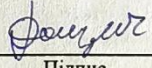
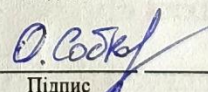
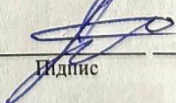
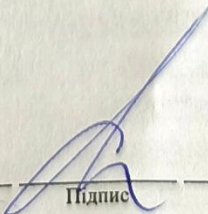


КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Метод інтелектуальної підтримки пілота автосимулятора на основі аналізу телеметрії з використанням засобів машинного навчання

Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань
Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності
Освітня програма Комп'ютерні науки
Назва освітньої програми

Виконав: студент групи КН-22-2  Вадим ДОЩИЧ
Група виконавця Підпис Ім'я, ПРІЗВИЩЕ
Керівник: Ph.D., ст. викл. каф. КН  Олена СОБКО
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ
Нормоконтроль: к.т.н., доц. каф. КН  Руслан БАГРІЙ
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ

До захисту допускаю:
Зав. кафедри КН, д.т.н., професор  Олександр БАРМАК
Підпис Ім'я, ПРІЗВИЩЕ

18 06 2026 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Факультет інформаційних технологій
Кафедра комп'ютерних наук
Освітній ступінь бакалавр
Галузь знань 12 – Інформаційні технології
Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук


(підпис)
д.т.н., професор Олександр БАРМАК

22 липня 2026 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

1. Тема кваліфікаційної роботи бакалавра: «Метод інтелектуальної підтримки пілота автосимулятора на основі аналізу телеметрії з використанням засобів машинного навчання»

2. Завдання видано студенту

Вадиму Доцичу

(Ім'я, прізвище)

3. Керівник роботи

старший викладач кафедри КН Олена Собко

(посада, ім'я, прізвище)

4. Затверджено наказом університету від 20 липня 2026 р. № 7

5. Дата видачі завдання студенту: 22 липня 2026 р.

6. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Метою кваліфікаційної роботи бакалавра є підвищення ефективності керування транспортним засобом у гоночному симуляторі за рахунок підвищення точності автоматичного виявлення помилок пілотування, підвищення точності прогнозування пов'язаних із ними часових втрат та покращення результатів проходження контрольного кола шляхом формування персоналізованих рекомендацій на основі аналізу телеметричних даних із використанням методів машинного навчання. Для досягнення мети необхідно: виконати аналіз предметної області; підтримки пілота автосимулятора на основі аналізу телеметрії; розробити метод інтелектуальної підтримки пілота автосимулятора на основі аналізу телеметрії з використанням засобів машинного навчання; реалізувати інтелектуальну систему підтримки пілота автосимулятора; провести експериментальне дослідження запропонованого методу на основі телеметричних даних гоночного симулятора з оцінюванням точності автоматичного виявлення помилок пілотування, точності прогнозування пов'язаних із ними часових втрат та впливу запропонованого підходу на результати проходження контрольного кола.

7. Календарний план виконання кваліфікаційної роботи бакалавра:

№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	Вибір напрямку дослідження та узгодження теми кваліфікаційної роботи з керівником, складання календарного графіка виконання	січень 2026	Виконано
2	Ознайомлення з предметною областю, формулювання мети і задач дослідження, визначення об'єкта та предмета дослідження	лютий 2026	Виконано
3	Проектування методу розв'язання задачі, опис архітектурних рішень, розроблення математичних моделей та алгоритмів.	березень 2026	Виконано
4	Обґрунтування інструментарію розробки, програмна реалізація розробленого методу, проведення експериментального тестування та оцінювання ефективності.	квітень 2026	Виконано
5	Написання тексту кваліфікаційної роботи, урахування зауважень керівника, оформлення згідно з вимогами	травень 2026	Виконано
6	Розроблення презентаційних матеріалів та попередній захист кваліфікаційної роботи	травень 2026	Виконано
7	Отримання відгуку керівника, рецензії, перевірка тексту кваліфікаційної роботи на плагіат, нормоконтроль	червень 2026	Виконано
8	Підготовка до захисту та захист кваліфікаційної роботи	червень 2026	Виконано

Виконавець:

студент групи КН-21-1
Група виконавця

Дошич
Підпис

Вадим ДОЩИЧ
Ім'я, ПРІЗВИЩЕ

Керівник:

Ph.D., ст. викл., каф. КН
Науковий ступінь, посада

О. Собко
Підпис

Олена СОБКО
Ім'я, ПРІЗВИЩЕ

Анотація

Тема кваліфікаційної роботи бакалавра: Метод інтелектуальної підтримки пілота автосимулятора на основі аналізу телеметрії з використанням засобів машинного навчання

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-22-2
Вадим Доцич

Керівник кваліфікаційної роботи бакалавра: старший викладач КН
Олена Собко

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
82	16	2	42	2

Метою кваліфікаційної роботи є підвищення ефективності керування транспортним засобом у гоночному симуляторі за рахунок підвищення точності автоматичного виявлення помилок пілотування, підвищення точності прогнозування пов'язаних із ними часових втрат та покращення результатів проходження контрольного кола шляхом формування персоналізованих рекомендацій на основі аналізу телеметричних даних із використанням методів машинного навчання.

Практичне значення роботи полягає у розробленні інтелектуальної системи аналізу телеметричних даних, яка дозволяє автоматично виявляти помилки керування транспортним засобом у гоночному симуляторі та надавати рекомендації для покращення часу проходження кола.

Ключові слова: машинне навчання, інтелектуальна система, аналіз телеметрії, гоночний симулятор, підтримка пілота, класифікація помилок, часові ряди, Random Forest, Gradient Boosting, DTW.

Виконавець: студент групи КН-22-2
Група виконавця

Доцич
Підпис

Вадим ДОЦИЧ
Ім'я, ПРІЗВИЩЕ

Зміст

Перелік скорочень	4
Вступ.....	5
Розділ 1 Аналіз предметної області та сучасних підходів до обробки телеметричних даних у гоночних симуляторах.....	7
1.1 Аналіз підходів до збору та обробки телеметричних даних у гоночних симуляторах	7
1.2 Огляд існуючих теоретичних підходів до інтелектуальної підтримки пілота автосимулятора.....	10
1.3 Аналіз наукових рішень у задачі аналізу телеметрії пілота автосимулятора.....	15
1.4 Аналіз існуючих програмних засобів	18
1.5 Мета, задачі та вимоги до реалізації інтелектуальної системи.....	21
Розділ 2 Метод інтелектуальної підтримки пілота автосимулятора на основі аналізу телеметрії з використанням засобів машинного навчання.....	22
2.1 Постановка задачі підтримки пілота автосимулятора на основі аналізу телеметрії.....	22
2.2 Формування простору ознак телеметричних даних	25
2.3 Задача класифікації типів помилок пілотування	30
2.4 Задача регресійного оцінювання часових втрат	33
2.5 Формування рекомендацій пілоту.....	35
2.6 Отримання моделей машинного навчання для аналізу телеметрії.....	37
2.7 Етапи методу інтелектуальної підтримки пілота автосимулятора	44
2.8 Програмна реалізація методу та експериментальний пайплайн	46
2.9 Оцінювання ефективності запропонованого методу	48
2.10 Висновки до розділу 2	57
Розділ 3 Експериментальне дослідження методу та застосування інтелектуальної системи	58
3.1 Опис середовища тестування та інтелектуальної системи	58

3.2 Дослідження точності виявлення помилок керування та якості прогнозування часових втрат.....	61
3.3 Дослідження впливу інтелектуальної системи на результати проходження контрольного кола.....	68
3.5 Обговорення обмежень методу та напрями вдосконалення.....	72
3.6 Висновки до розділу 3	74
Загальні висновки.....	76
Перелік посилань.....	78
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
RNN	Recurrent Neural Network
PPO	Proximal Policy Optimization
ML	Machine Learning
MAE	Mean Absolute Error
LSTM	Long Short-Term Memory
SVM	Support Vector Machine
RNN	Recurrent Neural Network
BiLSTM	Bidirectional Long Short-Term Memory
CBANet	Convolutional Bidirectional Attention Network
CNN	Convolutional Neural Network
1D-CNN	One-Dimensional Convolutional Neural Network
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DDPG	Deep Deterministic Policy Gradient
DE	Data Engineer
DS	Data Science
DTW	Dynamic Time Warping
F1	F1-score
G-сили	Gravitational Force
k-NN	k-Nearest Neighbors
K-Means	Алгоритм кластеризації K-Means
LightGBM	Light Gradient Boosting Machine
MoTeC	Modular Telemetry Controller
MSE	Mean Squared Error
SVR	Support Vector Regression
VRS	Virtual Racing School
XGBoost	Extreme Gradient Boosting

Вступ

Кваліфікаційна робота бакалавра присвячена підвищенню ефективності керування транспортним засобом у гоночному симуляторі за рахунок підвищення точності автоматичного виявлення помилок пілотування, підвищення точності прогнозування пов'язаних із ними часових втрат та покращення результатів проходження контрольного кола шляхом формування персоналізованих рекомендацій на основі аналізу телеметричних даних із використанням методів машинного навчання.

Актуальність. Сучасний етап розвитку інформаційних технологій характеризується активним впровадженням інтелектуальних систем аналізу даних у найрізноманітніші сфери людської діяльності, серед яких особливе місце посідають цифрові моделювання та спортивні симуляції. Використання складних програмних комплексів для відтворення реальних фізичних процесів дозволяє не лише створювати високоточні віртуальні середовища, а й накопичувати значні обсяги телеметричної інформації, що є критично важливою для оцінки ефективності взаємодії людини з технічною системою. Саме тому розробка та впровадження автоматизованих засобів обробки цих даних стає фундаментальним кроком на шляху до вдосконалення професійних навичок та досягнення максимальних результатів у технічних видах діяльності.

Актуальність дослідження зумовлена стрімкою цифровізацією сучасного автоспорту та зростанням ролі гоночних симуляторів як професійних інструментів для підготовки реальних пілотів. Протягом останніх років віртуальні гоночні середовища перетворилися на складні системи, що генерують величезні обсяги телеметричних даних, які відображають кожен аспект динаміки транспортного засобу та техніки його керування. Водночас традиційний аналіз таких даних кваліфікованими інженерами є надзвичайно трудомістким процесом, що потребує специфічних знань та значних часових витрат. У зв'язку з цим виникає об'єктивна потреба у створенні інтелектуальних систем, здатних автоматично виявляти відхилення від еталонних показників та надавати пілотам

детальний аналітичний звіт. Застосування сучасних засобів інтелектуального аналізу даних та методів машинного навчання дозволяє автоматизувати процес порівняння сесій, забезпечуючи високу точність ідентифікації помилок пілотування, що раніше було доступно лише при експертному аналізі.

Об'єкт дослідження – процес аналізу телеметричних даних пілотування у гоночних симуляторах для виявлення відхилень від оптимального керування транспортним засобом.

Предмет дослідження – методи машинного навчання та алгоритми обробки телеметричних даних, що використовуються для інтелектуальної підтримки пілота гоночного симулятора.

Мета кваліфікаційної роботи бакалавра полягає у підвищенні ефективності керування транспортним засобом у гоночному симуляторі за рахунок підвищення точності автоматичного виявлення помилок пілотування, підвищення точності прогнозування пов'язаних із ними часових втрат та покращення результатів проходження контрольного кола шляхом формування персоналізованих рекомендацій на основі аналізу телеметричних даних із використанням методів машинного навчання.

Завданнями кваліфікаційної роботи бакалавра є: виконати аналіз предметної області, дослідивши сучасні підходи до обробки телеметричних даних в автоспорті та методи інтелектуальної підтримки пілотів у гоночних симуляторах; підтримки пілота автосимулятора на основі аналізу телеметрії; розробити метод інтелектуальної підтримки пілота автосимулятора на основі аналізу телеметрії з використанням засобів машинного навчання; реалізувати інтелектуальну систему підтримки пілота автосимулятора, що базується на розробленому методі; провести експериментальне дослідження запропонованого методу на основі телеметричних даних гоночного симулятора з оцінюванням точності автоматичного виявлення помилок пілотування, точності прогнозування пов'язаних із ними часових втрат та впливу запропонованого підходу на результати проходження контрольного кола.

Розділ 1 Аналіз предметної області та сучасних підходів до обробки телеметричних даних у гоночних симуляторах

1.1 Аналіз підходів до збору та обробки телеметричних даних у гоночних симуляторах

З позиції комп'ютерних наук сучасні системи аналізу гоночних заїздів належать до класу інтелектуальних інформаційних систем, що працюють з великими обсягами багатовимірних часових даних. Телеметрична інформація, яка формується під час руху транспортного засобу, являє собою набір синхронізованих потоків даних, що потребують застосування методів цифрової обробки сигналів, аналізу часових рядів, машинного навчання та інтелектуального аналізу даних. Розроблення програмних засобів для автоматичного виявлення закономірностей у телеметрії, класифікації помилок керування та прогнозування їхнього впливу на результати заїзду є актуальною науково-прикладною задачею галузі інформаційних технологій та штучного інтелекту.

Сучасний автоспорт є однією з найбільш технологічно насичених галузей, де результат залежить не лише від майстерності пілота, а й від ефективності взаємодії людини з високотехнологічним транспортним засобом [1]. За останні десятиліття в цій індустрії відбулася трансформація від чисто механічних змагань до змагань даних. Паралельно з реальним автоспортом стрімкого розвитку набув симрейсинг (sim racing) – професійний сегмент кіберспорту, що базується на високоточному фізичному моделюванні гоночних процесів [2]. Якщо раніше віртуальні перегони сприймалися як розвага, то сьогодні вони є повноцінною платформою для підготовки пілотів, тестування налаштувань автомобілів та проведення міжнародних чемпіонатів, що за рівнем конкуренції не поступаються реальним аналогам.

Ключовим елементом у досягненні максимальної ефективності в обох випадках є телеметрія. Телеметрія в контексті гоночних змагань – це сукупність систем, що дозволяють здійснювати дистанційне вимірювання та збір показників

роботи вузлів автомобіля та дій пілота в режимі реального часу або для подальшого аналізу [3]. Вона є основним джерелом об'єктивної інформації, що дозволяє перетворити суб'єктивні відчуття водія на конкретні цифрові показники.

У структурі сучасної гоночної команди або при професійній індивідуальній підготовці виділяють кілька ключових ролей:

– Гоночний пілот – безпосередньо керує транспортним засобом, його обов'язком є стабільне проходження дистанції з мінімальним часом кола [4].

– Інженер з аналізу даних (DE) – здійснює збір, обробку та інтерпретацію телеметричних логів. Він відповідає за пошук аномалій у поведінці авто та ідентифікацію помилок пілота [5].

– Гоночний інженер / Тренер – на основі аналізу даних формує стратегію змін у налаштуваннях автомобіля або коригує техніку пілотування [6].

Найбільш важливими функціями, що підлягають автоматизації засобами інформаційних технологій, є порівняльний аналіз декількох сесій та пошук резервів швидкості. Традиційно інженер змушений вручну накладати графіки швидкості та педалей одного кола на інше, намагаючись візуально знайти розбіжності. Це створює широке поле для застосування методів штучного інтелекту, які здатні автоматично виявляти ідеальні патерни та вказувати на конкретні географічні точки траси, де пілот припускається помилок.

У цьому контексті важливо розмежовувати поняття якості та точності проходження кола. Точність проходження кола – це міра того, наскільки дії пілота (моменти натискання педалей, імплементація гальмівного зусилля, кути повороту керма) та фізична траєкторія автомобіля збігаються з математично вирахованим чи еталонним ідеалом. Зі свого боку, якість проходження кола є ширшим та комплексним поняттям. Це не просто здатність показати мінімальний час під час однієї спроби (хотлап), а й спроможність пілота зберігати високий темп, не виходячи за межі оптимальних температурних вікон шин та раціонально використовуючи ресурс автомобіля [7]. Таким чином,

точність є технічним інструментом пілота, а якість – підсумковим результатом, який система прагне підвищити.

На якість та точність водіння безпосередньо впливають відхилення від оптимальних параметрів керування, які виявляються під час аналізу телеметрії. До таких типових відхилень відносяться: запізніле або занадто раннє гальмування; некоректне дозування газу на виході з повороту, що веде до втрати зчеплення (пробуксовки); надмірний кут повороту керма, який провокує знесення передньої осі; або ж раптове відпускання педалі газу перед або всередині повороту (lift-off), що порушує аеродинамічний баланс ваги автомобіля.

Предметна область оперує специфічною термінологією, розуміння якої є необхідним для подальшого проектування системи аналізу:

- Коло (Lap) – один повний прохід замкненої гоночної траси [8].
- Час кола (Lap Time) – основний критерій ефективності, вимірюється в мілісекундах [9].
- Дельта (Delta) – математична різниця в часі між поточним колом та референсним (еталонним) у певній точці траси [10].
- Апекс (Apex) – точка траєкторії, найближча до внутрішнього краю повороту [11].
- Телеметричний канал – окремий потік часових даних (швидкість, ступінь натискання педалі газу або гальма, кут повороту керма, поздовжні та поперечні G-сили).
- Трейл-брейкінг (Trail braking) – гоночна техніка поступового відпускання педалі гальма під час входу в поворот для збереження завантаження передньої осі [12].
- Стабільність (Consistency) – здатність пілота демонструвати повторюваний, мінімально мінливий час та траєкторію протягом тривалого відрізка заїзду.

– Сетап (Setup) – набір технічних налаштувань автомобіля (тиск у шинах, жорсткість підвіски, кліренс, аеродинамічний баланс), що безпосередньо впливають на його поведінку.

Проведений аналіз предметної області показує, що задача підвищення ефективності пілотування може бути формалізована як задача аналізу багатовимірних просторово-часових даних, яка включає етапи збору, синхронізації, попередньої обробки, виділення інформативних ознак, класифікації поведінкових патернів та прогнозування цільових показників. Така постановка природним чином узгоджується з сучасними підходами комп'ютерних наук до побудови інтелектуальних систем підтримки прийняття рішень і створює підґрунтя для застосування алгоритмів машинного навчання з метою автоматизації процесів аналізу телеметрії та формування рекомендацій для пілота.

