуючи об'єднаний датасет без втрати попередньо обробленої інформації.

5.3 Перегляд статистики та графіків

Після успішного завантаження та первинної обробки вхідних даних центральна робоча область інтерфейсу трансформується у багаторівневу панель аналітичної візуалізації, що містить чотири інтерактивні графічні компоненти, кожен з яких вирішує специфічне завдання інтерпретації транспортних даних. Така компоновка забезпечує одночасний огляд різних аспектів поведінки водіїв

і дозволяє досліднику формувати цілісне уявлення про структуру датасету без перемикання між окремими вікнами.

Кругова діаграма розподілу стилів водіння візуалізує пропорційне співвідношення ідентифікованих водіїв за чотирма класами; кольорову схему підібрано з урахуванням семантики класів. Гістограма розподілу швидкостей відображає частоту потрапляння сегментів у певні діапазони швидкості. Нижче графічної панелі розташована інтерактивна таблиця водіїв зі зведеною статистикою по кожному учаснику руху: унікальний ідентифікатор, кількість точок траєкторії, домінуючий стиль водіння, середню швидкість, максимальне прискорення та інтегральний показник ризику. Поле пошуку фільтрує таблицю в реальному часі за ідентифікатором водія, а кнопка «Деталі» відкриває модальне вікно з розширеною статистикою обраного водія.

Приклад візуалізації графіків та статистики після завантаження тестового датасету наведено на рисунку 5.2, де видно кореляцію між розподілом стилів водіння на круговій діаграмі, профілем швидкостей на гістограмі та багатовимірною структурою на радарній діаграмі.



Рисунок 5.2 — Перегляд графіків та статистики

Натискання кнопки «Деталі» у рядку таблиці відкриває модальне вікно з поглибленою статистикою окремого водія: повним набором кінематичних метрик, профілями швидкості та прискорення, розподілом маневрів за типами та

інтегральними показниками ризику. Приклад такого модального вікна детальної статистики водія наведено на рисунку 5.3.

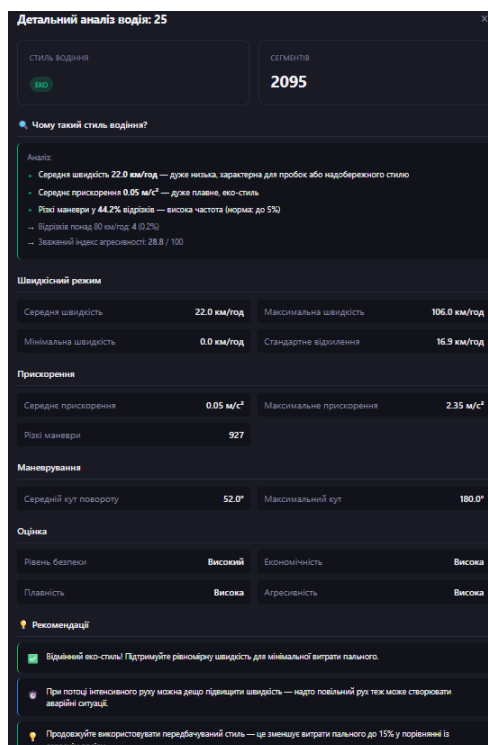


Рисунок 5.3 — Модальне вікно детальної статистики водія

Таке деталізоване подання дозволяє досліднику перейти від узагальненої картини всього датасету до поглибленого аналізу поведінки конкретного учасника руху: виявити характерні моделі розгону й гальмування, оцінити стабільність швидкісного режиму та зіставити індивідуальні показники водія із середніми значеннями по вибірці. Закриття модального вікна повертає користувача до загальної панелі візуалізації без втрати поточного стану таблиці, що забезпечує зручну ітеративну роботу з даними окремих водіїв.

5.4 Симуляція дорожніх умов

Панель симуляції є ключовим модулем застосунку, призначеним для моделювання впливу зовнішніх дорожньо-кліматичних умов на кінематичні характеристики транспортних засобів у межах наявного датасету. Інтерфейс панелі організовано за принципом покрокового вибору параметрів: користувач послідовно задає три категорії зовнішніх умов через елементи керування.

Перша група параметрів стосується погодних умов — користувач обирає один із п'яти сценаріїв: «ясно», «туман», «дощ», «сніг» або «заметіль», — кожен з яких активує відповідний набір коефіцієнтів, що імітують фізичні обмеження руху. Друга група визначає стан дорожнього покриття («суха», «мокра» або «ожеледь») і впливає на параметри тертя та інерції транспортного засобу. Третя група — час доби («день», «світанок», «сутінки», «ніч») — моделює вплив освітленості на видимість і час реакції водія. Додатково передбачено два інтерактивні повзунки: «рівень опадів» (від 0 % до 100 %) та «видимість» (від 10 до 1000 метрів), які градуують інтенсивність явищ і дальність огляду, пропорційно масштабуючи відповідні коефіцієнти впливу на кінематику.

Приклад візуалізації результатів симуляції дорожніх умов наведено на рисунку 5.4, де видно вплив обраних параметрів (наприклад, «дощ + мокра дорога + ніч») на розподіл швидкостей: симульований набір даних демонструє зсув медіани у бік нижчих швидкостей, збільшення дисперсії та зменшення частоти екстремальних прискорень порівняно з базовим реальним датасетом.