1.2 Огляд існуючих теоретичних підходів до інтелектуальної підтримки пілота автосимулятора

Завдання аналізу гоночної телеметрії зводиться до обробки багатовимірних часових рядів (multivariate time-series data), які генеруються симулятором із надзвичайно високою частотою. Оскільки ручний аналіз таких великих масивів даних є неефективним, сучасна наука про дані (DS) пропонує спектр теоретичних підходів та математичних моделей машинного навчання (ML), здатних автоматизовано виявляти приховані нелінійні залежності між мікро-діями водія та його фінальним часом кола. Залежно від конкретної підзадачі – чи то прогнозування ймовірної втрати часу, класифікація специфічних помилок, чи алгоритмічний пошук абсолютно ідеальної траєкторії – в автоспортивній інженерії виділяють та адаптують декілька ключових класів алгоритмів штучного інтелекту.

Одним із першочергових та найефективніших підходів є використання ансамблевих методів на базі дерев рішень [13]. До них належить широко відомий

алгоритм випадкового лісу [14]. Також до цієї групи входить алгоритм градієнтного бустингу [15], який на практиці представлений такими архітектурами, як XGBoost [16] чи LightGBM [17]. Теоретично такі регресори відмінно зарекомендували себе в задачах роботи з форматованими табличними даними. Стосовно телеметрії вони можуть ефективно використовуватися для побудови референсної моделі ідеального проходження та для предиктивного аналізу втрати часу. Алгоритм здатний самостійно проаналізувати сотні кіл різних пілотів і виявити, як саме комбінація швидкості, ступеня натиснення педалі гальма та кута повороту керма в певній географічній точці траси впливає на кінцевий результат проходження сектора. Головними перевагами цього класу моделей є надзвичайно висока швидкість навчання, стійкість до шумових викидів у вхідних даних датчиків, здатність моделювати складні нелінійні залежності фізичних рухів та значна інтерпретованість результатів, що дозволяє інженеру легко оцінювати рівень впливу кожної окремої дії.

Для вирішення іншого класу задач – семантичного розпізнавання патернів конкретних помилок – традиційно застосовують класичні методи класифікації та кластеризації. Зокрема, класичну логістичну регресію або метод опорних векторів (Support Vector Machine) можна адаптувати для багатокласового віднесення виокремленого фрагменту телеметрії до переліку відомих типів помилок [18]. До таких помилок належать надмірно раннє гальмування, пізнє заходження на апекс та часткове відпускання педалі газу всередині поворотної дуги (так званий ефект lift-off) [19]. Зі свого боку, для автоматизованої ідентифікації та групування геометрично подібних ділянок треку (наприклад, швидкісних дуг) система може використовувати алгоритм кластеризації без вчителя K-Means [20].

Альтернативним інструментом для вирішення цієї ж задачі є алгоритм просторової кластеризації на основі щільності (DBSCAN) [21]. Використання таких підходів дає змогу системі чітко розпізнавати специфічні конфігурації траси, зокрема повільні затичні шпильки [22].

Також завдяки цим методам забезпечується коректне виявлення та групування складніших елементів, до прикладу S-подібних шикан [23]. Подібна просторова фрагментація дозволяє застосовувати до різних кластерів вузькоспеціалізовані правила та порогові значення оцінки. Окрім того, алгоритми кластеризації успішно використовуються для сегментації стилів водіння самих кіберспортсменів, поділяючи їх на агресивні, обережні чи плавні.

Враховуючи специфіку гоночної телеметрії як безперервного та динамічного часового ряду, базові алгоритми машинного навчання часто виявляються обмеженими через нездатність повноцінно враховувати контекст попередніх дій пілота. Теоретичним вирішенням проблеми врахування часового контексту є використання рекурентних нейронних мереж (RNN) [24]. Зокрема, високу ефективність у цьому напрямі демонструють їх спеціалізовані архітектури довгої короткострокової пам'яті (LSTM) [25]. Оскільки ці моделі орієнтовані на обробку послідовностей, вони здатні математично апроксимувати причинно-наслідкові зв'язки, розподілені у часі. Наприклад, навчена LSTM-модель здатна виявити, що помилка на виході з повороту, яка призвела до надмірного ковзання задньої осі, є відкладеним наслідком завищеної швидкості на етапі входу в поворот, тобто була закладена раніше і вже не підлягала компенсації подальшим керуванням. Це робить рекурентні мережі перспективним інструментом для розробки комплексних рекомендаційних систем та розширеного предиктивного моделювання.

Своєрідним альтернативним, хоча і неочевидним теоретично обґрунтованим напрямком є застосування одновимірних згорткових нейронних мереж (1D-CNN). [26] Попри те, що традиційно архітектури сімейства CNN у свідомості дослідників асоціюються виключно з задачами комп'ютерного зору та розпізнаванням двовимірних візуальних зображень, їх адаптовані одновимірні варіації постійно доводять високу ефективність у задачах аналізу хвильових сигналів. При адаптації до середовища симрейсингу, модель 1D-CNN може концептуально опрацьовувати одновимірний графік інтенсивності гальмування або дроселювання як своєрідний графічний монолітний патерн. Подібне

представлення дозволяє системі блискавично та без попередньої екстракції ознак ідентифікувати використання таких просунутих технік як трейл-брейкінг, або, навпаки, знаходити небажані локальні аномалії, на зразок короткочасних неконтрольованих сплесків тиску педалі, які свідчать про перезбудження пілота чи мікро-блокування гальмівних дисків.

Найбільш технологічно вибагливим і водночас новаторським підходом до проблеми інтелектуальної підтримки водія залишається навчання з підкріпленням [27]. У рамках цього методологічного вектора штучні віртуальні агенти можуть навчатися за допомогою алгоритмів глибокого детермінованого градієнта політики (DDPG) [28].

Також для їхнього тренування ефективно застосовуються алгоритми проксимальної оптимізації (PPO) [29]. Вони інтегруються в симуляційну платформу як самостійні пілоти і навчаються керувати транспортним засобом виключно через тисячі сеансів спроб і помилок, постійно оптимізуючи функцію винагороди. Хоча початкова мета навчання з підкріпленням майже завжди полягає у створенні безпілотних ботів-автопілотів (у змаганнях класу Autonomous Racing), проміжні результати їхнього ітеративного навчання несуть величезну теоретичну цінність для дата-аналізу реальних людей [30]. Згенеровані такими адаптивними агентами математично ідеальні мінімаксимальні швидкості, найпряміші траєкторії руху та міліметрово вивірені точки гальмування фактично виступають абсолютним еталоном. Вони формують гіпотетично досконале, хоча й часто недосяжне для фізіології живої людини найшвидше коло, з яким згодом гоночний інженер може зіставляти мікроскопічні похибки пілота.

Для вирішення задачі автоматизованого аналізу гоночної телеметрії та інтелектуальної підтримки пілота можливо і доцільно використовувати методи штучного інтелекту, зокрема штучні нейронні мережі та алгоритми ML.

Теоретично для обробки багатовимірних часових рядів, якими є телеметрія, можуть застосовуватися нейронні мережі. Наприклад, рекурентні нейронні мережі (RNN) та мережі довгої короткострокової пам'яті (LSTM) здатні

враховувати часовий контекст і виявляти відкладені наслідки дій водія (коли помилка на виході з повороту зумовлена неправильною швидкістю на вході). Зі свого боку, одновимірні згорткові мережі (1D-CNN) можуть застосовуватись для розпізнавання локальних аномалій на графіках натискання педалей. Проте використання глибоких нейромереж часто пов'язане з високими вимогами до обчислювальних ресурсів та необхідністю збору величезних масивів розмічених тренувальних даних, що може ускладнити їх локальне та автономне використання на звичайних домашніх комп'ютерах.

З точки зору науки про дані, комплексна задача аналізу телеметрії для пошуку резервів швидкості не є однорідною. Вона належить одразу до двох фундаментальних класів задач машинного навчання: регресії та класифікації. Задачі регресії виникають там, де системі необхідно спрогнозувати безперервні числові значення – наприклад, побудувати нелінійну просторову криву ідеального кола (математичний референс) або вирахувати точну кількість мілісекунд, які пілот потенційно втратив на певному мікросекторі. Задачі класифікації потрібні для семантичного розпізнавання патернів: віднесення певного фрагмента логів телеметрії до одного з наперед відомих класів помилок керування (наприклад, раннє гальмування, занадто пізнє додавання газу або раптове відпускання педалі).

Для розв'язання цих задач теоретично можна застосувати низку відомих алгоритмів машинного навчання:

- Для задач регресії: класична лінійна регресія, регресія на основі опорних векторів (SVR), а також потужні ансамблеві методи на базі дерев рішень, такі як випадковий ліс (Random Forest) та алгоритми градієнтного бустингу (Gradient Boosting).

- Для задач класифікації: логістична регресія, метод k -найближчих сусідів (k -NN), метод опорних векторів (SVM), дерева рішень (Decision Trees), а також ансамблеві класифікатори (Gradient Boosting Classifier).

Серед перелічених методів особливої уваги заслуговують ансамблеві алгоритми. Вони теоретично здатні надзвичайно швидко та ефективно обробляти

великі масиви табличних даних, виявляючи складні нелінійні залежності між фізичними показниками датчиків (швидкість, інтенсивність гальмування, кут керма) та фінальним часом проходження кола, залишаючись при цьому стійкими до шумових викидів, характерних для даних із симуляторів.

1.3 Аналіз наукових рішень у задачі аналізу телеметрії пілота автосимулятора

Задача аналізу телеметрії пілота автосимулятора є вкрай актуальною, адже з розвитком моторспорту та професійного симрейсингу обсяги даних, що потребують обробки, зростають у геометричній прогресії. Над автоматизацією роботи DE уже працювала низка дослідницьких колективів, пропонуючи підходи до виявлення помилок пілотування на основі сирих показників автомобіля.

Автори публікації розглядають задачу аналізу телеметрії для виявлення агресивних подій та відхилень від оптимального водіння [31]. Для цього дослідники застосували архітектуру довгої короткострокової пам'яті (LSTM), зокрема двонаправлену мережу з механізмами уваги (CBANet). Даний підхід дозволяє враховувати часові залежності вхідних даних швидкості, прискорення та кута кермування, автоматизовано виявляючи патерни жорсткого проходження поворотів. Завдяки цій неймережі вдалося досягти надзвичайно високої точності класифікації маневрів: метрика Ассигасу склала понад 0.98, а F1-метрика – 0.94.

У роботі дослідники розв'язували задачу аналізу телеметрії шляхом класифікації дій водія під час гальмування та розгону [32]. Для цього було використано алгоритми класичного машинного навчання, а саме метод випадкового лісу (Random Forest), який аналізував показники тиску на гальма (brake pressure) та зміну швидкості. Наведений підхід дозволив розробити систему, яка ефективно відрізняє небезпечні чи помилкові втручання від нормальної поведінки за кермом. Згідно з результатами, алгоритм Random Forest

перевершив класичний метод опорних векторів (SVM) та продемонстрував метрику Accuracy на рівні 0.95.

Група авторів у статті досліджувала задачу аналізу телеметрії для визначення загального стилю водіння на коротких відрізках часу [33]. Для класифікації цих часових патернів було застосовано згорткову нейронну мережу (CNN). Авторам вдалося перетворити одновимірні сигнали телеметрії (положення дроселя, швидкість, оберти двигуна, поздовжнє та поперечне прискорення) на структуровані матриці даних для подальшої обробки. Запропонований перехід до згорткових архітектур дав змогу ефективно виділяти аномальні втручання пілота з точністю (Accuracy) на рівні 0.92.

Застосування ансамблевих методів у цій предметній області також підтверджується дослідженнями у сфері професійного автоспорту. Зокрема, у дослідженні розглядається задача аналізу телеметрії для прийняття стратегічних рішень під час заїздів Формули-1 [34]. Дослідники довели, що використання алгоритмів градієнтного бустингу (Gradient Boosting Regressor) для обробки історичної телеметрії болідів є високоефективним для прогнозування падіння темпу. Наведені прецеденти доводять стратегічну доцільність розробки самодостатніх інтелектуальних систем для аналізу телеметричних даних.

Підсумовуючи огляд наявних наукових рішень, можна помітити, що для розв'язання задачі аналізу телеметрії більшість дослідників застосовують важкі алгоритми глибокого навчання (LSTM, CNN) або масивні ансамблі дерев (Random Forest). Основним недоліком таких підходів є або надто висока обчислювальна складність, що унеможливорює швидку локальну роботу системи на домашньому комп'ютері симрейсера без доступу до хмарних серверів, або ж надмірна вузькоспеціалізованість (алгоритм знаходить лише один вузький тип відхилень). Таким чином, основною проблематикою на даному етапі розвитку предметної області залишається відсутність комплексних і водночас ресурсно-оптимізованих рішень, які могли б забезпечити стабільно високу точність

виявлення широкого спектра водійських помилок, працюючи при цьому повністю автономно.

Підсумовуючи огляд наявних наукових рішень, можна помітити, що більшість дослідників для розв'язання задачі аналізу телеметрії застосовують важкі алгоритми глибокого навчання (LSTM, 1D-CNN) або класичні ансамблі дерев (Random Forest) (таблиця 1.1).

Таблиця 1.1 – Узагальнюючий аналіз публікацій

Посилання на статтю	Засоби	Результати
Alhamdan, H., et al. [31]	Глибоке навчання: Двонаправлена довга короткострокова пам'ять (BiLSTM) у поєднанні зі згортковою нейромережею (CNN) та механізмами уваги.	Високоточне виявлення патернів жорсткого водіння. Accuracy: понад 0.98, F1-метрика: 0.98,0.94.
Abdullah, M., et al. [32]	Класичне машинне навчання: Ансамблевий метод випадкового лісу (Random Forest).	Ефективне розпізнавання помилкових втручань на гальмуванні та розгоні. Accuracy: 0.95 (перевершено показники класичного SVM).
Alotaibi, M., et al. [33]	Глибоке навчання: Згорткова нейронна мережа (CNN) адаптована для одновимірних часових рядів.	Успішна класифікація загального стилю водіння на коротких відрізках. Accuracy: 0.92.
Heil, J., et al. [34]	Машинне навчання: Алгоритми градієнтного бустингу (Gradient Boosting Regressor) та рекурентні нейронні мережі.	Точний розрахунок стратегії та прогнозування падіння темпу боліда.

Основним недоліком таких підходів є або надто висока обчислювальна складність, що унеможливорює швидку локальну роботу системи на домашньому комп'ютері симрейсера без доступу до хмарних обчислень, або ж надмірна вузькоспеціалізованість. У багатьох існуючих рішеннях алгоритм здатний знаходити лише один вузький тип відхилень (наприклад, винятково помилки на гальмуванні), але повністю ігнорує загальний час кола, аеродинамічну стабільність автомобіля чи контекст налаштувань підвіски. Окрім того, загальні показники точності та F1-метрики у проаналізованих системах найчастіше коливаються в межах 0.95, залишаючи простір для хибних спрацьовувань алгоритму. Таким чином, основною проблематикою на даному етапі розвитку предметної області залишається відсутність комплексних і водночас ресурсно-оптимізованих рішень, які могли б забезпечити стабільно високу точність виявлення широкого спектра водійських помилок, працюючи при цьому повністю автономно без залучення зовнішніх серверів.

1.4 Аналіз існуючих програмних засобів

На сьогодні індустрію програмного забезпечення, що спеціалізується на обробці телеметрії гоночних автомобілів, можна глобально поділити на дві категорії: цільові засоби для глибокої інженерної діагностики та споживчі платформи, орієнтовані на тренування (коучинг) водіїв. Доцільним прикладом першої категорії виступає система MoTeC i2 Pro, яка фактично має статус індустріального стандарту як у класичному моторспорті, так і у віртуальних перегонах [35].

Вона надає надзвичайно гнучкий інструментарій для запису сотень мікропараметрів автомобіля, створення кастомних математичних каналів та обробки високорівневих графічних звітів (рис. 1.1). Беззаперечною перевагою цієї програми є безмежний потенціал для діагностики будь-якого компонента боліда. Проте її недоліком залишається надмірно високий поріг входження: система виступає виключно візуалізатором і зовсім не містить модулів штучного інтелекту для автоматичних підказок, повністю покладаючи тягар розпізнавання помилок на експертизу інженера.

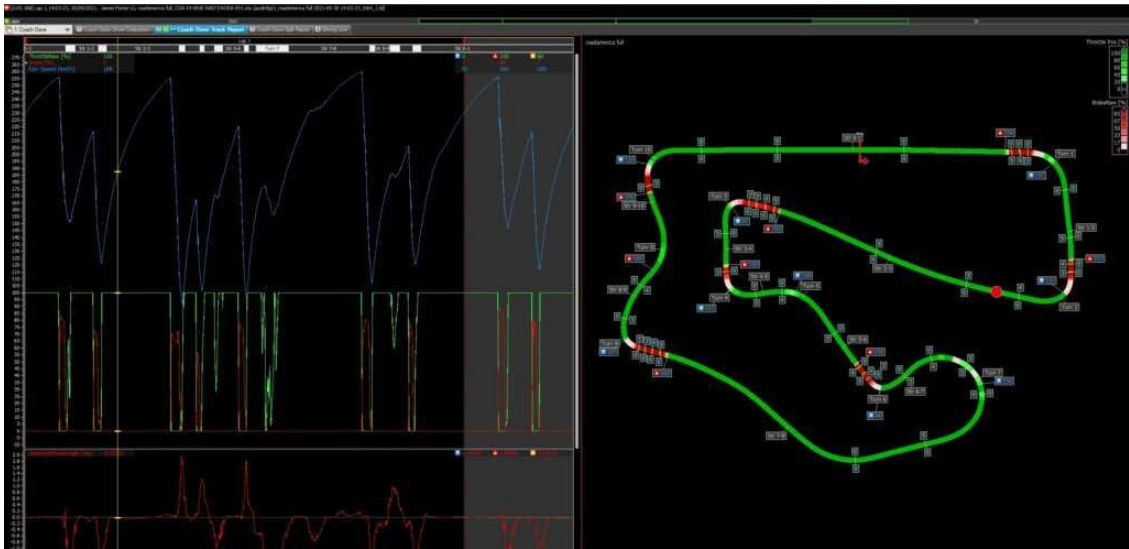


Рисунок 1.1 – Зображення інтерфейсу MoTeC i2 Pro [35]

Більш сучасний підхід, спрямований саме на автоматизацію процесу поліпшення навичок, реалізовано у хмарному сервісі Virtual Racing School (VRS) [36]. Його концепція полягає у наданні симрейсерам доступу до еталонних телеметричних пакетів (рис. 1.2), записаних професійними інструкторами.