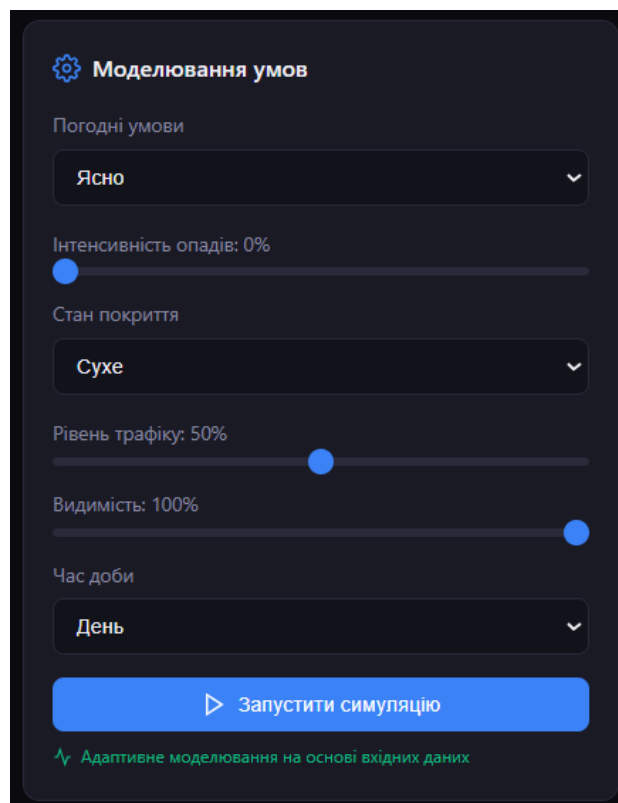


Рисунок 5.4 — Симуляція дорожніх умов

Візуальний зворотний зв'язок реалізовано через автоматичне оновлення графіків у режимі реального часу: на радарній діаграмі умов руху та гістограмі розподілу швидкостей одночасно відображаються обидва набори даних — реальний (базовий) і симульований (модифікований). Накладення двох розподілів на спільні осі дозволяє миттєво оцінити ступінь відхилення, виявити зсув піків, зміну дисперсії або появу нових режимів руху. Приклад оновлення графіків у режимі реального часу під час зміни параметрів симуляції наведено на рисунку 5.5.

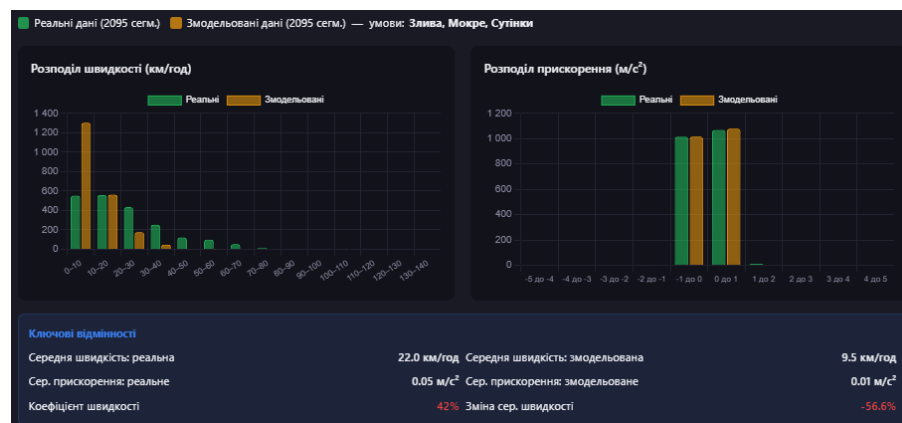


Рисунок 5.5 — Оновлення графіків у режимі реального часу

Завдяки миттєвому перерахунку та перемальовуванню графіків дослідник отримує безперервний інтерактивний зв'язок між заданими параметрами та результуючим розподілом кінематичних характеристик, що дозволяє в режимі експерименту підбирати поєднання погодних умов, стану покриття й часу доби та одразу спостерігати їхній сукупний ефект. Така інтерактивність суттєво прискорює пошук сценаріїв і робить процес налаштування симуляції наочним та зрозумілим навіть без попереднього аналізу числових показників.

5.5 Порівняння реальних та симульованих даних

Панель порівняльного аналізу стає доступною безпосередньо після успішного завершення симуляції та формування розширеного датасету. Цей інтерактивний модуль призначений для валідації якості згенерованих синтетичних даних шляхом їх зіставлення з оригінальними реальними

вимірюваннями, що дозволяє оцінити адекватність математичної моделі та точність алгоритмів генерації.

Режим «паралельно» організує простір відображення у вигляді двох симетричних колонок: ліва містить агреговані числові показники реального датасету, права — відповідні метрики симульованого набору. Для кожного ключового параметра (середня швидкість, дисперсія прискорення, максимальне гальмування, частка агресивних маневрів тощо) відображаються абсолютні значення та статистичні характеристики розподілу, а розташований між колонками блок відхилень автоматично обчислює різницю у відсотковому й абсолютному вираженні. Режим «накладання» забезпечує глибший візуальний аналіз шляхом накладення двох наборів даних на спільні графічні осі, де реальні дані представлено одним кольором, а симульовані — іншим із напівпрозорим заповненням.

Приклад порівняльних даних, отриманих у режимі «накладання», наведено на рисунку 5.6, де видно високий ступінь збігу розподілів швидкостей та прискорень між реальними й симульованими записами.