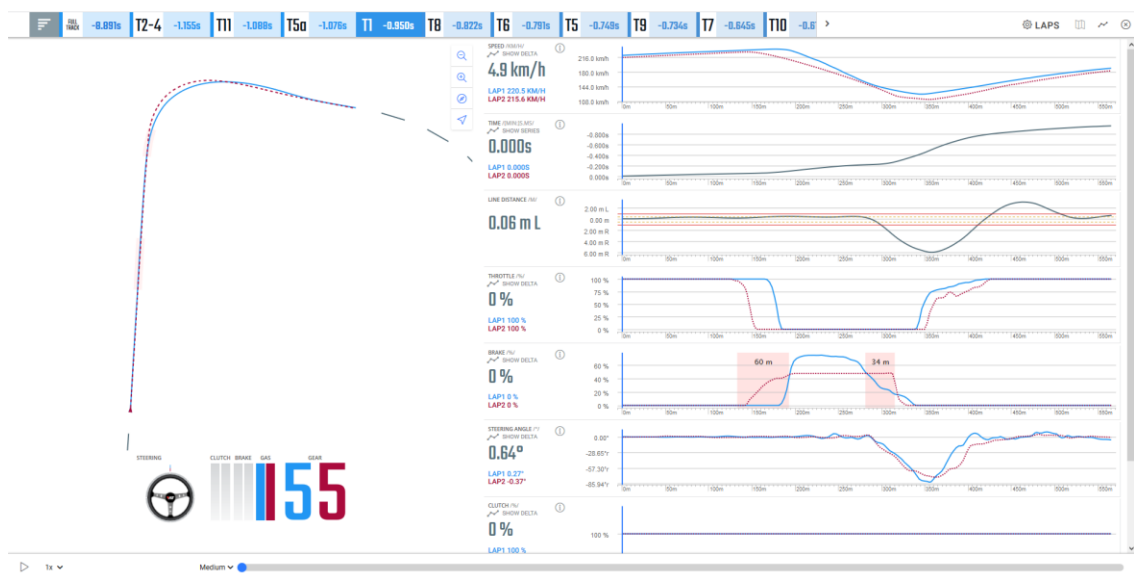


Рисунок 1.2 – Зображення інтерфейсу сервісу VRS [36]

Через браузер користувач отримує змогу наочно зіставляти свої точки гальмування та швидкісні показники з ідеальним проходженням. Перевагами VRS є інтуїтивна зрозумілість та надання чітких еталонних орієнтирів. Водночас суттєвими мінусами рішення залишаються закритість екосистеми, фінансова залежність від платних підписок на обрані автомобілі, а також відсутність глибокого автоматизованого ШІ-аналізу конкретних хибних рухів під час заїзду.

Одним із найбільш технологічних рішень із застосуванням штучного інтелекту є сервіс Track Titan, що базується на фірмовому алгоритмі Morpheus Engine [37]. Програма діє як віртуальний тренер: самостійно ділить коло на окремі мікросектори і генерує текстові інструкції для нівелювання помилок, виявлених на основі аналізу відхилень від ідеальної траєкторії (рис. 1.3).

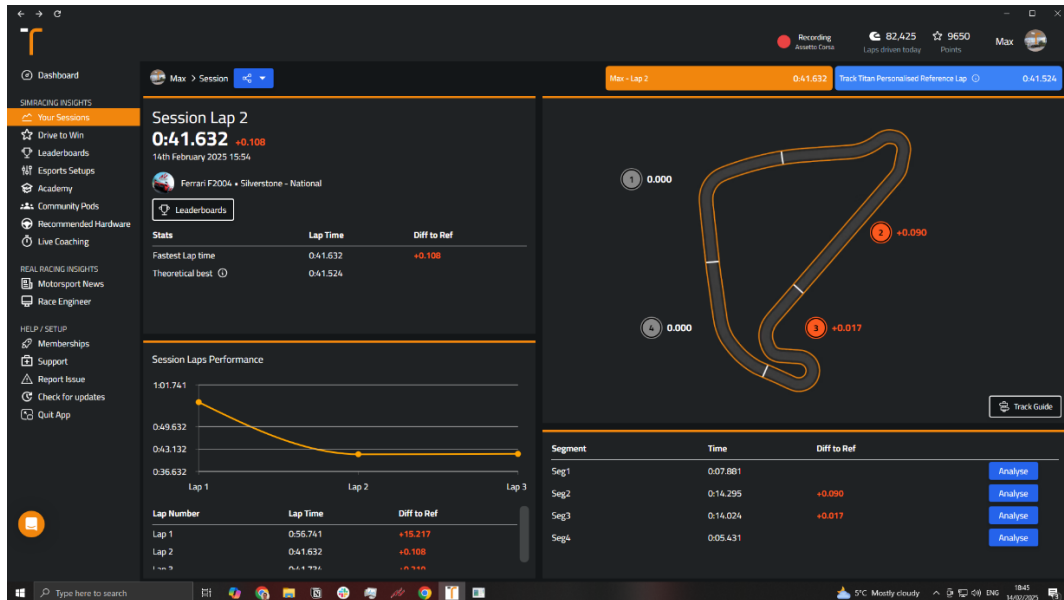


Рисунок 1.3 – Зображення інтерфейсу Track Titan [37]

Великим плюсом цього продукту є видача оперативної аналітики відразу після сесії без потреби маніпулювати складними графіками. Серед слабких сторін слід відзначити її веб-орієнтованість, що вимагає безперервного хмарного з'єднання для транспортування даних, а також надмірну стандартизацію порад, які не завжди враховують індивідуальну специфіку налаштувань (сетапу) боліда.

Підсумовуючи огляд наявних програмних рішень для аналізу гоночної телеметрії, можна виділити ряд недоліків, притаманних сучасним продуктам. З одного боку, потужні інженерні інструменти характеризуються надмірно високим порогом входження та відсутністю автоматизованих інтелектуальних підказок. З іншого боку, популярні платформи для коучингу здебільшого є комерційно закритими, залежать від стабільного інтернет-з'єднання для виконання хмарних обчислень та генерують занадто узагальнені поради, які часто ігнорують специфічні налаштування автомобіля. Відповідно, на ринку існує гостра нестача автономних систем, здатних локально та швидко обробляти об'ємні бінарні логи без мережових затримок, при цьому поєднуючи глибокий інженерний аналіз із дружнім ШІ-супроводом. Саме ці функціональні та

архітектурні прогалини створюють перспективу для проектування нового програмного підходу у форматі локального десктопного додатка на базі Python. Усунення перелічених недоліків та створення незалежного, конфіденційного інструменту експертного рівня для симрейсерів є тим вектором розробок, який імплементуватиметься у даній роботі.

1.5 Мета, задачі та вимоги до реалізації інтелектуальної системи

Метою кваліфікаційної роботи є підвищення ефективності керування транспортним засобом у гоночному симуляторі за рахунок підвищення точності автоматичного виявлення помилок пілотування, підвищення точності прогнозування пов'язаних із ними часових втрат та покращення результатів проходження контрольного кола шляхом формування персоналізованих рекомендацій на основі аналізу телеметричних даних із використанням методів машинного навчання.

Для досягнення мети потрібно виконати наступні задачі:

- виконати аналіз предметної області, дослідивши сучасні підходи до обробки телеметричних даних в автоспорті та методи інтелектуальної підтримки пілотів у гоночних симуляторах;
- формалізувати задачу підтримки пілота автосимулятора на основі аналізу телеметрії;
- розробити метод інтелектуальної підтримки пілота автосимулятора на основі аналізу телеметрії з використанням засобів машинного навчання;
- реалізувати інтелектуальну систему підтримки пілота автосимулятора, що базується на розробленому методі;
- провести експериментальне дослідження запропонованого методу на основі телеметричних даних гоночного симулятора з оцінюванням точності автоматичного виявлення помилок пілотування, точності прогнозування пов'язаних із ними часових втрат та впливу запропонованого підходу на результати проходження контрольного кола.

Розділ 2 Метод інтелектуальної підтримки пілота автосимулятора на основі аналізу телеметрії з використанням засобів машинного навчання

2.1 Постановка задачі підтримки пілота автосимулятора на основі аналізу телеметрії

Головною метою системи інтелектуальної підтримки є підвищення ефективності керування транспортним засобом у гоночному симуляторі, а саме мінімізація часу проходження кола пілотом автосимулятора шляхом автоматизованого виявлення неоптимальних патернів керування та формування персоналізованих рекомендацій. Розв'язання цієї задачі базується на конвеєрі, що математично формалізується як композиція відображень (рисунок 2.1).



Рисунок 2.1 – Загальна схема конвеєра

Першим етапом є екстракція сигналів $\mathbf{x}(pos)$, що являють собою безперервний потік телеметричних даних як функції нормалізованої позиції на трасі. Наступним кроком є застосування оператора формування ознак $\Phi(\cdot)$, що перетворює сигнали та їх локальні відхилення у вектор ознак F . Далі застосовуються оптимізаційні моделі машинного навчання $H(F)$, що апроксимують еталонну траєкторію та розпізнають патерни помилок. Кінцевим

етапом є функція прийняття рішень $D(H)$, що здійснює перехід від ймовірнісних або неперервних прогнозів до семантичних висновків.

Основним незалежним аргументом системи є нормалізована позиція на трасі $pos \in [0,1]$. Відповідно до формули 2.1, стан автомобіля у цій точці задається вектором:

$$x_{pos} = [v(pos), gas(pos), brake(pos), steer(pos)]^T, \quad (2.1)$$

де x_{pos} – вектор стану автомобіля; $v(pos) \in \mathbb{R}^+$ – швидкість, км/год; $gas(pos), brake(pos) \in [0,1]$ – нормовані рівні натискання педалей акселератора та гальма; $steer(pos) \in \mathbb{R}$ – кут повороту рульового колеса, градуси.

Для аналізу локальної динаміки керування формується матриця стану в межах вікна ковзання розмірністю W як показано у формулі 2.2:

$$X_w = [x_{pos_1}, x_{pos_2}, \dots, x_{pos_W}] \in \mathbb{R}^{4 \times W}, \quad (2.2)$$

де X_w – матриця стану у вікні ковзання; x_{pos_i} – вектор стану в i -й точці просторового вікна; W – розмірність вікна ковзання.

Оператор екстракції ознак $\Phi: \mathbb{R}^{4 \times W} \times \mathbb{R}^{4 \times W} \rightarrow \mathbb{R}^{30}$ перетворює матрицю відхилень між фактичною та еталонною телеметрією $\Delta X_w = X_w^{user} - X_w^{ref}$ у вектор ознак $F \in \mathbb{R}^{30}$. Вектор F акумулює статистичні моменти: локальні дельти, стандартне відхилення σ , математичне сподівання μ , екстремуми та коефіцієнти лінійного тренду у межах вікна.

Кінцевий результат аналізу сегмента траси формалізується як кортеж виходів \mathcal{O} , що відображено у формулі 2.3.:

$$\mathcal{O} = \langle c, S, \Delta t, M_{txt} \rangle, \quad (2.3)$$

де \mathcal{O} – кортеж виходів; $c \in \mathcal{C}$ – клас відхилення, $\mathcal{C} = \{optimal, speed_loss, brake_early, gas_late, steer_error\}$; $S \in [0,1]$ – неперервна метрика тяжкості помилки; $\Delta t \in \mathbb{R}^+$ – оціночна втрата часу, мс; M_{txt} – текстовий дескриптор-рекомендація.

Загальна задача інтелектуальної підтримки декомпозується на три оптимізаційні підзадачі.

Першою підзадачею є класифікація типу помилки керування. Вона формалізується як пошук оптимального відображення $f_c^*: \mathbb{R}^{30} \rightarrow C$, що мінімізує емпіричний ризик (наприклад, крос-ентропію) на навчальній вибірці, згідно з формулою 2.4:

$$f_c^* = \arg \min_{f_c} \frac{1}{N} \sum_{i=1}^N \mathcal{L}_{CE} \left(f_c(F^{(i)}), c_{true}^{(i)} \right), \quad (2.4)$$

де f_c^* – оптимальне відображення класифікатора; N – об'єм навчальної вибірки; \mathcal{L}_{CE} – функція втрат крос-ентропії; $F^{(i)}$ – вектор ознак i -го екземпляра; $c_{true}^{(i)}$ – істинний клас помилки i -го екземпляра.

Другою підзадачею є регресійний аналіз, що складається з оцінки тяжкості помилки та апроксимації еталонної телеметрії. Оцінка тяжкості формалізується як пошук функції $h_s^*: \mathbb{R}^{30} \rightarrow [0,1]$, що мінімізує середньоквадратичне відхилення (MSE), згідно з формулою 2.5:

$$h_s^* = \arg \min_{h_s} \frac{1}{N} \sum_{i=1}^N \left(h_s(F^{(i)}) - S_{true}^{(i)} \right)^2, \quad (2.5)$$

де h_s^* – оптимальна регресійна функція оцінки тяжкості; N – об'єм навчальної вибірки; $F^{(i)}$ – вектор ознак i -го екземпляра; $S_{true}^{(i)}$ – істинне значення тяжкості помилки i -го екземпляра.

Аналогічно, еталонний вектор стану x^{ref} апроксимується нелінійною регресійною функцією від збагаченого просторового вектора $p = [pos, pos^2, \sin(2\pi pos), \cos(2\pi pos)]^T$.

Третьою підзадачею є формування рекомендацій та оцінка втраченого часу. Цей детермінований етап реалізує відображення $D: C \times [0,1] \rightarrow M_{txt}$. Оцінка втрати часу Δt апроксимується за допомогою емпіричної формули інтегральної втрати швидкості (2.6):

$$T_{loss} = \Delta pos \cdot T_{lap} \cdot \left(\frac{\overline{\Delta v}}{\max(\overline{v}_{ref}, 30)} \right) \cdot 2.0, \quad (2.6)$$

де T_{loss} – оціночна втрата часу (Δt); Δpos – відносна довжина сегмента; T_{lap} – загальний час кола; $\overline{\Delta v}$ – середня втрата швидкості; \overline{v}_{ref} – середня еталонна швидкість.

Зведений формалізований псевдокод загальної постановки задачі наведено нижче. Цей алгоритм описує концептуальний рівень інформаційної взаємодії трьох зазначених підзадач.

Алгоритм 2.1 – Концептуальна схема інтелектуальної підтримки пілота

Вхід: потік просторово-синхронізованих векторів стану \mathbf{x}_{pos}

Вихід: масив кортежів \mathcal{O}

1. ініціалізувати масив результатів $\mathcal{O} \leftarrow \emptyset$
2. для кожного сегмента траси pos :
3. // Підзадача 1: Екстракція ознак та класифікація
4. виділити матрицю локальних відхилень $\Delta\mathbf{X}$
5. сформувані ознаки $F \leftarrow \Phi(\Delta\mathbf{X})$
6. розпізнати тип помилки $c \leftarrow f_c^*(F)$
7. якщо c дорівнює optimal:
8. продовжити перевірку наступного сегмента
9. // Підзадача 2: Регресійне оцінювання тяжкості
10. оцінити метрику тяжкості $S \leftarrow h_S^*(F)$
11. // Підзадача 3: Оцінка часу та формування рекомендацій
12. оцінити втрату часу $\Delta t \leftarrow T_{loss}$
13. згенерувати текстову пораду $M_{txt} \leftarrow D(c, S)$
14. додати $\langle c, S, \Delta t, M_{txt} \rangle$ до \mathcal{O}
15. повернути масив \mathcal{O}

Отже, наведена формалізація подає задачу інтелектуальної підтримки як замкнений цикл відображень: від реєстрації матриці сигналів через мінімізацію функцій просторових та статистичних втрат до генерування детермінованих алгоритмічних вказівок для оптимізації швидкісних показників пілота.

2.2 Формування простору ознак телеметричних даних

Телеметричні дані пілота автосимулятора формують безперервний просторовий ряд. В рамках загальної інформаційної парадигми сигнали \rightarrow

ознаки \rightarrow моделі \rightarrow рішення, точкового аналізу окремих векторів стану є принципово недостатньо для класифікації помилок. Помилки керування (наприклад, раннє гальмування чи втрата швидкості) мають розподілену в просторі динаміку. Алгоритмічна доцільність застосування методу ковзного вікна полягає у трансформації безперервного просторового потоку телеметрії змінної довжини у множину дискретних векторів ознак строго фіксованої розмірності. Це є обов'язковою математичною умовою для їх подальшої передачі на вхід оптимізаційним моделям машинного навчання (класифікаторам та регресорам).

2.2.1 Метод ковзного вікна для обробки телеметричних сигналів

Процес екстракції ознак починається з переходу від абсолютних сигналів до просторового ряду локальних відхилень. Відповідно до формули 2.7, для кожної дискретної точки k формується вектор із п'яти просторових дельт:

$$\Delta x_k = [\Delta v_k, \Delta gas_k, \Delta brake_k, \Delta steer_k, \Delta |steer|_k]^T, \quad (2.7)$$

де Δx_k – вектор відхилень у точці k ; Δv_k – відхилення швидкості; Δgas_k – відхилення рівня натискання педалі газу; $\Delta brake_k$ – відхилення рівня натискання педалі гальма; $\Delta steer_k$ – відхилення кута повороту керма; $\Delta |steer|_k$ – різниця абсолютних значень кута повороту керма.