Порівняння реальних та змодельованих даних		Паралельно		Накладення	
РЕАЛЬНІ ДАНІ		ЗМОДЕЛЬОВАНІ ДАНІ			
Середня швидкість	22.0 км/год	Середня швидкість	7.5 км/год		
Максимальна швидкість	106.0 км/год	Максимальна швидкість	44.2 км/год		
Мінімальна швидкість	0.0 км/год	Мінімальна швидкість	0.0 км/год		
Середнє прискорення	0.05 м/с²	Середнє прискорення	0.00 м/с²		
Максимальне прискорення	2.35 м/с²	Максимальне прискорення	0.24 м/с²		
Сегментів	2095	Коефіцієнт швидкості	33%		
		Умови	Сніг, Лід, Сутінки		
Різниця (змодельовані vs реальні)					
Середня швидкість		-65.7%	Максимальна швидкість	-58.3%	
Середнє прискорення		-90.0%	Загальний вплив	-77.9%	

Рисунок 5.6 — Порівняння реальних та симульованих даних

Незначні відхилення на окремих ділянках графіків перебувають у межах статистичної похибки та не впливають на загальну придатність датасету для навчання класифікаторів стилю водіння. Така візуалізація підтверджує коректність реалізованих алгоритмів симуляції та надає досліднику зручний інструмент для швидкої оцінки якості згенерованих даних.

5.6 Генерація та експорт розширеного датасету

Для отримання фінального розширеного датасету користувач переходить до спеціалізованої панелі генерації. Центральним елементом керування є інтерактивний повзунок, що дозволяє визначати цільовий обсяг вихідної вибірки в діапазоні від 1000 до 10 000 записів. Після встановлення бажаного значення та натискання кнопки «Згенерувати датасет» активується алгоритм формування масиву, який об'єднує реальні GPS-треки із симульованими записами, згенерованими на основі стохастичних моделей кінематичних процесів. Кожен запис автоматично збагачується контекстними метаданими зовнішніх умов (weather, surface, timeOfDay, trafficLoad, visibility) та нормованими ознаками швидкості й прискорення.

Сформований датасет доступний для завантаження у двох універсальних форматах. Кнопка «Експорт JSON» ініціює генерацію файлу у форматі JavaScript Object Notation, який зберігає повну ієрархічну структуру об'єктів і є зручним для архівування експериментів та повторного імпорту. Кнопка «Експорт CSV» формує плоский табличний файл у форматі Comma-Separated Values, де кожен рядок відповідає окремому запису, а стовпці — окремим ознакам. Для забезпечення цілісності даних та зручності керування версіями система автоматично генерує унікальне ім'я файлу з міткою часу у форматі YYYYMMDD_NHMMSS. Приклад інтерфейсу генерації та експорту розширеного датасету наведено на рисунку 5.7.

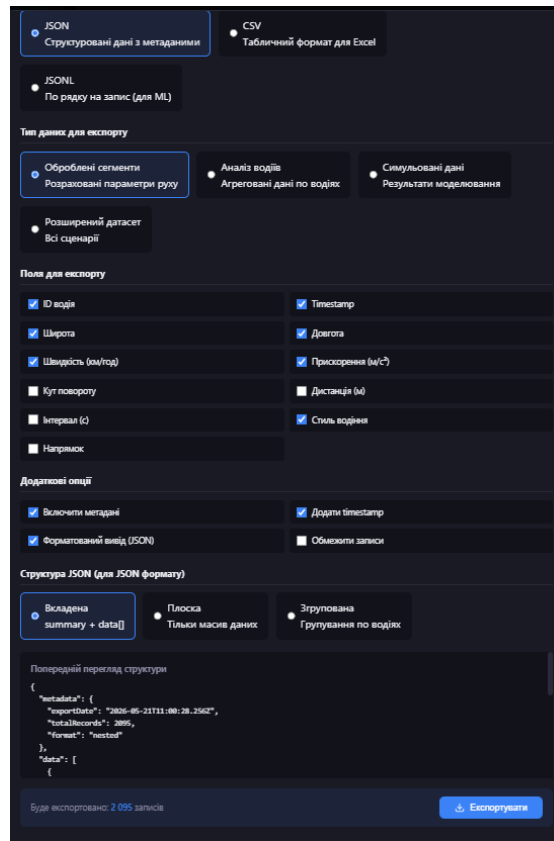


Рисунок 5.7 — Генерація та експорт датасету

Висновки до розділу 5

У п'ятому розділі описано роботу користувача із системою. Сформульовано системні вимоги до застосунку: сучасний веббраузер із підтримкою ECMAScript 2020, локальна обробка даних із гарантією конфіденційності. Поетапно розглянуто типовий сценарій роботи — від завантаження файлів T-Drive двома способами через перегляд статистики та графіків до симуляції дорожніх умов, порівняння реальних і симульованих даних та експорту розширеного датасету.

Описано дві аналітичні діаграми, інтерактивну таблицю водіїв із пошуком та детальним модальним вікном, панель симуляції з покроковим вибором погодних умов, стану покриття й часу доби, а також панель порівняння у режимах «паралельно» та «накладання». Роботу інтерфейсу проілюстровано відповідними рисунками.

ВИСНОВКИ

В результаті виконання дипломної роботи — розроблено програмне забезпечення для розширення транспортних даних шляхом моделювання дорожніх умов. Створено клієнтський вебзастосунок, що повністю у середовищі браузера виконує завантаження реальних GPS-треків формату T-Drive, обчислення кінематичних параметрів руху, класифікацію стилів водіння, симуляцію впливу дорожньо-кліматичних умов та генерацію розширеного синтетичного датасету.