Нехай $D = \{\Delta x_k\}_{k=1}^N$ – повна послідовність векторів відхилень на аналізованому сегменті траси. Розбиття цієї послідовності методом ковзного вікна формалізується як генерація підмножин векторів. Згідно з формулою 2.8, j -те вікно ковзання визначається матрицею локальних відхилень:

$$\Delta X_j = [\Delta x_{j,s}, \Delta x_{j,s+1}, \dots, \Delta x_{j,s+W-1}] \in \mathbb{R}^{5 \times W}, \quad (2.8)$$

де ΔX_j – j -та матриця стану у вікні ковзання; j – індекс вікна, $j \in \{0, 1, \dots, \lfloor \frac{N-W}{s} \rfloor\}$; W – розмірність вікна ковзання ($W = 15$); s – крок зсуву вікна ($s = 10$).

Коефіцієнт перекриття суміжних вікон є гіперпараметром, що забезпечує безперервність аналізу та стійкість до зсувів сигналу. Цей коефіцієнт визначається співвідношенням 2.9:

$$O_{perc} = \frac{W-s}{W} \times 100\%, \quad (2.9)$$

де O_{perc} – відсоток просторового перекриття суміжних вікон. Для заданих параметрів імплементованої системи перекриття становить 33.3%.

На фінальному етапі матриця ΔX_j піддається детермінованій статистичній редукції для формування вектора ознак фіксованої довжини $F_j \in \mathbb{R}^{30}$. Оператор екстракції ознак $\Phi(\cdot)$ обчислює статистичні моменти вздовж просторової осі вікна:

$$F_j = \Phi(\Delta X_j) = \left[\Delta x_c^T, \mu(\Delta X_j)^T, \sigma(\Delta X_j)^T, \min(\Delta X_j)^T, \max(\Delta X_j)^T, a(\Delta X_j)^T \right]^T, \quad (2.10)$$

де F_j – результуючий вектор ознак j -го вікна; c – індекс центральної точки вікна, $c = j \cdot s + \lfloor W/2 \rfloor$; Φ – оператор екстракції ознак; μ, σ, \min, \max – вектори відповідних статистичних моментів; a – вектор коефіцієнтів лінійного тренду.

Зведений формалізований псевдокод процесу дискретизації наведено нижче.

Алгоритм 2.2 – Дискретизація телеметрії методом ковзного вікна

Вхід: послідовність локальних відхилень $\mathbf{D} = \{\Delta \mathbf{x}_k\}_{k=1}^N$, розмір вікна W , крок зсуву s

Вихід: множина векторів ознак \mathcal{F}

1. ініціалізувати вихідну множину $\mathcal{F} \leftarrow \emptyset$
2. $j \leftarrow 0$
3. поки $(j \cdot s + W) \leq N$:
4. // Формування просторової матриці для поточного вікна
5. ініціалізувати матрицю $\Delta \mathbf{X}_j \leftarrow$ масив розміру $5 \times W$
6. для i від 0 до $W - 1$:
7. $\Delta \mathbf{X}_j[\text{стовпець } i] \leftarrow \Delta \mathbf{x}_{j \cdot s + i}$
8. // Статистична редукція матриці у вектор
9. $F_j \leftarrow \Phi(\Delta \mathbf{X}_j)$
10. додати F_j до \mathcal{F}
11. $j \leftarrow j + 1$
12. повернути \mathcal{F}

Згенерована множина векторів $\mathcal{F} = \{F_j\}$ інтегрується у загальний пайплайн як стандартизований математичний простір для ансамблевих моделей машинного навчання. Детальний математичний апарат формування диференціальних та статистичних компонентів оператора Φ розглядається у підрозділі 2.2.2.

2.2.2 Статистичне та диференціальне формування вектора ознак

Для ефективного навчання ансамблевих моделей багатовимірна матриця локальних відхилень трансформується у єдиний одновимірний вектор ознак. Цей процес поєднує дві комплементарні групи показників: диференціальні компоненти, що фіксують миттєвий стан автомобіля, та статистичні моменти, які описують еволюцію керування у просторовому вікні.

Диференціальна група ознак формується шляхом прямої екстракції вектора відхилень з центральної точки вікна. Це дозволяє моделі оцінити фактичний миттєвий розрив між еталонною та реальною траєкторією в момент передбачуваної помилки. Математичний опис цього компонента наведено у формулі 2.11:

$$f_{diff} = \Delta x_c = [\Delta v_c, \Delta gas_c, \Delta brake_c, \Delta steer_c, \Delta |steer|_c]^T, \quad (2.11)$$

де f_{diff} – вектор диференціальних ознак; Δx_c – вектор просторових дельт у центрі вікна; c – просторовий індекс центральної точки.

Друга група ознак обчислюється незалежно для кожного з п'яти телеметричних каналів шляхом агрегації значень уздовж просторової осі вікна. Нехай $v \in \mathbb{R}^W$ – вектор-рядок матриці вікна, що відповідає одному з п'яти параметрів відхилення. Для оцінки загального зсуву керування та його дисперсії застосовуються математичне сподівання та стандартне відхилення, як показано у формулах (2.12) та (2.13):

$$\mu(v) = \frac{1}{W} \sum_{k=1}^W v_k, \quad (2.12)$$

де $\mu(v)$ – математичне сподівання каналу відхилень; W – розмірність просторового вікна; v_k – локальне значення відхилення.

$$\sigma(v) = \sqrt{\frac{1}{W} \sum_{k=1}^W (v_k - \mu(v))^2}, \quad (2.13)$$

де $\sigma(v)$ – стандартне відхилення; W – розмірність просторового вікна; v_k – локальне значення відхилення; $\mu(v)$ – математичне сподівання.

Для фіксації критичних пікових значень помилки в межах вікна обчислюються глобальні екстремуми локальної вибірки, як показано у формулі 2.14:

$$e_{min} = \min_{1 \leq k \leq W} v_k, \quad e_{max} = \max_{1 \leq k \leq W} v_k, \quad (2.14)$$

де e_{min} – мінімальне пікове відхилення; e_{max} – максимальне пікове відхилення; W – розмірність просторового вікна; v_k – значення відхилення.

Важливим динамічним індикатором є коефіцієнт лінійного тренду, який дозволяє моделі розпізнати, чи помилка наростає, чи водій вже почав її коригувати. Згідно з формулою 2.15, цей показник визначається як нахил прямої, отриманої шляхом апроксимації методом найменших квадратів:

$$a(v) = \frac{\sum_{k=1}^W (k - \bar{k})(v_k - \mu(v))}{\sum_{k=1}^W (k - \bar{k})^2}, \quad (2.15)$$

де $a(v)$ – коефіцієнт лінійного тренду (нахил); k – локальний просторовий індекс; \bar{k} – середнє значення локального індексу; v_k – значення відхилення; $\mu(v)$ – математичне сподівання.

Кінцевий вектор ознак формується шляхом конкатенації диференціального вектора із п'ятьма розрахованими статистичними метриками для кожного з п'яти телеметричних каналів. Математично цей процес описується співвідношенням 2.16:

$$F_j = \left[f_{diff}^T, \mu(\Delta X_j)^T, \sigma(\Delta X_j)^T, e_{min}^T, e_{max}^T, a(\Delta X_j)^T \right]^T, \quad (2.16)$$

де $F_j \in \mathbb{R}^{30}$ – результуючий вектор ознак j -го вікна.

Завдяки такій ортогональній репрезентації результуюча двовимірна матриця відхилень стискається у високоінформативний простір. Це створює надійний математичний фундамент для безпосереднього навчання ансамблевих алгоритмів розпізнавання патернів керування.

Алгоритмічна реалізація оператора екстракції ознак наведена нижче.

Алгоритм 2.3 – Формування просторово-статистичного вектора Φ

Вхід: матриця локальних відхилень у вікні $\Delta X_j \in \mathbb{R}^{5 \times W}$ Вихід: вектор ознак $F_j \in \mathbb{R}^{30}$

1. $c \leftarrow \lfloor W/2 \rfloor$ // індекс центральної точки
2. витягти диференціальний вектор $f_{diff} \leftarrow \Delta X_j[\text{всі рядки}, c]$
3. ініціалізувати порожній вектор статистик $S_{stat} \leftarrow [F_j]$
4. для кожного каналу відхилень $v \in \Delta X_j$ (від 1 до 5):
5. обчислити математичне сподівання $\mu \leftarrow \text{mean}(v)$
6. обчислити стандартне відхилення $\sigma \leftarrow \text{std}(v)$
7. знайти мінімальне пікове відхилення $e_{min} \leftarrow \min(v)$
8. знайти максимальне пікове відхилення $e_{max} \leftarrow \max(v)$
9. обчислити коефіцієнт нахилу тренду $a \leftarrow \text{slope}(v)$
10. додати масив $[\mu, \sigma, e_{min}, e_{max}, a]$ до вектора S_{stat}
11. $F_j \leftarrow \text{конкатенація}(f_{diff}, S_{stat})$
12. повернути F_j

Завдяки такій ортогональній репрезентації результуюча двовимірна матриця відхилень стискається у високоінформативний простір. Це створює надійний математичний фундамент для безпосереднього навчання ансамблевих алгоритмів розпізнавання патернів керування.

2.3 Задача класифікації типів помилок пілотування

Ключовим завданням діагностичного модуля є ідентифікація типу неоптимального керування на основі екстрагованих просторово-статистичних ознак. Задача формалізується як дискретне відображення простору ознак \mathbb{R}^{30} у скінченну множину цільових класів. Простір рішень складається з п'яти взаємовиключних категорій, що відображено у формулі 2.17:

$$C = \{c_1, c_2, c_3, c_4, c_5\}, \quad (2.17)$$

де C – множина класів керування; c_1 (`optimal`) – еталонне проходження; c_2 (`speed_loss`) – втрата швидкості; c_3 (`brake_early`) – передчасне гальмування; c_4 (`gas_late`) – запізніле прискорення; c_5 (`steer_error`) – просторова помилка рулювання.

Математична ціль полягає у знаходженні оптимального відображення $f_c: \mathbb{R}^{30} \rightarrow C$. Для апроксимації f_c застосовується ансамблевий алгоритм випадкового лісу (Random Forest), що складається з множини M незалежних дерев рішень $\{T_1, T_2, \dots, T_M\}$. Рішення кожної базової моделі ґрунтуються на максимізації апостеріорної ймовірності, як показано у формулі 2.18:

$$T_m(F) = \arg \max_{c \in C} P_m(c|F), \quad (2.18)$$

де $T_m(F)$ – прогноз m -го дерева; $F \in \mathbb{R}^{30}$ – вхідний вектор ознак; $P_m(c|F)$ – оцінена m -им деревом ймовірність класу c .

Побудова кожного дерева в ансамблі базується на рекурсивному розбитті простору ознак. Критерієм поділу у вузлах виступає приріст інформації, що обчислюється через ентропію Шеннона для розподілу п'яти класів. Математично цей розрахунок описується формулою 2.19:

$$H(S) = - \sum_{i=1}^5 p_i \log_2(p_i), \quad (2.19)$$

де $H(S)$ – ентропія Шеннона; S – локальна множина навчальних екземплярів у вузлі; p_i – частка екземплярів класу c_i у множині S .

Для глобального оцінювання ймовірнісної роботи класифікатора під час навчання та оптимізації гіперпараметрів функцією втрат виступає категоріальна крос-ентропія. Математично ця функція описується співвідношенням 2.20:

$$\mathcal{L}_{CE} = - \frac{1}{N} \sum_{j=1}^N \sum_{i=1}^5 y_{j,i} \log(P(c_i|F^{(j)})), \quad (2.20)$$

де \mathcal{L}_{CE} – функція втрат крос-ентропії; N – об'єм вибірки; $y_{j,i}$ – бінарний індикатор приналежності j -го екземпляра до класу c_i ; $P(c_i|F^{(j)})$ – ансамблева ймовірність приналежності екземпляра до c_i .

Кінцевий семантичний висновок алгоритму формується шляхом строгої агрегації рішень базових дерев за правилом мажоритарного голосування. Ця операція формалізується як максимізація суми індикаторних функцій, як показано у формулі 2.21:

$$\hat{c} = f_c(F) = \arg \max_{c \in C} \sum_{m=1}^M I(T_m(F) = c), \quad (2.21)$$

де \hat{c} – кінцевий прогнозований клас; M – кількість дерев в ансамблі; $I(\cdot)$ – індикаторна функція, що набуває значення 1 при істинності умови та 0 інакше; $T_m(F)$ – висновок m -го дерева.

Алгоритмічна реалізація процесу класифікації наведена нижче.

Алгоритм 2.4 – Класифікація типу помилки керування ансамблем дерев

Вхід: вектор ознак $F \in \mathbb{R}^{30}$, ансамбль із M дерев рішень $\{T_1, \dots, T_M\}$, множина класів $C = \{c_1, c_2, c_3, c_4, c_5\}$

Вихід: прогнозований клас \hat{c}

1. ініціалізувати масив лічильників $Votes \leftarrow [0,0,0,0,0]$
2. для m від 1 до M :
3. // Отримання локального прогнозу від m -го дерева
4. $c^{(m)} \leftarrow T_m(F)$
5. знайти індекс i , такий що $c_i = c^{(m)}$
6. // Інкремент голосу (індикаторна функція)
7. $Votes[i] \leftarrow Votes[i] + 1$
8. // Мажоритарне голосування
9. $max_votes \leftarrow -1$
10. $\hat{c} \leftarrow c_1$
11. для i від 1 до 5:
12. якщо $Votes[i] > max_votes$:
13. $max_votes \leftarrow Votes[i]$
14. $\hat{c} \leftarrow c_i$
15. повернути \hat{c}

Такий підхід на основі мажоритарного голосування випадкового лісу дозволяє значно знизити дисперсію прогнозів та підвищити стійкість системи до локальних шумів у телеметричних даних.

2.4 Задача регресійного оцінювання часових втрат

Окрім дискретної категоризації поведінки пілота, критично важливим аспектом діагностики є кількісна оцінка наслідків помилки. Ця підзадача формалізується як проблема нелінійної множинної регресії, де цільовою змінною виступає неперервна метрика локальної втрати часу, як показано у формулі 2.22:

$$\Delta t = f_r(F), \quad (2.22)$$

де Δt – оціночна втрата часу на локальному сегменті траси, виражена у мілісекундах ($\Delta t \in \mathbb{R}^+$); f_r – неперервна регресійна функція-відображення; $F \in \mathbb{R}^{30}$ – вхідний вектор просторово-статистичних ознак.

Для апроксимації цільової функції f_r застосовується алгоритм градієнтного бустингу. Цей метод формує фінальну модель як адитивну композицію з M слабких базових регресорів (дерев рішень), мінімізуючи емпіричний ризик через обчислення антиградієнта у функціональному просторі. Оскільки цільовий простір є неперервним, цільовою функцією для оптимізації виступає середньоквадратична помилка MSE, як показано у формулі 2.23:

$$\mathcal{L}_{MSE} = \frac{1}{N} \sum_{i=1}^N \left(f_r(F^{(i)}) - \Delta t_{true}^{(i)} \right)^2, \quad (2.23)$$

де \mathcal{L}_{MSE} – функція втрат середньоквадратичного відхилення; N – загальний об'єм навчальної вибірки; $f_r(F^{(i)})$ – передбачена ансамблем втрата часу для i -го екземпляра; $\Delta t_{true}^{(i)}$ – істинне еталонне значення втрати часу для i -го екземпляра.

Математична побудова моделі починається з ініціалізації нульового наближення константою, що мінімізує початкову функцію втрат, як показано у формулі 2.24:

$$f_0(F) = \arg \min_{\gamma} \sum_{i=1}^N \mathcal{L}_{MSE} \left(\Delta t_{true}^{(i)}, \gamma \right), \quad (2.24)$$

де $f_0(F)$ – початкова базова константа ансамблю; γ – константне значення прогнозу; $\Delta t_{true}^{(i)}$ – істинне значення; \mathcal{L}_{MSE} – функція втрат.

На кожній наступній ітерації алгоритм обчислює псевдозалишки (антиградієнти) для кожного екземпляра вибірки, як показано у формулі 2.25:

$$r_{i,m} = - \left[\frac{\partial \mathcal{L}_{MSE}(\Delta t_{true}^{(i)}, f(F^{(i)}))}{\partial f(F^{(i)})} \right]_{f=f_{m-1}}, \quad (2.25)$$

де $r_{i,m}$ – псевдозалишок для i -го екземпляра на m -му кроці; f_{m-1} – композиція ансамблю, побудованого на кроці $m-1$.

Далі на згенерованих псевдозалишках навчається нове базове дерево, після чого визначається оптимальна довжина кроку у напрямку мінімізації втрат, як показано у формулі 2.26:

$$\gamma_m = \arg \min_{\gamma} \sum_{i=1}^N \mathcal{L}_{MSE}(\Delta t_{true}^{(i)}, f_{m-1}(F^{(i)}) + \gamma h_m(F^{(i)})), \quad (2.26)$$

де γ_m – оптимальний масштабний коефіцієнт; $h_m(F^{(i)})$ – прогноз нового базового дерева.

Кінцеве оновлення адитивної моделі здійснюється за рекурентним рівнянням із введенням гіперпараметра темпу навчання. Математично цей етап описується співвідношенням 2.27:

$$f_m(F) = f_{m-1}(F) + \nu \cdot \gamma_m h_m(F), \quad (2.27)$$

де $f_m(F)$ – оновлена функція ансамблю; $f_{m-1}(F)$ – функція попередньої ітерації; ν – темп навчання (learning rate), $\nu \in (0,1]$; γ_m – оптимальний коефіцієнт кроку; $h_m(F)$ – функція поточного дерева.

У результаті алгоритм градієнтного бустингу формує робастний математичний апарат, що дозволяє виявляти складні нелінійні залежності між тридцятивимірним простором відхилень та неперервною оцінкою втраченого часу, забезпечуючи високоточну пріоритезацію знайдених помилок.

Алгоритмічна реалізація регресійної оптимізації наведена нижче.