Відповідно до завдань, сформульованих у першому розділі, отримано такі результати:

1. Проаналізовано сучасні методи моделювання дорожніх умов та підходи до розширення транспортних даних, що дозволило визначити вимоги до програмного забезпечення для генерації синтетичних транспортних датасетів на основі реальних GPS-траєкторій;

2. Спроектовано модульну архітектуру клієнтського вебзастосунку, яка забезпечує незалежність функціональних компонентів, масштабованість системи та централізоване керування даними між модулями обробки, моделювання та візуалізації;

3. Розроблено алгоритм обчислення кінематичних параметрів руху транспортних засобів, що визначає пройдено відстань, швидкість, прискорення та характеристики маневрування на основі послідовності GPS-координат;

4. Розроблено алгоритм класифікації стилю водіння, який здійснює автоматичне віднесення водіїв до категорій агресивного, нормального, обережного або еко-водіння на основі аналізу кінематичних показників руху;

5. Розроблено алгоритм моделювання дорожніх умов, що враховує вплив погодних явищ, стану дорожнього покриття та часу доби шляхом зміни кінематичних параметрів транспортного засобу відповідно до заданих коефіцієнтів впливу;

6. Розроблено модуль генерації розширеного датасету, який формує синтетичні записи на основі вихідних транспортних даних, доповнює їх метаданими дорожніх умов та забезпечує експорт результатів у форматах JSON і CSV;

7. Реалізовано інтерактивний вебінтерфейс для завантаження, аналізу, моделювання та візуалізації транспортних даних, що містить засоби перегляду статистики, графічного аналізу та порівняння реальних і змодельованих показників;

8. Проведено тестування програмного забезпечення, у ході якого підтверджено працездатність реалізованих алгоритмів, стабільність функціонування системи та обчислювальну коректність результатів аналізу й моделювання транспортних даних.

Практичне значення розробки полягає у створенні доступного інструменту для отримання контрольованих, статистично репрезентативних і повністю розмічених синтетичних даних у сфері інтелектуальних транспортних систем, без витрат на збір реальних даних в екстремальних дорожніх умовах. Перспективами подальших досліджень є розширення набору модельованих умов руху, інтеграція складніших моделей поведінки водія, додавання моделювання взаємодії з пішоходами та велосипедистами, а також розгортання застосунку як повноцінного вебсервісу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Yuan J., Zheng Y., Xie X., Sun G. Driving with knowledge from the physical world. Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11). San Diego, 2011. P. 316–324.
2. T-Drive trajectory data sample. Microsoft Research : вебсайт. URL: <https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/>.
3. Lopez P. A., Behrisch M., Bieker-Walz L. et al. Microscopic traffic simulation using SUMO. Proceedings of the 21st IEEE International Conference on Intelligent Transportation Systems (ITSC). Maui, 2018. P. 2575–2582.
4. SUMO — Simulation of Urban MObility. Documentation. Eclipse / DLR : вебсайт. URL: <https://sumo.dlr.de/docs/> (дата звернення: 15.04.2026).
5. PTV Vissim. Multimodal traffic simulation software. PTV Group : вебсайт. URL: <https://www.ptvgroup.com/en/products/ptv-vissim> (дата звернення: 16.04.2026).
6. Graser A. MovingPandas: efficient structures for movement data in Python. GI_Forum. 2019. Vol. 7, No. 1. P. 54–68.
7. Geotab. Fleet management and telematics platform. Geotab Inc. : вебсайт. URL: <https://www.geotab.com/> (дата звернення: 16.04.2026).
8. Krauss S. Microscopic modeling of traffic flow: investigation of collision-free vehicle dynamics : Ph.D. thesis. University of Cologne, 1998. 116 p.
9. Sinnott R. W. Virtues of the haversine. Sky and Telescope. 1984. Vol. 68, No. 2. P. 159.
10. Kilpeläinen M., Summala H. Effects of weather and weather forecasts on driver behaviour. Transportation Research Part F: Traffic Psychology and Behaviour. 2007. Vol. 10, No. 4. P. 288–299.

11. Meiring G. A. M., Myburgh H. C. A review of intelligent driving style analysis systems and related artificial intelligence algorithms. *Sensors*. 2015. Vol. 15, No. 12. P. 30653–30682.
12. Zheng Y. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology*. 2015. Vol. 6, No. 3. Article 29.
13. Treiber M., Hennecke A., Helbing D. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*. 2000. Vol. 62, No. 2. P. 1805–1824.
14. Wiedemann R. Simulation des Straßenverkehrsflusses. *Schriftenreihe des Instituts für Verkehrswesen der Universität Karlsruhe*, Heft 8. Karlsruhe, 1974.
15. Ericsson E. Independent driving pattern factors and their influence on fuel-use and exhaust emission factors. *Transportation Research Part D: Transport and Environment*. 2001. Vol. 6, No. 5. P. 325–345.
16. Murphey Y. L., Milton R., Kiliaris L. Driver's style classification using jerk analysis. 2009 IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems. Nashville, 2009. P. 23–28.
17. Vaitkus V., Lengvenis P., Žylius G. Driving style classification using long-term accelerometer information. 2014 19th International Conference on Methods and Models in Automation and Robotics (MMAR). Międzyzdroje, 2014. P. 641–644.
18. Johnson D. A., Trivedi M. M. Driving style recognition using a smartphone as a sensor platform. 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC). Washington, 2011. P. 1609–1615.
19. Akin D., Sisiopiku V. P., Skabardonis A. Impacts of weather on traffic flow characteristics of urban freeways in Istanbul. *Procedia – Social and Behavioral Sciences*. 2011. Vol. 16. P. 89–99.
20. Eboli L., Mazzulla G., Pungillo G. Combining speed and acceleration to define car users' safe or unsafe driving behaviour. *Transportation Research Part C: Emerging Technologies*. 2016. Vol. 68. P. 113–125.

21. Zheng Y., Capra L., Wolfson O., Yang H. Urban computing: concepts, methodologies, and applications. ACM Transactions on Intelligent Systems and Technology. 2014. Vol. 5, No. 3. Article 38.

22. ECMAScript 2020 Language Specification (ECMA-262, 11th edition). Ecma International : вебсайт. URL: <https://www.ecma-international.org/publications-and-standards/standards/ecma-262/> (дата звернення: 20.04.2026).

23. Chart.js. Documentation. Open-source HTML5 charting library : вебсайт. URL: <https://www.chartjs.org/docs/latest/> (дата звернення: 20.04.2026).

ДОДАТОК А

Фрагменти програмного коду

У додатку наведено ключові фрагменти програмного коду застосунку: константи коефіцієнтів дорожніх умов та основні функції обробки й аналізу транспортних даних.

А.1 Константи коефіцієнтів дорожніх умов

```
const ROAD_CONDITIONS = {
  dry: { speedFactor: 1.00, accelFactor: 1.00, label: "Суха дорога", color: "#8B7355" },
  wet: { speedFactor: 0.85, accelFactor: 0.75, label: "Мокра дорога", color: "#4A7A9C" },
  rain: { speedFactor: 0.70, accelFactor: 0.60, label: "Сильний дощ", color: "#2C5F8A" },
  snow: { speedFactor: 0.50, accelFactor: 0.45, label: "Сніг", color: "#E8E8E8" },
  ice: { speedFactor: 0.35, accelFactor: 0.30, label: "Лід", color: "#C0E0FF" },
  night: { speedFactor: 0.80, accelFactor: 0.85, label: "Ніч", color: "#1a1a2e" }
};

const WEATHER_FACTORS = {
  clear: { speedFactor: 1.0, accelFactor: 1.0 },
  cloudy: { speedFactor: 0.95, accelFactor: 0.95 },
  fog: { speedFactor: 0.6, accelFactor: 0.7 },
  rain: { speedFactor: 0.75, accelFactor: 0.65 },
  heavy_rain: { speedFactor: 0.55, accelFactor: 0.5 },
  snow: { speedFactor: 0.45, accelFactor: 0.4 },
  blizzard: { speedFactor: 0.25, accelFactor: 0.25 }
};

const SURFACE_FACTORS = {
  dry: { speedFactor: 1.0, accelFactor: 1.0 },
  wet: { speedFactor: 0.85, accelFactor: 0.75 },
  puddles: { speedFactor: 0.7, accelFactor: 0.6 },
  snow_covered: { speedFactor: 0.5, accelFactor: 0.45 },
  icy: { speedFactor: 0.35, accelFactor: 0.3 },
  black_ice: { speedFactor: 0.2, accelFactor: 0.15 }
};

const TIME_FACTORS = {
  day: { speedFactor: 1.0, accelFactor: 1.0 },
  dawn: { speedFactor: 0.9, accelFactor: 0.92 },
  dusk: { speedFactor: 0.85, accelFactor: 0.88 },
  night: { speedFactor: 0.75, accelFactor: 0.8 }
};
```

А.2 Обчислення відстані за формулою Гаверсина

```
function haversineDistance(lat1, lon1, lat2, lon2) {
  const R = 6371000;
  const phi1 = lat1 * Math.PI / 180;
  const phi2 = lat2 * Math.PI / 180;
  const deltaPhi = (lat2 - lat1) * Math.PI / 180;
  const deltaLambda = (lon2 - lon1) * Math.PI / 180;
```

```

    const a = Math.sin(deltaPhi / 2) ** 2 + Math.cos(phi1) * Math.cos(phi2) *
Math.sin(deltaLambda / 2) ** 2;
    const c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));

    return R * c;
}

```

A.3 Обчислення азимута (напрямку) руху

```

function calculateBearing(lat1, lon1, lat2, lon2) {
    const phi1 = lat1 * Math.PI / 180;
    const phi2 = lat2 * Math.PI / 180;
    const lambda1 = lon1 * Math.PI / 180;
    const lambda2 = lon2 * Math.PI / 180;

    const y = Math.sin(lambda2 - lambda1) * Math.cos(phi2);
    const x = Math.cos(phi1) * Math.sin(phi2) - Math.sin(phi1) * Math.cos(phi2) *
Math.cos(lambda2 - lambda1);

    return (Math.atan2(y, x) * 180 / Math.PI + 360) % 360;
}

```

A.4 Обчислення перцентилі

```

function percentile(arr, p) {
    const sorted = [...arr].sort((a, b) => a - b);
    const idx = Math.floor((p / 100) * sorted.length);
    return sorted[idx] || 0;
}

// =====
// DATA PARSING
// =====