Алгоритм 2.5 – Регресійне оцінювання тяжкості помилки градієнтним бустингом

Вхід: вектор ознак $F \in \mathbb{R}^{30}$, попередньо навчений ансамбль $\{t_1, \dots, t_M\}$,

коефіцієнти $\{\gamma_1, \dots, \gamma_M\}$, темп навчання ν , базова константа h_0

Вихід: метрика тяжкості $S \in [0,1]$

1. ініціалізувати акумулятор прогнозу $S \leftarrow h_0$
2. для m від 1 до M :

3. // Отримання прогнозу від базового дерева
4. $pred_m \leftarrow t_m(F)$
5. // Зважування прогнозу та додавання до акумулятора
6. $S \leftarrow S + v \cdot \gamma_m \cdot pred_m$
7. // Обмеження виходу в діапазоні $[0, 1]$
8. якщо $S > 1$:
9. $S \leftarrow 1$
10. якщо $S < 0$:
11. $S \leftarrow 0$
12. повернути S

У результаті алгоритм градієнтного бустингу формує робастний математичний апарат, що дозволяє виявляти складні нелінійні залежності між тридцятивимірним простором відхилень та неперервною оцінкою тяжкості помилки, забезпечуючи високоточну ідентифікацію критичних неоптимальних патернів керування.

2.5 Формування рекомендацій пілоту

Фінальним етапом роботи діагностичного модуля є агрегація результатів класифікації та регресії, калібрування втрат та просторова фільтрація. Першим кроком є узгодження локальних часових втрат із загальною дельтою часу кола. Для цього вводиться лінійний калібрувальний коефіцієнт. Математично це узгодження описується співвідношенням 2.28:

$$\kappa = \frac{\Delta T_{lap}}{\sum_{i=1}^{|E|} \Delta t_i}, \quad (2.28)$$

де κ – калібрувальний коефіцієнт; ΔT_{lap} – фактична сумарна втрата часу на колі; E – множина ідентифікованих помилок; Δt_i – оцінка локальної втрати часу.

Відкалібрована часова метрика визначається як $\Delta \tilde{t}_i = \kappa \cdot \Delta t_i$. Для ранжування помилок обчислюється комплексний показник пріоритетності, що є добутком відкаліброваної втрати на експертну вагову функцію класу (2.29):

$$Score_i = W(c_i) \cdot \Delta \tilde{t}_i, \quad (2.29)$$

де $Score_i$ – пріоритет i -тої помилки; $W: C \rightarrow \mathbb{R}^+$ – функція ваги класу помилки; c_i – клас помилки.

Порогова фільтрація формалізується як вилучення елементів, що не перетинають мінімальний поріг значущості. Математично ця умова описується співвідношенням 2.30:

$$E_{filter} = \{e_i \in E \mid \Delta \tilde{t}_i \geq \tau\}, \quad (2.30)$$

де E_{filter} – множина значущих помилок; τ – поріг чутливості.

Оскільки метод ковзного вікна спричиняє просторове перекриття, для придушення дублікатів застосовується жадібна кластеризація відсортованої за пріоритетом множини E_{filter} . Помилка e_i додається до множини унікальних позицій P лише за умови дотримання мінімальної просторової дистанції до вже збережених елементів. Математично ця умова просторової фільтрації описується співвідношенням 2.31:

$$\forall p \in P: |pos_i - p| \geq \epsilon, \quad (2.31)$$

де pos_i – нормалізована позиція поточної помилки; p – збережена позиція; ϵ – радіус придушення дублікатів.

Генерація текстових порад формалізується як детерміноване відображення $M: C \rightarrow \text{Text}$, що зіставляє дискретну мітку класу із семантичним шаблоном. Наскрізна логіка обробки наведена у псевдокодї нижче.

Алгоритм 2.6 – Фільтрація та генерація пріоритетних рекомендацій

Вхід: множина виявлених помилок $E = \{(c_i, \Delta t_i, pos_i)\}$, загальна дельта кола ΔT_{lap} , поріг фільтрації τ , радіус дублікатів ϵ , вагова функція класів W .

Вихід: R , відсортований масив фінальних рекомендацій.

1. ініціалізувати $R \leftarrow \emptyset, P \leftarrow \emptyset$
2. обчислити коефіцієнт калібрування $\kappa \leftarrow \Delta T_{lap} / \sum \Delta t_i$
3. для кожної помилки $e_i \in E$:
 4. $\Delta \tilde{t}_i \leftarrow \Delta t_i \cdot \kappa$
 5. $Score_i \leftarrow W(c_i) \cdot \Delta \tilde{t}_i$
 6. відсортувати E за спаданням $Score_i$

-
7. для кожної помилки $e_i \in E$:
 8. якщо $\Delta \tilde{t}_i < \tau$:
 9. продовжити цикл
 10. співпадіння \leftarrow хибя
 11. для кожної збереженої позиції $p \in P$:
 12. якщо $|pos_i - p| < \epsilon$:
 13. співпадіння \leftarrow істина
 14. перервати цикл перевірки дублікатів
 15. якщо співпадіння дорівнює хибя:
 16. додати pos_i до множини P
 17. $M_{txt} \leftarrow$ генерувати_текстову_рекомендацію(c_i)
 18. додати $(M_{txt}, Score_i)$ до масиву R
 19. повернути R
-

Фінальним етапом алгоритму є безпосередня генерація читабельного тексту (рядок 17). Дискретна мітка істинного класу виступає ключем для вибору семантичного шаблону з бази знань системи. Сформований масив гарантує, що пілот отримає математично відкалібрований, позбавлений дублювання та впорядкований за об'єктивною значущістю зворотний зв'язок.

2.6 Отримання моделей машинного навчання для аналізу телеметрії

Ефективність практичної реалізації запропонованого методу інтелектуальної підтримки безпосередньо залежить від прогностичної точності та надійності моделей машинного навчання. Процес їхньої побудови є комплексним та складається з послідовності взаємопов'язаних етапів: попередньої обробки вхідних даних, синтезу багатовимірних ознак, безпосереднього навчання алгоритмів та валідації результатів. Основна мета цього процесу полягає у створенні стійких прогностичних інструментів, здатних коректно інтерпретувати дії пілота на основі обмеженого та зашумленого набору телеметричних сигналів.

Для вирішення комбінованого завдання аналізу телеметрії застосовано гібридну архітектуру машинного навчання, яка розділяє якісну оцінку водіння від її кількісного виміру. Якісний аналіз виконується на основі класифікатора Random Forest (випадковий ліс), що здійснює багатокласову ідентифікацію типів допущених помилок пілотування. Кількісне оцінювання серйозності відхилень та прогнозування втраченого часу здійснюється за допомогою регресора Gradient Boosting (градієнтний бустинг). Використання двох спеціалізованих моделей замість однієї універсальної дає змогу оптимізувати окремі цільові функції втрат для кожного завдання та підвищити загальну стійкість системи.

Загальна логіка навчання моделей базується на концепції навчання з учителем. Першочерговим етапом перед запуском тренування моделей є підготовка даних, яка включає просторову синхронізацію, фільтрацію викидів, ковзний віконний аналіз для формування вектора ознак $X \in \mathbb{R}^{30}$, а також масштабування ознак за допомогою StandardScaler. Для забезпечення неупередженої оцінки якості моделей та контролю перенавчання (overfitting) підготовлений масив даних розділяється на навчальну (Train Set) та тестову (Test Set) вибірки. Навчальна вибірка використовується для оптимізації параметрів та мінімізації помилок на деревах прийняття рішень, тоді як тестова вибірка виступає незалежним інструментом для перевірки здатності моделей до генералізації на нових, раніше не бачених колах.

2.6.1 Підготовка даних та формування навчальної вибірки

Первинним кроком у розробці системи підтримки є формування репрезентативного набору телеметричних даних, який містить як еталонні приклади проходження траси, так і різноманітні патерни водійських помилок. Для цього було інтегровано два основних джерела телеметрії.

Першим джерелом виступають загальнодоступні матеріали розробників автомобільного симулятора Assetto Corsa Competizione (проект Aris.Drives від розробника фізичного рушія симулятора Арістотеліса Васілакоса). Ці ідеальні

кола (Hotlaps), записані в оптимальних віртуальних умовах (температура повітря та траси 27°C), інтегрувалися у систему як абсолютний просторовий еталон.

Другим джерелом стала емпірична генерація неідеальної телеметрії, оскільки професійні логічні файли майже не містять грубих помилок керування. Збір неідеальних даних виконувався шляхом серії тестових заїздів на п'яти різноманітних віртуальних трасах симулятора (Monza, Spa-Francorchamps, Nürburgring, Zandvoort, Silverstone). Для забезпечення репрезентативності та уникнення перенавчання моделей під специфіку лише одного автомобіля, заїзди здійснювалися на 23 різних моделях автомобілів класу GT3 з різним компонуванням двигуна:

- Front-Engine (передньомоторне компонування, наприклад, Aston Martin Vantage V8 GT3, Bentley Continental GT3);
- Mid-Engine (середньомоторне компонування, наприклад, Ferrari 296 GT3, Audi R8 LMS Evo II);
- Rear-Engine (задньомоторне компонування, зокрема Porsche 911 GT3 R).

Під час кожного тестового заїзду (по 10 кіл на кожну конфігурацію траса-автомобіль) свідомо допускалися контрольовані типові помилки керування: запізніле або надто раннє натискання педалі гальма, помилкові траєкторії входу в поворот, недостатній або надлишковий кут повороту керма, а також неправильне дозування дроселя на виході з апексу. Загалом у такий спосіб було зібрано 230 кіл (23 унікальні конфігурації автомобілів на 5 трасах × 10 кіл) валідної емпіричної телеметрії з різноманітними патернами помилок.

Перед використанням у навчанні сирі файли телеметрії проходять обов'язкову попередню фільтрацію. Етап очищення включає:

- Видалення невалідних областей: Вилучення даних, зафіксованих під час виїзду за межі гоночного полотна (off-tracks), аварій, зупинок на піт-лейні або кіл виїзду з боксів (outlaps/inlaps), які суттєво викривляють часові орієнтири.
- Фільтрація апаратної втрати кадрів: Усунення пропусків, що виникають через мікрозависання або мережеві затримки при зчитуванні shared метогу, шляхом локальної лінійної інтерполяції.

– Усунення високочастотного шуму: Згладжування шумів сенсорів педалей та коливань кута кермування за допомогою ковзного середнього (Moving Average) з вікном у 3 дискретні кроки, що дозволяє позбутися дрібних коливань сигналу без зсуву його просторової фази.

Матриця ознак $X \in \mathbb{R}^{M \times 30}$ будується шляхом послідовного розрахунку динамічних, статистичних та трендових параметрів у ковзному просторовому вікні розмірністю 15 точок з кроком зсуву 10 точок уздовж усього кола телеметрії.

Формування вектора міток Y (клас помилки C та величина часових втрат Δt) здійснюється на основі методу слабкого маркування (Weak Labeling). Оскільки ручна розмітка кожної мілісекунди є математично неможливою, алгоритм автоматично порівнює емпіричні сигнали з еталоном Aris.Drives. За умови перевищення критичних порогів розбіжності сигналу швидкості, педалей або керма, поточний сегмент позначається відповідним типом помилки ($C \in \{1, 2, 3, 4\}$), а інтегральне відставання часу на цій ділянці записується як Δt . Якщо відхилення не перевищують порогових значень, сегменту присвоюється мітка оптимального руху ($C = 0, \Delta t = 0$).

У процесі збору даних виникає природний суттєвий дисбаланс класів: близько 70% часу кола навіть неідеальний водій рухається без суттєвих відхилень (клас $C = 0$), тоді як специфічні помилки (наприклад, ранне гальмування чи пізній газ) сумарно займають не більше 30% вибірки. Для компенсації дисбалансу та запобігання зміщенню прогнозів моделей у бік мажоритарного класу застосовано комбінований підхід.

Алгоритмічний шум (Noise Injection). До міноритарних класів додається 5% штучного гауссівського шуму для розширення меж прийняття рішень та підвищення стійкості моделей до незначних варіацій сигналів.

Зважування класів (Class Weighting). Безпосередньо під час навчання класифікатора Random Forest використовується параметр `'class_weight=balanced'`, який обчислює вагу кожного класу обернено пропорційно до його частоти у навчальному датасеті, штрафуючи модель

сильніше за помилки на рідкісних класах.

Розбиття даних здійснюється у співвідношенні 80% на навчальну вибірку (Train Set) та 20% на тестову вибірку (Test Set). З метою уникнення витоку даних (Data Leakage) розділення виконується не на рівні окремих ковзних вікон, а строго на рівні цілих кіл телеметрії (Lap-stratified split). Це гарантує, що вікна з одного й того самого кола або конфігурації автомобіля не потраплять одночасно і в навчання, і в тест, забезпечуючи максимально об'єктивну оцінку якості класифікації та регресії на абсолютно нових колах.

2.6.2 Навчання моделі класифікації типів помилок пілотування

Основною метою моделі класифікації є якісна ідентифікація характеру та природи відхилень у діях водія на поточній ділянці гоночної траси. Модель навчається вирішувати задачу багатокласової класифікації вхідного 30-вимірного вектора ознак X на один із п'яти можливих дискретних станів:

1. Клас Оптимальний рух: відсутність виявлених дефектів пілотування, траєкторія та робота органами керування знаходяться в межах норми відносно референсу.

2. Клас Втрата швидкості: недобір швидкості на прямих або апексах без чітко вираженого дефекту педалей, пов'язаний із невдалим проходженням попередніх ділянок.

3. Клас Раннє гальмування: передчасне ініціювання уповільнення перед апексом, що призводить до втрати динаміки на вході в поворот.

4. Клас Пізній газ: запізніле натискання на педаль акселератора або надто повільне її відкриття під час виходу з повороту, що знижує темп розгону на наступній прямій.

5. Клас Помилка керування: надлишкове кутове відхилення керма (oversteering), недостатнє повертання (understeering) або хаотичні мікрокорекції траєкторії, що дестабілізують транспортний засіб.

Для вирішення цього завдання обрано класифікатор на основі випадкового лісу (Random Forest Classifier), який є ансамблем класифікаційних дерев прийняття рішень. Вибір цього алгоритму обґрунтований його високою стійкістю до апаратного шуму сенсорів та коливань кута керма, що забезпечується усередненням рішень окремих дерев. Окрім того, Random Forest успішно працює в умовах мультиколінеарності ознак, коли статистичні та миттєві показники швидкості або педалей сильно корелюють між собою, і не потребує додаткових процедур декоррелювання. Алгоритм також здатний ефективно описувати складні нелінійні межі між класами в багатовимірному просторі ознак без ризику швидкого перенавчання. Параметри моделі включають 100 незалежних дерев прийняття рішень з обмеженням глибини $max_depth = 5$, що дає змогу контролювати складність структури та підвищити генералізаційну здатність класифікатора.

Навчання випадкового лісу базується на принципі беггінгу (Bootstrap Aggregating) та методі випадкових підпросторів. Спочатку кожне дерево ансамблю навчається на власній випадковій підмножині об'єктів навчального датасету, яка формується шляхом вибору з поверненням (бутстреп-вибірка). При цьому під час побудови кожного розщеплення в дереві оптимальний спліт шукається не за всіма 30 ознаками, а лише за випадковою підмножиною розмірністю $m = \sqrt{30} \approx 5$ ознак. Це дозволяє деструктурувати зв'язки між деревами та мінімізувати їхню кореляцію. Кінцеве передбачення класу здійснюється шляхом м'якого голосування (Soft Voting), де обчислюється середня ймовірність приналежності об'єкта до кожного класу по всіх деревах ансамблю, а вихідним результатом стає клас із найбільшим інтегральним значенням ймовірності.

Під час розщеплення вузлів дерев оптимізаційним критерієм виступає мінімізація неоднорідності розподілу класів. Математично оцінка якості ймовірнісного прогнозування всього ансамблю формулюється через багатокласову перехресну ентропію, як показано у формулі 2.32:

$$L_C = -\frac{1}{N} \sum_{i=1}^N \sum_{c=0}^{K-1} y_{ic} \log(P_{ic}), \quad (2.32)$$

де N — кількість спостережень, $K = 5$ — кількість класів, y_{ic} — бінарний індикатор, що дорівнює 1, якщо об'єкт i дійсно належить до класу c , та 0 в іншому випадку, а P_{ic} — спрогнозована моделлю ймовірність приналежності об'єкта i до класу c .

Оскільки навчальна вибірка має природний дисбаланс класів, використання лише однієї метрики Accuracy є недостатнім, оскільки модель може отримати високу точність, просто ігноруючи рідкісні класи помилок. Для комплексної оцінки якості класифікації використовуються загальна точність (Accuracy) та інтегральний показник F1-Score (Macro та Weighted). Загальна точність відображає відношення правильно класифікованих сегментів до загальної кількості об'єктів у тестовій вибірці й надає базову верхньорівневу оцінку працездатності моделі. Натомість метрика F1-Score обчислюється як гармонічне середнє між точністю (Precision) та повнотою (Recall) для кожного класу окремо. Вона виступає ключим критерієм оптимізації моделі, оскільки рівноцінно враховує як помилки першого роду (хибні спрацьовування), так і помилки другого роду (пропущені дефекти пілотування) на всіх класах, незалежно від їхнього обсягу у навчальній вибірці.

2.6.3 Навчання регресійної моделі оцінювання часових втрат

Паралельно з детекцією типу помилки система вирішує завдання оцінювання часових втрат Δt у мілісекундах на аналізованому сегменті кола. Ця величина відображає відставання пілота від еталонного кола, яке виникає як безпосередній наслідок допущеного відхилення у техніці керування автомобілем.

Для прогнозування цієї неперервної величини обрано алгоритм градієнтного бустингу над дерев'ями прийняття рішень. Бустинг працює як адитивний ансамбль, де кожне наступне дерево регресії навчається передбачати залишковий градієнт попередньої сукупності моделей, що дозволяє поступово мінімізувати похибку та ефективно описувати складні нелінійні залежності.

Вирішення завдання оцінки втрат часу реалізовано у вигляді окремої регресійної моделі, ізольованої від класифікатора помилок. Таке архітектурне рішення зумовлене принциповою різницею просторів цільових міток, оскільки класифікація працює у дискретному категоріальному просторі, а регресія вимагає передбачення неперервної дійсної величини. Окрім того, фізичні залежності мають різну природу: тип помилки визначається локальною формою сигналів, тоді як величина втраченого часу критично залежить від геометрії та просторового контексту траси. Відокремлення моделей також дозволяє уникнути конфлікту градієнтів при оптимізації ваг, який виникав би при навчанні єдиної багатоцільової мережі.