```

A.5 Парсинг GPS-даних формату T-Drive

```

function parseGPSData(content, filename) {
    const lines = content.trim().split('\n');
    const data = [];

    for (const line of lines) {
        const parts = line.split(',');
        if (parts.length >= 4) {
            const [taxiId, timestamp, longitude, latitude] = parts;
            const date = new Date(timestamp.replace(' ', 'T'));

            if (!isNaN(date.getTime())) {
                data.push({
                    taxiId: taxiId.trim(),
                    timestamp: date,
                    longitude: parseFloat(longitude),
                    latitude: parseFloat(latitude),
                    source: filename
                });
            }
        }
    }

    return data.sort((a, b) => {

```

```

        if (a.taxiId !== b.taxiId) return a.taxiId.localeCompare(b.taxiId);
        return a.timestamp - b.timestamp;
    });
}

// =====
// DATA PROCESSING
// =====

```

A.6 Обчислення кінематичних параметрів руху

```

function calculateMovementParameters(data) {
    const processed = [];
    const grouped = {};

    for (const point of data) {
        if (!grouped[point.taxiId]) grouped[point.taxiId] = [];
        grouped[point.taxiId].push(point);
    }

    for (const [taxiId, points] of Object.entries(grouped)) {
        let prevSpeed = 0;
        let prevBearing = 0;

        for (let i = 1; i < points.length; i++) {
            const prev = points[i - 1];
            const curr = points[i];

            const timeDiff = (curr.timestamp - prev.timestamp) / 1000;
            if (timeDiff <= 0 || timeDiff > 600) continue;

            const distance = haversineDistance(prev.latitude, prev.longitude, curr.latitude,
curr.longitude);
            const speedMs = distance / timeDiff;
            const speedKmh = speedMs * 3.6;

            if (speedKmh > 200) continue;

            const acceleration = (speedKmh - prevSpeed) / 3.6 / timeDiff;
            const bearing = calculateBearing(prev.latitude, prev.longitude, curr.latitude,
curr.longitude);
            let turnAngle = Math.abs(bearing - prevBearing);
            if (turnAngle > 180) turnAngle = 360 - turnAngle;

            processed.push({
                taxiId,
                timestamp: curr.timestamp,
                latitude: curr.latitude,
                longitude: curr.longitude,
                distance,
                timeDiff,
                speedKmh,
                acceleration: Math.max(-10, Math.min(10, acceleration)),
                bearing,
                turnAngle: i > 1 ? turnAngle : 0
            });

            prevSpeed = speedKmh;
            prevBearing = bearing;
        }
    }
}

```

```
    }  
    return processed;  
}
```

ДОДАТОК Б



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Навчально-науковий інститут атомної та теплової енергетики
Кафедра інженерії програмного забезпечення в енергетиці

Програмне забезпечення для розширення транспортних даних шляхом моделювання дорожніх умов

Білогривий Владислав Юрійович,
ТВ-22 Доцент, к.е.н., Гусєва І. І.

Київ – 2026

1/18



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Актуальність

Актуальність теми полягає у постійному зростанні кількості транспортних даних, які використовуються для аналізу дорожнього руху, систем моніторингу транспорту та інтелектуальних транспортних систем.
Більшість існуючих датасетів містять лише базові GPS-координати та не враховують погодні умови, стан дорожнього покриття чи поведінку водія в різних ситуаціях.
Тому виникає необхідність створення програмного забезпечення, яке дозволяє моделювати дорожні умови та генерувати розширені транспортні дані.



2/18



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Постановка завдання

Мета

Метою є розширення транспортних даних шляхом моделювання дорожніх умов. Для досягнення поставленої мети необхідно виконати такі завдання:

Завдання роботи

- Проаналізувати методи моделювання дорожніх умов;
- Спроектувати модульну архітектуру вебзастосунку;
- Розробити алгоритми обчислення кінематичних параметрів та класифікації стилю водіння;
- Класифікувати стиль водіння за інтегральним індексом Driving Style Index;
- Реалізувати завантаження і валідацію даних, модуль симуляції дорожніх умов та генерацію розширеного датасету (JSON, CSV);
- Створити інтерактивний вебінтерфейс;
- Реалізувати генерацію розширеного датасету;
- Провести тестування та оцінити якість згенерованих даних.

3/18



Огляд аналогів

SUMO
СИМУЛЯЦІЯ РУХУ

Потужний відкритий симулятор, але генерує трафік «з нуля» й потребує складного калібрування.

PTV Vissim
СИМУЛЯЦІЯ РУХУ

Детальна модель водія Віденська, проте платний і не призначений для роботи з GPS-треками.

MovingPandas
АНАЛІЗ ТРАЕКТОРІЙ

Гнучка Python-бібліотека, але без вебінтерфейсу й без моделювання дорожніх умов.

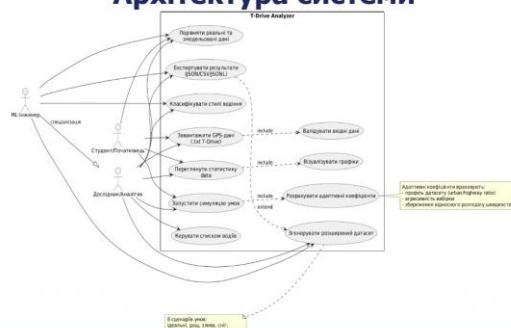
Geotab
ТЕЛЕМАТИКА

Аналіз поведінки в реальному часі, але закритий SaaS і потребує апаратних датчиків.

4/18



Архітектура системи



4/18



Опис моделювання дорожніх умов

Призначення

Модуль штучно відтворює зовнішні дорожні умови та масштабує кінематичні параметри реальних GPS-треків.

Так отримуються нові реалістичні траєкторії для сценаріїв, яких немає у вихідному наборі, що розширює датасет за різноманітністю умов руху.

Які умови моделюються

- Погода: ясно, туман, дощ, сніг, замітіль
- Стан покриття: суха, мокра, калюжі, сніг, ожеледь
- Час доби: день / ніч
- Інтенсивність опадів
- Рівень трафіку
- Видимість

Модель: коефіцієнти впливу (speed / accel)

Погода	х швидк.	х приск.	Покриття	х швидк.	х приск.
Ясно	1.00	1.00	Суха	1.00	1.00
Туман	0.75	0.85	Мокра	0.85	0.75
Дощ	0.70	0.60	Калюжі	0.70	0.60
Сніг	0.50	0.45	Сніг	0.50	0.45
Замітіль	0.25	0.25	Ожеледь	0.35	0.30

Час доби додає окремий множник (день 1.0 · ніч 0.75). Слайдери опадів і видимості вносять лінійні поправки $\pm 10\%$.

Як моделюємо та як використовуємо

Як: обрані погода, покриття й час доби дають комбінований коефіцієнт; ним масштабуються параметри руху кожної точки треку, шум зберігає природні флуктуації.

Комбінування: фактори перемножуються — можна задати складний сценарій «ніч + дощ + мокра дорога + високий трафік».

Як використовуємо: отримуюмо нові траєкторії для умов, яких немає в реальному наборі, і так розширюємо датасет за різноманітністю.

6/18



Опис стилів водіння

Аналізується

- ❖ середня швидкість;
- ❖ різкі прискорення;
- ❖ гальмування;
- ❖ повороти;
- ❖ маневрування.

Нормалізація

$$x_i^{norm} = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

Середня швидкість

$$V = \frac{1}{n} \sum_{i=1}^n v_i$$

Інтенсивність гальмування

$$B = \frac{1}{n} \sum_{i=1}^n |b_i|$$

Приклад вагових коефіцієнтів

Параметр	Коефіцієнт
Швидкість	0.35
Прискорення	0.30
Гальмування	0.20
Маневри	0.15

Формула індексу стилю водіння

$$DSI = w_1 V + w_2 A + w_3 B + w_4 T$$

Середнє прискорення

$$A = \frac{1}{n} \sum_{i=1}^n |a_i|$$

Кількість різких маневрів

$$T = N_{turns} + N_{hard\ braking} + N_{rapid\ acceleration}$$

Класифікація стилів

Drive Style Index (DSI)	Стиль
0-25	Eco
25-50	Cautious
50-75	Normal
75-100	Aggressive

7/18



Алгоритм обчислення кінематичних даних

Алгоритм обробки GPS-даних:

- Зчитування GPS-треків (ID водія, час, координати)
- Сортування даних за водієм і часом
- Обчислення базових параметрів руху:
 - відстань між точками
 - інтервал часу
- Розрахунок кінематичних характеристик:
 - швидкість
 - прискорення
 - напрямок руху (bearing)
 - кут повороту
- Фільтрація аномальних значень
- Формування набору ознак для аналізу стилю водіння

Формула Гаверсина

$$d = 2 R * \arcsin \left(\sin^2 \left(\frac{\varphi_2 - \varphi_1}{2} \right) + \cos \varphi_1 * \cos \varphi_2 * \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right) \right)$$

$$\Delta \phi = \phi_2 - \phi_1 \text{ — різниця широти}$$

$$\Delta \lambda = \lambda_2 - \lambda_1 \text{ — різниця довготи}$$

8/18



Вхідні дані

Формат запису T-Drive

- ID водія (taxiId);
- часова мітка (YYYY-MM-DD HH:MM:SS);
- довгота (longitude);
- широта (latitude).

Після завантаження записи сортується за водієм і часом, формуючи сегменти руху. Підтримується накопичувальне додавання кількох файлів в один датасет.

Формат вхідних даних

13,2000-02-03 05:20:04,116.35124,39.91248
13,2000-02-03 05:25:06,116.3504,39.91886
13,2000-02-03 05:30:08,116.3506,39.91889
13,2000-02-03 05:35:10,116.35281,39.91125
13,2000-02-03 05:40:12,116.35289,39.91129
13,2000-02-03 05:45:14,116.35071,39.91208
13,2000-02-03 05:50:16,116.35134,39.91222
13,2000-02-03 05:55:17,116.35144,39.94288
13,2000-02-03 06:00:19,116.42521,39.94809
13,2000-02-03 06:05:21,116.47051,39.96301
13,2000-02-03 06:10:23,116.47051,39.96301
13,2000-02-03 06:15:24,116.46112,39.9794
13,2000-02-03 06:20:26,116.44156,39.97168
13,2000-02-03 06:25:28,116.40088,39.96758
13,2000-02-03 06:30:30,116.4008,39.94762
13,2000-02-03 06:35:31,116.4008,39.94762
13,2000-02-03 06:40:33,116.44778,39.95089
13,2000-02-03 06:45:35,116.42574,39.93247
13,2000-02-03 06:50:37,116.35361,39.91864
13,2000-02-03 07:00:40,116.3528,39.91883
13,2000-02-03 07:05:42,116.35382,39.90624
13,2000-02-03 07:10:43,116.35299,39.90621
13,2000-02-03 07:15:45,116.35299,39.90621
13,2000-02-03 07:20:47,116.35688,39.91113
13,2000-02-03 07:25:49,116.36078,39.90863
13,2000-02-03 07:30:51,116.36084,39.9087