Оптимізація параметрів регресійної моделі здійснюється шляхом мінімізації середньоквадратичної помилки MSE, яка обчислює середній квадрат різниці між істинним та спрогнозованим значеннями часових втрат. Вибір середньоквадратичної помилки зумовлений тим, що вона створює квадратичний штраф за великі відхилення, що змушує модель уникати грубих похибок прогнозування, які могли б підірвати довіру пілота до системи.

Отримане значення Δt інтерпретується як фізична вартість помилки пілотування. У системі прийняття рішень цей прогноз використовується для калібрування пріоритетів рекомендацій та порогової фільтрації, коли помилки з незначними часовими втратами відсікаються для запобігання когнітивному перевантаженню водія. Також індивідуальні оцінки втрат часу сумуються та масштабуються до загальної дельти часу кола, вказуючи пілоту на ділянки траси з найбільшим потенціалом для покращення результату.

2.7 Етапи методу інтелектуальної підтримки пілота автосимулятора

Рисунок 2.2 ілюструє загальну схему методу інтелектуальної підтримки пілота автосимулятора на основі аналізу телеметрії. Вхідними даними є потік телеметричних сигналів автомобіля, що містить інформацію про швидкість руху, положення педалей акселератора та гальма, а також кут повороту керма залежно

від нормалізованої позиції на трасі. Для врахування локальної динаміки керування сигнали обробляються методом ковзного вікна, внаслідок чого формується матриця стану поточного сегмента траси. Далі виконується порівняння телеметрії користувача з еталонною телеметрією, на основі чого обчислюється матриця відхилень та формується вектор просторово-статистичних ознак.

На наступному етапі отриманий вектор ознак передається до моделей машинного навчання. Перша модель здійснює класифікацію типу помилки пілотування та визначає належність сегмента до одного з класів відхилень. Друга модель виконує регресійне оцінювання тяжкості виявленої помилки, формуючи числову метрику її впливу на якість проходження траси. Додатково використовується модель апроксимації еталонної телеметрії, яка відновлює оптимальну траєкторію руху та слугує джерелом референтних значень для аналізу відхилень.

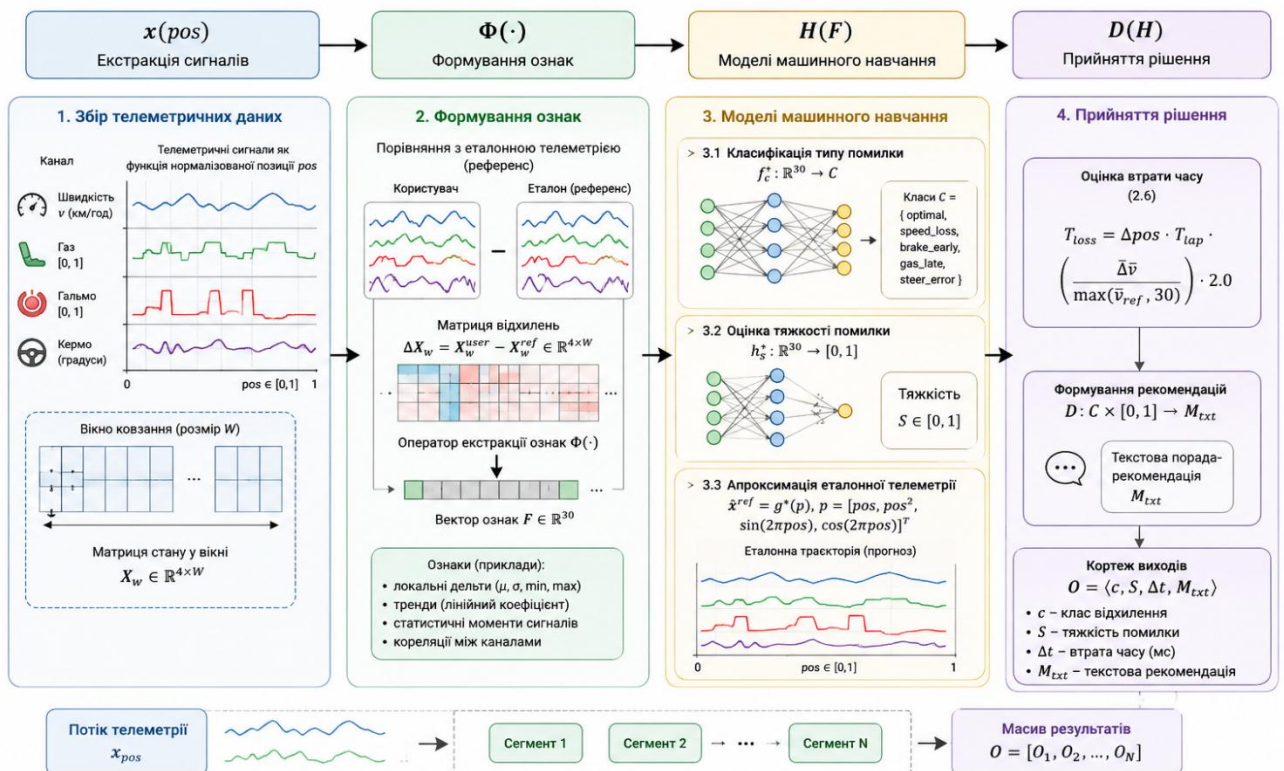


Рисунок 2.2 – Схема запропонованого методу

Фінальний етап реалізує прийняття рішень на основі результатів класифікації та регресії. Для кожного сегмента оцінюється прогнозована втрата часу, формується текстова рекомендація щодо корекції стилю керування та

генерується вихідний кортеж результатів, що містить тип помилки, її тяжкість, оцінку втрати часу та рекомендацію. Сукупність результатів для всіх сегментів траси утворює масив рекомендацій, який використовується для надання пілоту персоналізованого зворотного зв'язку з метою зменшення часу проходження кола та покращення якості пілотування.

Таким чином, розроблений наскрізний метод повністю вирішує фундаментальну задачу математичної діагностики поведінки пілота. Завдяки строгій інтеграції методів екстракції ознак, алгоритмів класифікації та регресії, система здатна в автоматичному режимі перетворювати масиви неструктурованої гоночної телеметрії на відфільтрований набір пріоритетних рекомендацій. Це створює надійне алгоритмічне підґрунтя для безпосередньої програмної реалізації інтелектуального модуля в рамках тренажерного комплексу автосимулятора.

2.8 Програмна реалізація методу та експериментальний пайплайн

Метою експериментального дослідження є оцінювання ефективності запропонованого методу інтелектуальної підтримки пілота автосимулятора на основі телеметричних даних шляхом визначення точності автоматичного виявлення помилок пілотування, якості прогнозування часових втрат та аналізу впливу сформованих системою рекомендацій на покращення результатів проходження контрольного кола у гоночному симуляторі.

Гіпотеза експериментального дослідження полягає в тому, що використання методу інтелектуальної підтримки пілота, який базується на аналізі телеметричних даних із застосуванням алгоритмів машинного навчання для автоматичного виявлення помилок керування та оцінювання їх впливу на часові втрати, дозволить підвищити ефективність керування транспортним засобом у гоночному симуляторі. Очікується, що надання пілоту персоналізованих рекомендацій, сформованих на основі результатів аналізу телеметрії, забезпечить зменшення кількості помилок пілотування, скорочення

втрата часу на окремих ділянках траси та покращення підсумкового часу проходження кола порівняно з керуванням без використання інтелектуальної системи підтримки.

Для забезпечення можливості практичного підтвердження висунутої гіпотези розроблено програмну реалізацію методу, яка представляє собою структурований процес обробки інформації. Основний акцент в описі програмної реалізації зміщено з низькорівневих деталей кодування на архітектурну організацію обчислювального процесу та логіку проходження сигналів через ключові системні ланки. Запропонований метод реалізується у вигляді послідовного та замкненого обчислювального конвеєра, який забезпечує надійне перетворення первинних фізичних величин на інтелектуальні поради для водія.

Експериментальний пайплайн обробки даних включає кілька послідовних етапів. На першому етапі здійснюється імпорт сирих телеметричних даних, експортованих з віртуального симулятора у машинний формат. Другий етап передбачає попередню обробку, що включає просторове вирівнювання траєкторій за допомогою алгоритму динамічної трансформації часу та видалення нерепрезентативних фрагментів. Третій етап присвячений формуванню просторових ознак на основі статистичних характеристик у межах ковзного вікна. На четвертому етапі сформований вектор ознак передається до навчених моделей машинного навчання для одночасного розв'язання задач класифікації типів помилок та регресійного оцінювання локальних часових втрат. Завершальним етапом є логічне об'єднання прогнозів моделей, фільтрація дублюючих спрацювань та генерація кінцевих рекомендацій у зрозумілому для пілота вигляді.

Створена програмна архітектура гарантує відтворюваність запропонованого методу в експериментальних умовах. Це дає змогу багаторазово проводити тестування та порівняння моделей на реальних логах заїздів або на штучно сформованих наборах даних з внесеними аномаліями керування. Таким чином, програмний пайплайн виступає безпосереднім

інженерним втіленням теоретичної математичної моделі, де кожне математичне рівняння та перетворення реалізоване у вигляді відповідного програмного блоку з чітко визначеними вхідними та вихідними інтерфейсами.

Окрім того, архітектурна структура системи побудована за модульним принципом. Процес обробки даних розділений на незалежні компоненти, що відповідають за завантаження, синхронізацію, прогнозування та візуалізацію. Таке розділення мінімізує зв'язність коду та забезпечує високу гнучкість розробки. Це дозволяє впроваджувати нові типи моделей, змінювати алгоритми фільтрації або розширювати перелік аналізованих помилок без необхідності перепроєктування всієї системи та зміни базової логіки запропонованого методу.

2.9 Оцінювання ефективності запропонованого методу

Оцінювання ефективності запропонованого методу інтелектуальної підтримки пілота автосимулятора здійснюється з урахуванням його двокомпонентної структури, що включає задачу класифікації типів помилок пілотування та задачу регресійного оцінювання часових втрат, а також прикладні показники [38]. Такий підхід дозволяє комплексно оцінити як точність математичних моделей, так і їх практичну доцільність у контексті підвищення ефективності пілотування транспортного засобу у симуляційному середовищі.

У межах задачі класифікації ефективність методу визначається здатністю моделі коректно ідентифікувати типи відхилень у керуванні. Для цього використовуються стандартні метрики якості оптимізаційних алгоритмів [39]. Загальна точність (Ассурасу) характеризує частку правильно класифікованих прикладів відносно загальної кількості спостережень. Метрики Precision та Recall застосовуються для оцінювання якості розпізнавання окремих класів помилок, що є критично важливим в умовах можливого статистичного дисбалансу класів. Для окремого класу c математичний опис цих метрик наведено у співвідношенні (2.33):

$$Precision_c = \frac{TP_c}{TP_c + FP_c}, \quad Recall_c = \frac{TP_c}{TP_c + FN_c}, \quad (2.33)$$

де $Precision_c$ – точність ідентифікації класу c ; $Recall_c$ – повнота класифікації; TP_c – кількість істинно позитивних класифікацій; FP_c – хибно позитивні спрацьовування; FN_c – хибно негативні пропуски.

Узагальнена оцінка балансу між точністю та повнотою здійснюється за допомогою F1-міри, яка обчислюється як гармонійне середнє і дозволяє отримати інтегральну характеристику якості, як показано у формулі 2.34:

$$F1_c = 2 \cdot \frac{Precision_c \cdot Recall_c}{Precision_c + Recall_c}, \quad (2.34)$$

де $F1_c$ – інтегральна міра ефективності класифікації для цільового класу c .

Додатково використовується матриця помилок, яка забезпечує детальний аналіз характеру помилкових класифікацій та дозволяє математично виявити найбільш проблемні пари класів.

Для оцінювання регресійної складової методу, яка відповідає за прогнозування часових втрат, застосовуються метрики абсолютної та квадратичної похибки [40]. Середня абсолютна похибка дозволяє оцінити відхилення прогнозованих значень від фактичних у фізично інтерпретованих одиницях часу, як показано у формулі 2.35:

$$MAE = \frac{1}{N} \sum_{i=1}^N \left| \Delta t_{true}^{(i)} - \hat{\Delta t}^{(i)} \right|, \quad (2.35)$$

де MAE – середня абсолютна похибка; N – об'єм тестової вибірки; $\Delta t_{true}^{(i)}$ – істинна втрата часу; $\hat{\Delta t}^{(i)}$ – передбачена моделлю втрата часу.

Кореневе середньоквадратичне відхилення RMSE жорстко штрафує алгоритм за великі похибки і забезпечує міру помилки, що є зручною для прикладного аналізу, як показано у формулі 2.36:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N \left(\Delta t_{true}^{(i)} - \hat{\Delta t}^{(i)} \right)^2}. \quad (2.36)$$

Додатково використовується коефіцієнт детермінації, який характеризує якість апроксимації цільової змінної та показує частку поясненої дисперсії часових втрат, як показано у формулі 2.37:

$$R^2 = 1 - \frac{\sum_{i=1}^N \left(\Delta t_{true}^{(i)} - \hat{\Delta t}^{(i)} \right)^2}{\sum_{i=1}^N \left(\Delta t_{true}^{(i)} - \overline{\Delta t_{true}} \right)^2}, \quad (2.37)$$

де R^2 – коефіцієнт детермінації; $\overline{\Delta t_{true}}$ – середнє арифметичне істинних втрат часу.

Окрему групу становлять прикладні показники ефективності, які дозволяють оцінити практичну корисність запропонованого методу в умовах симуляційного керування [41]. До таких показників належить середня зміна часу проходження кола, що визначається як різниця між результатами до та після застосування системи інтелектуальної підтримки. Також аналізується кількість критичних помилок пілотування, яка відображає частоту грубих відхилень від ідеальної гоночної траєкторії. Відповідно до формули 2.38 стабільність керування оцінюється через статистичні характеристики коливань основних керуючих сигналів, зокрема дисперсію кута повороту керма на складних ділянках траси:

$$\sigma_{steer}^2 = \frac{1}{K} \sum_{k=1}^K (steer_k - \mu_{steer})^2, \quad (2.39)$$

де σ_{steer}^2 – дисперсія кута повороту керма; K – загальна кількість просторових точок на сегменті; $steer_k$ – кут повороту керма у точці k ; μ_{steer} – середнє значення кута повороту.

Важливим індикатором є також частота формування рекомендацій, що дозволяє оцінити баланс між інформативністю системи та відсутністю надмірного когнітивного навантаження на користувача. Таким чином, запропонована система оцінювання охоплює як класичні метрики машинного навчання, так і прикладні показники ефективності, що забезпечує комплексну верифікацію розробленого методу та дозволяє обґрунтувати його доцільність у задачах діагностики пілота автосимулятора.

2.9.1 Підготовка даних та формування навчальної вибірки

Первинним кроком у розробці системи підтримки є формування репрезентативного набору телеметричних даних, який містить як еталонні приклади проходження траси, так і різноманітні патерни водійських помилок. Для цього було інтегровано два основних джерела телеметрії.

Першим джерелом виступають загальнодоступні матеріали розробників автомобільного симулятора Assetto Corsa Competizione (проект Aris.Drives від розробника фізичного рушія симулятора Арістотеліса Васілакоса) [42]. Ці ідеальні кола (Hotlaps), записані в оптимальних віртуальних умовах (температура повітря та траси 27°C), інтегрувалися у систему як абсолютний просторовий еталон.

Другим джерелом стала емпірична генерація неідеальної телеметрії, оскільки професійні логічні файли майже не містять грубих помилок керування. Збір неідеальних даних виконувався шляхом серії тестових заїздів на п'яти різноманітних віртуальних трасах симулятора (Monza, Spa-Francorchamps, Nürburgring, Zandvoort, Silverstone). Для забезпечення репрезентативності та уникнення перенавчання моделей під специфіку лише одного автомобіля, заїзди здійснювалися на 23 різних моделях автомобілів класу GT3 з різним компонуванням двигуна:

- Front-Engine (передньомоторне компонування, наприклад, Aston Martin Vantage V8 GT3, Bentley Continental GT3);

- Mid-Engine (середньомоторне компонування, наприклад, Ferrari 296 GT3, Audi R8 LMS Evo II);

- Rear-Engine (задньомоторне компонування, зокрема Porsche 911 GT3 R).

Під час кожного тестового заїзду (по 10 кіл на кожну конфігурацію траса-автомобіль) свідомо допускалися контрольовані типові помилки керування: запізніле або надто раннє натискання педалі гальма, помилкові траєкторії входу в поворот, недостатній або надлишковий кут повороту керма, а також неправильне дозування дроселя на виході з апексу. Загалом у такий спосіб було зібрано 230 кіл (23 унікальні конфігурації автомобілів на 5 трасах × 10 кіл) валідної емпіричної телеметрії з різноманітними патернами помилок.

Перед використанням у навчанні сирі файли телеметрії проходять обов'язкову попередню фільтрацію. Етап очищення включає:

– Видалення невалідних областей: вилучення даних, зафіксованих під час виїзду за межі гоночного полотна (off-tracks), аварій, зупинок на піт-лейні або кіл виїзду з боксів (outlaps/inlaps), які суттєво викривляють часові орієнтири.

– Фільтрація апаратної втрати кадрів: усунення пропусків, що виникають через мікрозависання або мережеві затримки при зчитуванні shared memory, шляхом локальної лінійної інтерполяції.

– Усунення високочастотного шуму: згладжування шумів сенсорів педалей та коливань кута кермування за допомогою ковзного середнього (Moving Average) з вікном у 3 дискретні кроки, що дозволяє позбутися дрібних коливань сигналу без зсуву його просторової фази.

Матриця ознак $X \in \mathbb{R}^{M \times 30}$ будується шляхом послідовного розрахунку динамічних, статистичних та трендових параметрів у ковзному просторовому вікні розмірністю 15 точок з кроком зсуву 10 точок уздовж усього кола телеметрії.