Де дані розширюються

Вхідний набір містить лише чотири поля. Розширення відбувається на двох етапах:

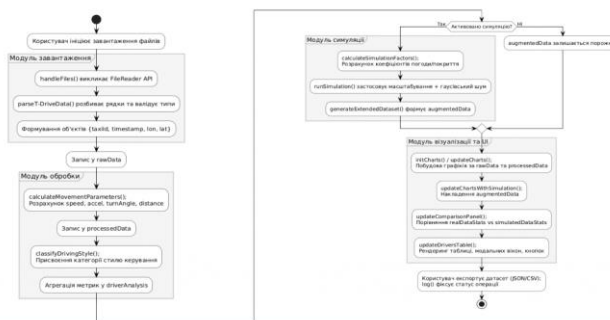
- Кінематична обробка** додає швидкість, прискорення, гальмування, кут повороту, дистанцію та індекс стилю водіння.
- Моделювання умов** додає метадані середовища: погоду, стан покриття, час доби, інтенсивність опадів, трафік і видимість.

Так чотири початкові поля перетворюються на повноцінний розширений запис для ML.

9/18



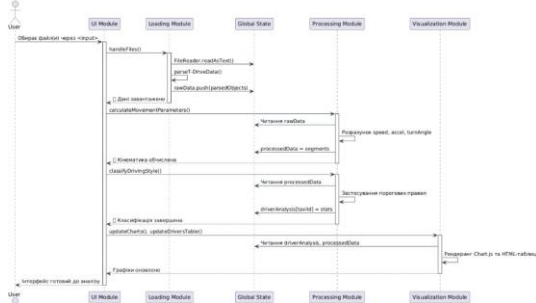
Конвеєр обробки та аналізу даних



10/18



Послідовність взаємодії при завантаженні та обробці



11/18



Використані технології



HTML5
структура інтерфейсу



CSS3
оформлення, адаптивна верстка (Grid / Flexbox)



JavaScript ES2020+
всю логіку: парсинг, розрахунки, класифікація, симуляція



Chart.js 4.x
Інтерактивні діаграми (підключення через CDN)

Вхідні дані: TXT-файли датасету T-Drive

Вихідні дані: розмічений датасет у форматах CSV · JSON · JSONL для ML-конвеєрів

12/18

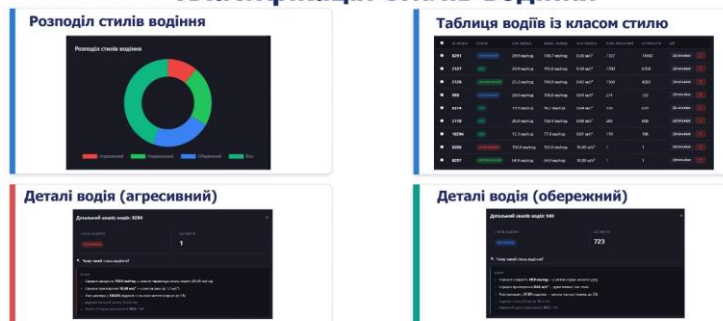


Моделювання дорожніх умов

13/18



Класифікація стилів водіння



14/18



Характеристика, генерація та експорт датасету

15/18



Приклад моделювання дорожніх умов

Реальний кейс: агресивний водій під впливом змодельованих умов

Сценарій: Заметіль + Сніг + Сутінки · Коефіцієнт швидкості 80% · 1 сегмент траси

Середня швидкість

реальні: 103.0 км/год

змодельовані: 83.4 км/год

$\Delta -20.0\%$

Середнє прискорення

реальні: 10.00 м/с²

змодельовані: 1.07 м/с²

$\Delta -89.3\%$

Загальний вплив умов

інтегральна зміна руху

$\Delta -54.6\%$

Порівняння «Паралельно» — реальні vs змодельовані



Накладення розподілів швидкості та прискорення



Висновок: модель фактично коректно реагує на умови — швидкість і прискорення агресивного водія падають до рівня обережного руху, що підтверджує реальність змодельованих даних.

16/18



Висновок

У результаті виконання роботи розроблено програмне забезпечення для розширення транспортних даних шляхом моделювання дорожніх умов. Проведено аналіз існуючих підходів до моделювання дорожньої ситуації, спроектовано модульну архітектуру вебзастосунку та реалізовано алгоритми обчислення кінематичних параметрів транспортного руху.

У роботі розроблено механізм класифікації стилів водіння на основі мультикритеріальної системи оцінювання, реалізовано засоби завантаження та валідації вхідних даних, а також модуль симуляції дорожніх умов. Програмний продукт забезпечує формування розширених датасетів у форматах JSON і CSV із додаванням метаданих про умови руху, що підвищує інформативність транспортних даних та розширює можливості їх подальшого аналізу.

Розроблений вебзастосунок надає засоби візуалізації результатів аналізу, порівняння реальних і симульованих даних, а також інструменти для дослідження поведінки водіїв. Проведене тестування підтвердило працездатність, стабільність та достатню обчислювальну точність реалізованих алгоритмів.

Отримані результати можуть бути використані для підготовки даних у задачах машинного навчання, розробки телематичних систем, дослідження транспортної мобільності та аналізу впливу дорожніх умов на поведінку водіїв.

17/18



Дякую за увагу!

18/18



Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

—
National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"