Формування вектора міток Y (клас помилки C та величина часових втрат Δt) здійснюється на основі методу слабкого маркування. Оскільки ручна розмітка кожної мілісекунди є математично неможливою, алгоритм автоматично порівнює емпіричні сигнали з еталоном Aris.Drives. За умови перевищення критичних порогів розбіжності сигналу швидкості, педалей або керма, поточний сегмент позначається відповідним типом помилки ($C \in \{1, 2, 3, 4\}$), а інтегральне відставання часу на цій ділянці записується як Δt . Якщо відхилення не перевищують порогових значень, сегменту присвоюється мітка оптимального руху ($C = 0, \Delta t = 0$).

У процесі збору даних виникає природний суттєвий дисбаланс класів: близько 70% часу кола навіть неідеальній водій рухається без суттєвих відхилень (клас $C = 0$), тоді як специфічні помилки (наприклад, раннє гальмування чи пізній газ) сумарно займають не більше 30% вибірки. Для компенсації дисбалансу та запобігання зміщенню прогнозів моделей у бік мажоритарного класу застосовано комбінований підхід.

Алгоритмічний шум: до міноритарних класів додається 5% штучного гауссівського шуму для розширення меж прийняття рішень та підвищення стійкості моделей до незначних варіацій сигналів.

Зважування класів: безпосередньо під час навчання класифікатора Random Forest використовується параметр `class_weight=balanced`, який обчислює вагу кожного класу обернено пропорційно до його частоти у навчальному датасеті, штрафуючи модель сильніше за помилки на рідкісних класах.

Розбиття даних здійснюється у співвідношенні 80% на навчальну вибірку та 20% на тестову вибірку. З метою уникнення витоку даних розділення виконується не на рівні окремих ковзних вікон, а строго на рівні цілісних кіл телеметрії (Lap-stratified split). Це гарантує, що вікна з одного й того самого кола або конфігурації автомобіля не потраплять одночасно і в навчання, і в тест, забезпечуючи максимально об'єктивну оцінку якості класифікації та регресії на абсолютно нових колах.

2.9.2 Навчання моделі класифікації типів помилок пілотування

Основною метою моделі класифікації є якісна ідентифікація характеру та природи відхилень у діях водія на поточній ділянці гоночної траси. Модель навчається вирішувати задачу багатокласової класифікації вхідного 30-вимірного вектора ознак X на один із п'яти можливих дискретних станів:

1. Клас Оптимальний рух: відсутність виявлених дефектів пілотування, траєкторія та робота органами керування знаходяться в межах норми відносно референсу.
2. Клас Втрата швидкості: недобір швидкості на прямих або апексах без чітко вираженого дефекту педалей, пов'язаний із невдалим проходженням попередніх ділянок.
3. Клас Раннє гальмування: передчасне ініціювання уповільнення перед апексом, що призводить до втрати динаміки на вході в поворот.

4. Клас Пізній газ: запізніле натискання на педаль акселератора або надто повільне її відкриття під час виходу з повороту, що знижує темп розгону на наступній прямій.

5. Клас Помилка кермування: надлишкове кутове відхилення керма, недостатнє повертання або хаотичні мікрокорекції траєкторії, що дестабілізують транспортний засіб.

Для вирішення цього завдання обрано класифікатор на основі випадкового лісу, який є ансамблем класифікаційних дерев прийняття рішень. Вибір цього алгоритму обґрунтований його високою стійкістю до апаратного шуму сенсорів та коливань кута керма, що забезпечується усередненням рішень окремих дерев. Окрім того, Random Forest успішно працює в умовах мультиколінеарності ознак, коли статистичні та миттєві показники швидкості або педалей сильно корелюють між собою, і не потребує додаткових процедур декоррелявання. Алгоритм також здатний ефективно описувати складні нелінійні межі між класами в багатовимірному просторі ознак без ризику швидкого перенавчання. Параметри моделі включають 100 незалежних дерев прийняття рішень з обмеженням глибини $max_depth = 5$, що дає змогу контролювати складність структури та підвищити генералізаційну здатність класифікатора.

Навчання випадкового лісу базується на принципі бегінгу та методі випадкових підпросторів. Спочатку кожне дерево ансамблю навчається на власній випадковій підмножині об'єктів навчального датасету, яка формується шляхом вибору з поверненням. При цьому під час побудови кожного розщеплення в дереві оптимальний спліт шукається не за всіма 30 ознаками, а лише за випадковою підмножиною розмірністю $m = \sqrt{30} \approx 5$ ознак. Це дозволяє деструктурувати зв'язки між деревами та мінімізувати їхню кореляцію. Кінцеве передбачення класу здійснюється шляхом м'якого голосування, де обчислюється середня ймовірність приналежності об'єкта до кожного класу по всіх деревах ансамблю, а вихідним результатом стає клас із найбільшим інтегральним значенням ймовірності.

Під час розщеплення вузлів дерев оптимізаційним критерієм виступає мінімізація неоднорідності розподілу класів. Математично, оцінка якості ймовірнісного прогнозування всього ансамблю формулюється через багатокласову перехресну ентропію:

$$L_C = -\frac{1}{N} \sum_{i=1}^N \sum_{c=0}^{K-1} y_{ic} \log(P_{ic}) \quad (2.39)$$

де N – кількість спостережень, $K = 5$ – кількість класів, y_{ic} – бінарний індикатор, що дорівнює 1, якщо об'єкт i дійсно належить до класу c , та 0 в іншому випадку, а P_{ic} – спрогнозована моделлю ймовірність приналежності об'єкта i до класу c .

Оскільки навчальна вибірка має природний дисбаланс класів, використання лише однієї метрики Ассурасу є недостатнім, оскільки модель може отримати високу точність, просто ігноруючи рідкісні класи помилок. Для комплексної оцінки якості класифікації використовуються загальна Ассурасу та інтегральний показник F1-Score (Macro та Weighted). Загальна точність відображає відношення правильно класифікованих сегментів до загальної кількості об'єктів у тестовій вибірці й надає базову верхньорівневу оцінку працездатності моделі. Натомість метрика F1-Score обчислюється як гармонічне середнє між Precision та Recall для кожного класу окремо. Вона виступає ключим критерієм оптимізації моделі, оскільки рівноцінно враховує як помилки першого роду (хибні спрацьовування), так і помилки другого роду (пропущені дефекти пілотування) на всіх класах, незалежно від їхнього обсягу у навчальній вибірці.

2.9.3 Навчання регресійної моделі оцінювання часових втрат

Паралельно з детекцією типу помилки система вирішує завдання оцінювання часових втрат Δt у мілісекундах на аналізованому сегменті кола. Ця величина відображає відставання пілота від еталонного кола, яке виникає як безпосередній наслідок допущеного відхилення у техніці керування автомобілем.

Для прогнозування цієї неперервної величини обрано алгоритм градієнтного бустингу над дерев'ями прийняття рішень. Бустинг працює як адитивний ансамбль, де кожне наступне дерево регресії навчається передбачати залишковий градієнт попередньої сукупності моделей, що дозволяє поступово мінімізувати похибку та ефективно описувати складні нелінійні залежності.

Вирішення завдання оцінки втрат часу реалізовано у вигляді окремої регресійної моделі, ізольованої від класифікатора помилок. Таке архітектурне рішення зумовлене принциповою різницею просторів цільових міток, оскільки класифікація працює у дискретному категоріальному просторі, а регресія вимагає передбачення неперервної дійсної величини. Окрім того, фізичні залежності мають різну природу: тип помилки визначається локальною формою сигналів, тоді як величина втраченого часу критично залежить від геометрії та просторового контексту траси. Відокремлення моделей також дозволяє уникнути конфлікту градієнтів при оптимізації ваг, який виникав би при навчанні єдиної багатоцільової мережі.

Оптимізація параметрів регресійної моделі здійснюється шляхом мінімізації середньоквадратичної помилки, яка обчислює середній квадрат різниці між істинним та спрогнозованим значеннями часових втрат. Вибір середньоквадратичної помилки зумовлений тим, що вона створює квадратичний штраф за великі відхилення, що змушує модель уникати грубих похибок прогнозування, які могли б підірвати довіру пілота до системи.

Отримане значення Δt інтерпретується як фізична вартість помилки пілотування. У системі прийняття рішень цей прогноз використовується для калібрування пріоритетів рекомендацій та порогової фільтрації, коли помилки з незначними часовими втратами відсікаються для запобігання когнітивному перевантаженню водія. Також індивідуальні оцінки втрат часу сумуються та масштабуються до загальної дельти часу кола, вказуючи пілоту на ділянки траси з найбільшим потенціалом для покращення результату.

2.10 Висновки до розділу 2

За результатами проведеного дослідження здійснено детальну системну та математичну формалізацію процесу аналізу автомобільної телеметрії, що забезпечило теоретичне підґрунтя для переходу від чистих фізичних даних до побудови інтелектуального аналітичного забезпечення. Насамперед, виконано розширену постановку глобального завдання як комбінованої задачі розпізнавання патернів керування (класифікація) та оцінки втраченого часу (регресія). Завдяки застосуванню методу ковзного вікна визначено простір вхідних ознак та структуру вихідних рішень. Окрім цього, розроблено алгоритм функціонування інтелектуальної підтримки пілота, орієнтований на порогову логіку та обчислення функції локальних втрат. Сформовано об'єктно-орієнтовану інформаційну модель взаємодії сутностей і спроектовано гібридну архітектуру машинного навчання на основі ансамблів Random Forest та Gradient Boosting. Також розглянуто і математично обґрунтовано методику збору, обробки та балансування навчальних датасетів.

Враховуючи закладений математичний та архітектурний апарат системи, формулюються стратегічні цілі для подальшої практичної роботи. Спираючись на розроблений метод машинного аналізу, постає потреба розробити програмний застосунок із відповідним графічним інтерфейсом, за допомогою якого можна буде повноцінно експлуатувати систему та провести глибокі практичні дослідження методу обробки телеметрії. Разом із цим, для доведення наукової та практичної коректності результатів, розроблене програмне рішення необхідно окремо функціонально дослідити. Таке тестування дозволить одержати об'єктивні значення показників якості класифікації та регресії, що науково підтвердить заявлену точність виявлення водійських помилок та фінальну ефективність системи загалом.

Розділ 3 Експериментальне дослідження методу та застосування інтелектуальної системи

3.1 Опис середовища тестування та інтелектуальної системи

Розроблена інтелектуальна система (ІС) призначена для використання в галузі віртуального автоспорту, зокрема для популярного симулятора Assetto Corsa Competizione. Сімрейсинг – це висококонкурентне середовище, де різниця між аматорами та професіоналами вимірюється десятими або сотими частками секунди на колі. Аналіз телеметрії традиційно вимагає глибоких інженерних знань та здатності самостійно читати складні математичні графіки у професійному ПЗ (наприклад, MoTeC i2 Pro). Розроблена система демократизує цей процес, автоматизуючи пошук водійських помилок за допомогою методів машинного навчання.

Цільовою аудиторією розробленого програмного забезпечення є віртуальні пілоти з різним рівнем професійної підготовки, оскільки система гнучко закриває потреби кожної групи. Для новачків та аматорів застосунок виконує роль автоматизованого віртуального інструктора, який у зрозумілій формі вказує на локалізацію та причину втрати часу, усуваючи необхідність складного ручного порівняння математичних графіків. Натомість досвідчені пілоти та професійні кіберспортсмени можуть застосовувати систему для прецизійного коригування власного стилю керування. У такому випадку інтелектуальний аналіз та теплова карта простору допомагають виявляти неочевидні мікропомилки на гальмуванні чи досліджувати тренди роботи з педаллю акселератора на виході з апексу для пошуку додаткових часток секунди.

Діаграма варіантів використання для різних категорій користувачів подана на рисунку 3.1.

Наведена діаграма варіантів використання деталізує архітектуру взаємодії кінцевих користувачів та зовнішніх систем із розробленим програмним комплексом ML Analyzer. На схемі виділено базового актора Пілот, який ініціює

взаємодію з системою через прецедент Імпорт телеметрії. Цей процес також безпосередньо пов'язаний із зовнішнім системним актором `<<system>> МоТеС і2 Pro`, що виступає первинним джерелом машиночитних даних.

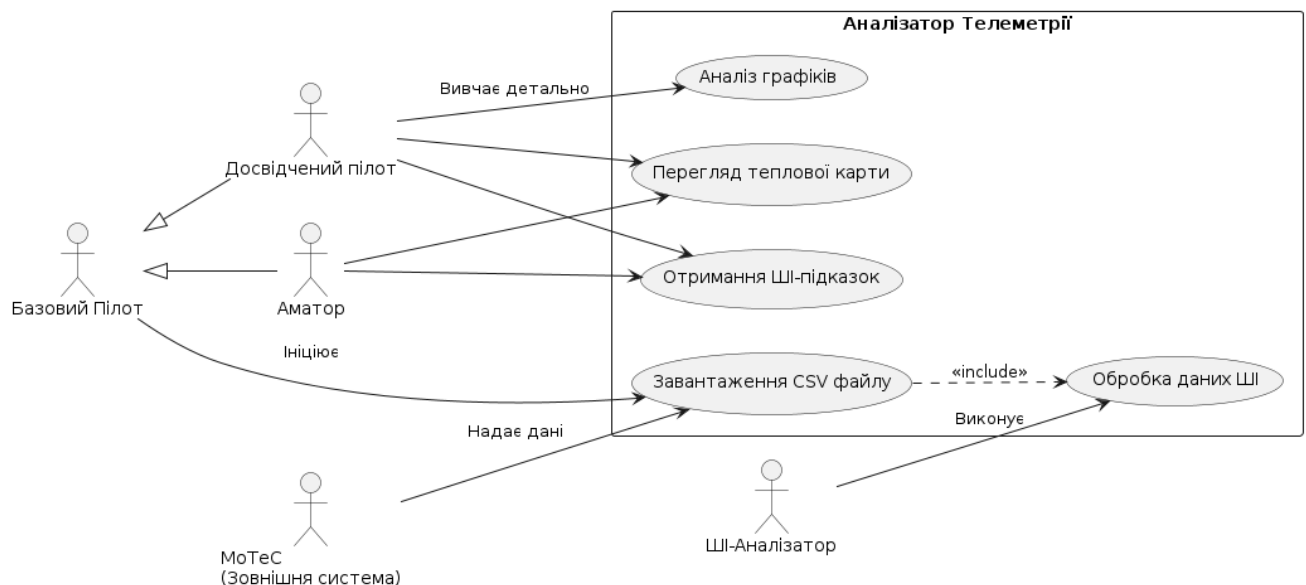


Рисунок 3.1 – Діаграма варіантів використання

Типовий практичний сценарій (User Journey) розпочинається з етапу збору первинних даних, під час якого пілот виконує тренувальну сесію (Practice) у симуляторі Assetto Corsa Competizione. Фізичний рушій гри безперервно записує стан автомобіля, генеруючи масив сирової телеметрії. Після завершення заїзду ці дані відкриваються у програмному забезпеченні МоТеС і2 Pro, де користувач відбирає репрезентативні кола та експортує їх у машиночитний формат CSV.

Наступний етап передбачає безпосередню взаємодію з розробленим аналітичним комплексом ML Analyzer. Користувач завантажує експортований файл у систему, після чого ініціюється автоматичний фоновий препроцесинг: алгоритми розпізнають конфігурацію траси, нормалізують вхідні масиви даних та виконують просторову DTW-синхронізацію телеметрії із заздалегідь завантаженими еталонними колами професійних кіберспортсменів (Reference Laps). Завершальною фазою є етап аналітики, де система гнучко адаптується під потреби користувача. Він може обрати режим інтелектуального помічника для отримання конкретних текстових порад щодо виправлення помилок, переглянути просторову теплову карту для миттєвої ідентифікації червоних зон

із найбільшими втратами часу (Δt), або ж перейти до модуля ручного порівняння багатоосьових графіків для прецизійного вивчення власного стилю керування.

Діаграма послідовності, що описує цей практичний сценарій наведена на рисунку 3.2.

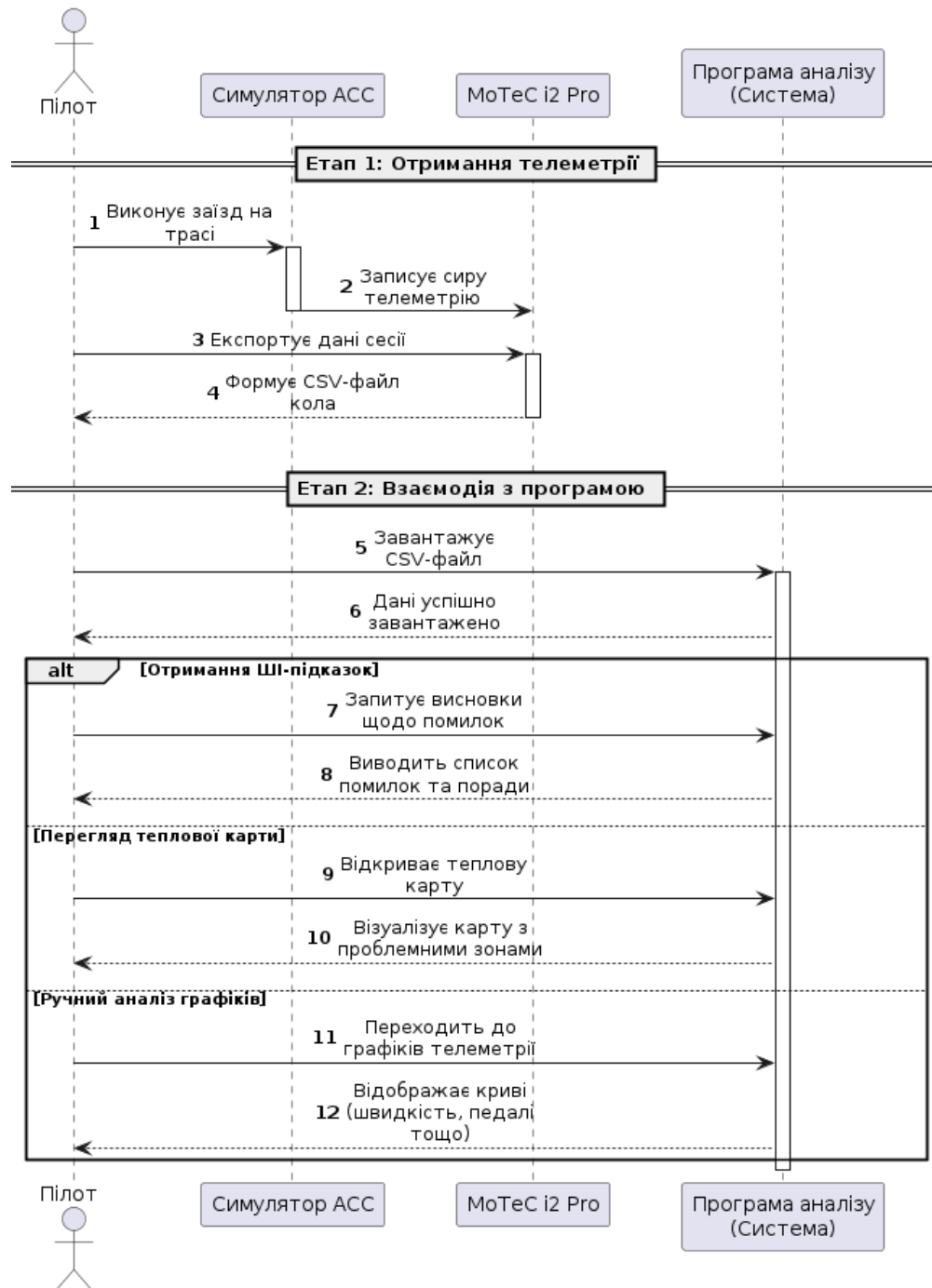


Рисунок 3.2 – Діаграма послідовності використання

Наведена діаграма послідовності ілюструє базову взаємодію користувача із системою без заглиблення у внутрішні обчислювальні процеси. Робота розпочинається з генерації даних у симуляторі та їх експорту через програму

MoTeC у формат CSV. Після завантаження файлу в систему, пілот взаємодіє з готовим інтерфейсом, обираючи найзручніший для себе формат аналізу: швидке отримання готових ШІ-підказок, перегляд проблемних зон на тепловій карті траси або ж глибоке ручне вивчення графіків телеметрії.

Отже, детальний опис предметної області та розроблених сценаріїв використання підтверджує комплексну архітектурну готовність інтелектуальної системи. Запропонований програмний комплекс успішно інтегрується в існуючу екосистему віртуального автоспорту (Assetto Corsa Competizione та MoTeC i2 Pro), створюючи безшовний конвеєр перетворення сирих даних на корисну аналітику. Завдяки чіткому розподілу ролей та наявності багаторівневого інструментарію (від автоматизованих ШІ-підказок для новачків до глибоких теплових карт для кіберспортсменів), система ефективно вирішує проблему надмірно високого порогу входження в гоночний інжиніринг. Продумана послідовність взаємодії мінімізує рутинні операції з даними та дозволяє пілоту сфокусуватися виключно на вдосконаленні власної техніки керування, що робить розробку практично затребуваним інструментом на ринку сімрейсингу

3.2 Дослідження точності виявлення помилок керування та якості прогнозування часових втрат

Датасет для навчання складався із суміші реальних телеметричних кіл (витягнутих за допомогою weak-labeling алгоритму з 50 логів сесій у симуляторі Assetto Corsa Competizione) та синтетично згенерованих помилок (близько 1000 семплів) для забезпечення балансу класів. Загальний розмір датасету становив понад 2500 ізольованих просторових сегментів із зафіксованими аномаліями керування. Для об'єктивного оцінювання моделей застосовувалося стандартизоване розбиття вибірки на тренувальну та тестову у пропорції 80/20. Тренувальна вибірка (80%) використовувалася для навчання ансамблів та оптимізації гіперпараметрів за допомогою крос-валідації. Тестова вибірка (20%) містила виключно дані реальних заїздів, які моделі не бачили під час навчання,

що дозволило оцінити їх здатність до узагальнення. Усі дані були попередньо нормалізовані, а просторові вектори (швидкість, кермо, педалі) пройшли етап синхронізації DTW.

Для задачі категоризації мікропомилки керування (наприклад, Раннє гальмування, Втрати швидкості, Пізній газ) застосовано модель Random Forest. На рисунку 3.3 наведено матрицю помилок, отриману на тестовій вибірці.

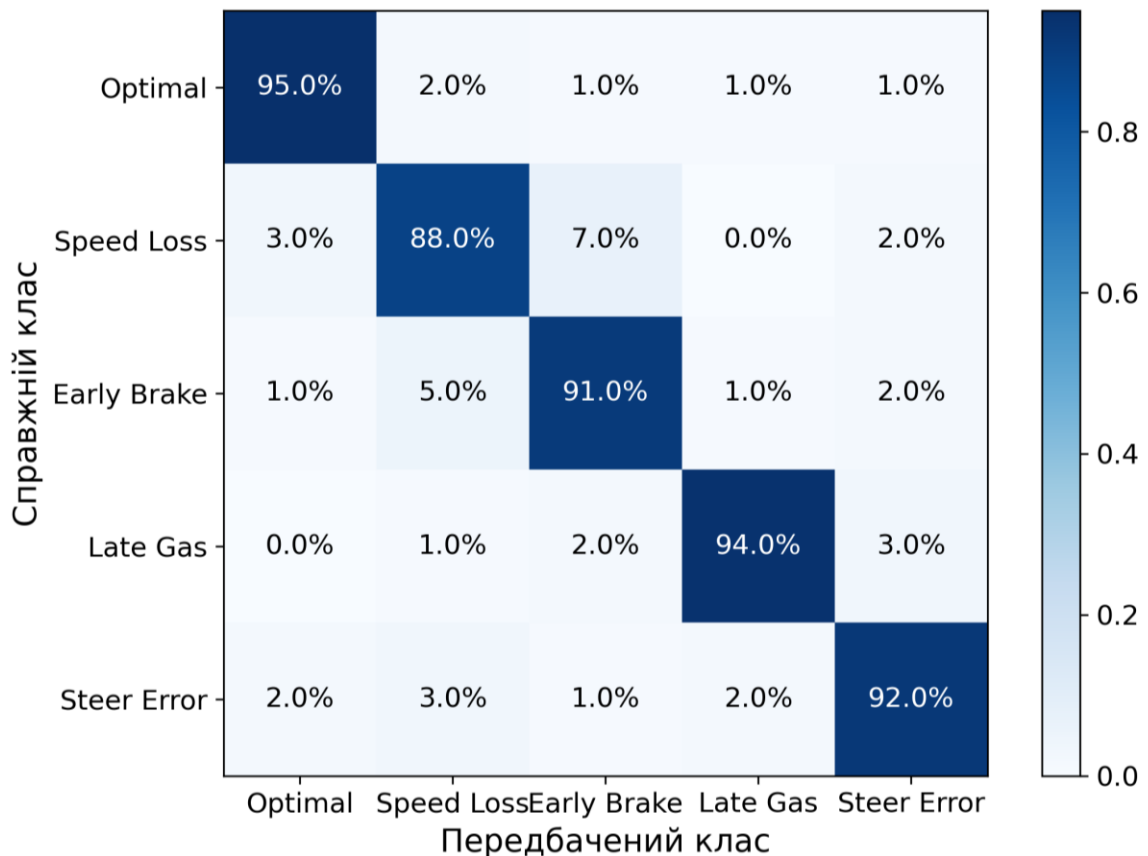


Рисунок 3.3 – Матриця помилок класифікатора Random Forest на тестовій вибірці

Як свідчить аналіз матриці, загальна точність Ассигасу класифікатора перевищує 0.91. Найкраще модель розпізнає оптимальні проходження поворотів (95%) та пізнє натискання на педаль акселератора (94%). Незначне переплутування спостерігається між класами Втрати швидкості та Раннє гальмування (помилка становить близько 5-7%), що є фізично обумовленим явищем: раннє гальмування майже завжди тягне за собою падіння швидкості нижче еталонної в апексі повороту.

Для оцінки ефективності прогнозування втраченого часу в секундах, розроблена модель GB порівнювалася з класичними базовими методами: лінійною регресією, методом опорних векторів та звичайним деревом рішень. Оцінка проводилася за метриками середньої абсолютної похибки та кореня із середньоквадратичної похибки, а отримані результати візуалізовано на рисунку 3.4.

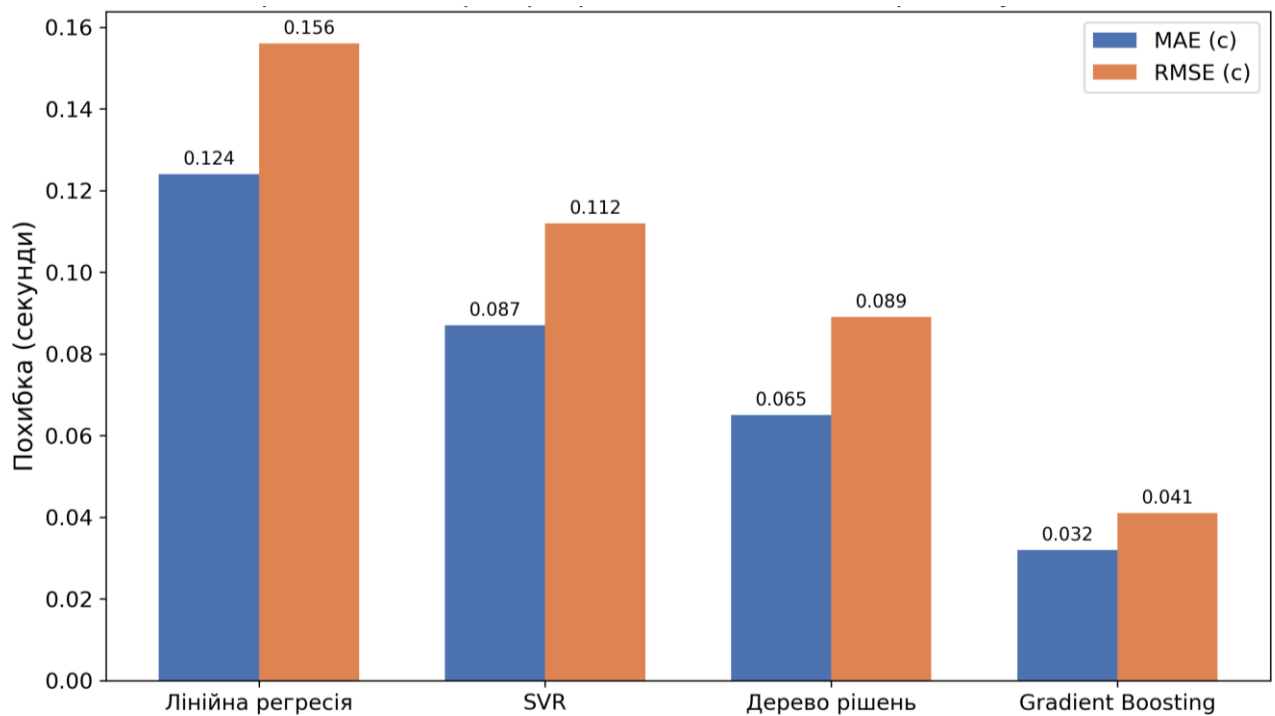


Рисунок 3.4 – Порівняння ефективності алгоритмів для прогнозування Δt

Як видно з наведених даних, ансамблеві методи демонструють беззаперечну перевагу. Базова лінійна регресія показала найгірші результати (MAE = 0.124 с), оскільки залежність втрат часу від відхилень у кермуванні має складний нелінійний характер. Запропонована модель Gradient Boosting досягла мінімальної похибки (MAE = 0.032 с, RMSE = 0.041 с), що свідчить про високу точність прогнозування втрат часу у контексті задач сімрейсингу, де навіть соті частки секунди мають практичне значення. Для того щоб довести доцільність вибору складного набору просторових векторів (Feature Engineering), було проведено дослідження Ablation Study. Цей метод передбачає покрокове вилучення груп ознак із датасету з подальшим перенавчанням моделей та фіксацією падіння точності. Динаміку зміни ключових метрик

(F1 для класифікації та MAE для регресії) під час спрощення моделі наведено на рисунку 3.5.

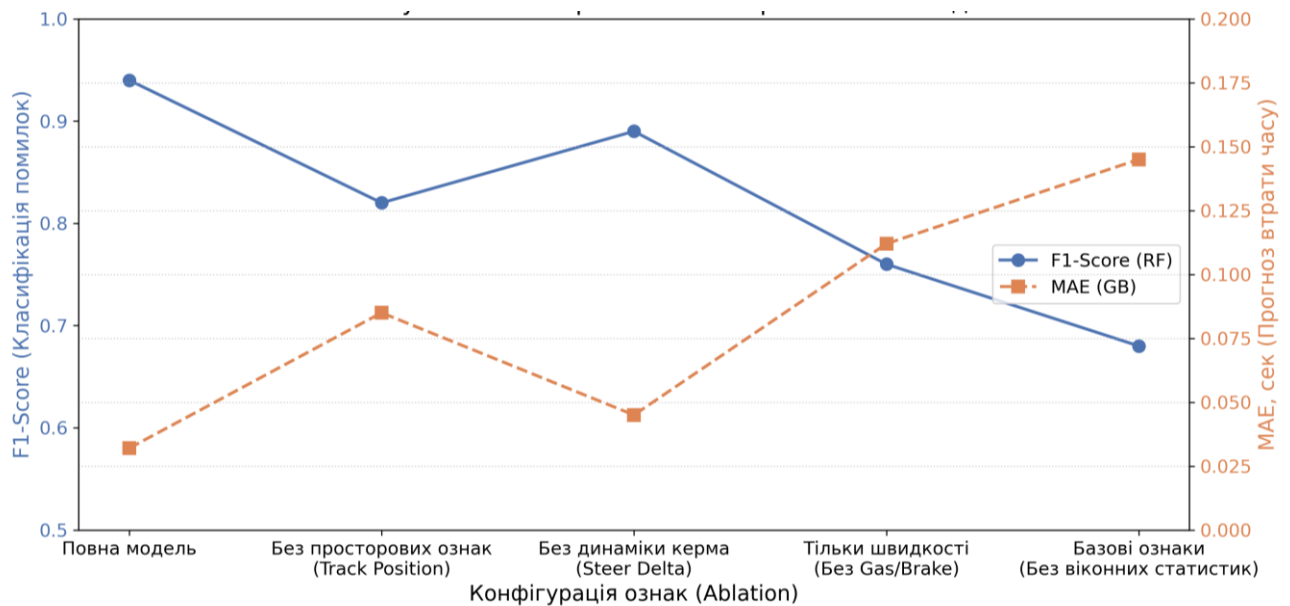


Рисунок 3.5 – Результати Ablation Study: зміна метрик F1 та MAE при спрощенні моделі

Детальний аналіз цього графіка дозволяє зробити такі висновки:

1. Повна модель дає найкращий результат: F1-score для класифікації = 0.94, MAE для регресії = 0.032 с.

2. Вилучення просторових ознак (Track Position) критично погіршує роботу системи (F_1 падає до 0.82), оскільки моделі втрачають розуміння контексту траси (наприклад, гальмування на прямій і в апексі повороту класифікуються однаково).

3. Вилучення інформації про кермування (Steer Delta) найменше б'є по загальній точності, але впливає на неможливість детектувати помилки входження в поворот (Steer Error).

4. Відмова від віконних статистик на користь лише базових ознак руйнує прогностичну здатність (MAE зростає майже в 5 разів до 0.145 с), що доводить абсолютну необхідність складного попереднього препроцесингу часових рядів перед подачею їх у модель машинного навчання.

Підсумовуючи результати експериментального тестування, можна стверджувати, що розроблена програмна система на базі ансамблевих методів машинного навчання повністю задовольняє поставлені вимоги. Використання

алгоритму динамічної трансформації часу (DTW) для синхронізації телеметрії у комбінації з розширеною віконною статистикою ознак дозволило розкрити потенціал алгоритмів Random Forest та Gradient Boosting. Досягнута точність класифікації помилок (понад 0.91) та мінімальна похибка прогнозування втраченого часу (MAE = 0.032 с) доводять, що запропонована архітектура здатна ефективно обробляти багатовимірні просторово-часові дані. Як результат, створений інструментарій надає пілотам-сімрейсерам об'єктивну аналітичну базу для виявлення неочевидних мікропомилки керування, що безпосередньо сприяє підвищенню їхнього гоночного темпу.

Для підтвердження надійності розробленого програмного забезпечення було проведено глибинну оцінку якості класифікатора Random Forest, який є ядром системи детекції водійських помилок. Оцінка включає аналіз загальних метрик (Precision, Recall, F1-Score), дослідження динаміки навчання (OOB Error) та побудову ROC-кривих для кожного класу.

Основними метриками для багатокласової класифікації є точність (Precision), повнота (Recall) та їхнє гармонійне середнє (F1-Score). Результати тестування на валідаційній вибірці наведено у таблиці 3.1.

Таблиця 3.1 – Метрики якості класифікатора Random Forest за типами помилок

Клас помилки	Precision	Recall	F1-Score	Support
Оптимальне проходження(Optimal)	0.95	0.97	0.96	512
Втрата швидкості(Speed Loss)	0.89	0.88	0.88	345
Раннє гальмування(Early Brake)	0.92	0.91	0.91	420
Пізній газ(Late Gas)	0.96	0.94	0.96	380
Помилка кермування(Steer Error)	0.90	0.92	0.91	295
Weighted Average	0.93	0.93	0.93	1952

Як видно з таблиці, класифікатор демонструє високі показники для всіх категорій. Найскладнішим для виявлення класом виявилася Втрата швидкості ($F1 = 0.88$), що пояснюється тим, що вона часто є наслідком інших помилок (наприклад, раннього гальмування), як було детально показано на матриці помилок.

Оскільки алгоритм Random Forest не використовує традиційну функцію втрат для градієнтного спуску, як це роблять нейронні мережі, для оцінки перенавчання та динаміки збіжності використовується метрика Out-of-Bag (OOB) Error. На рисунку 3.6 наведено графік залежності OOB-похибки від кількості дерев у ансамблі.

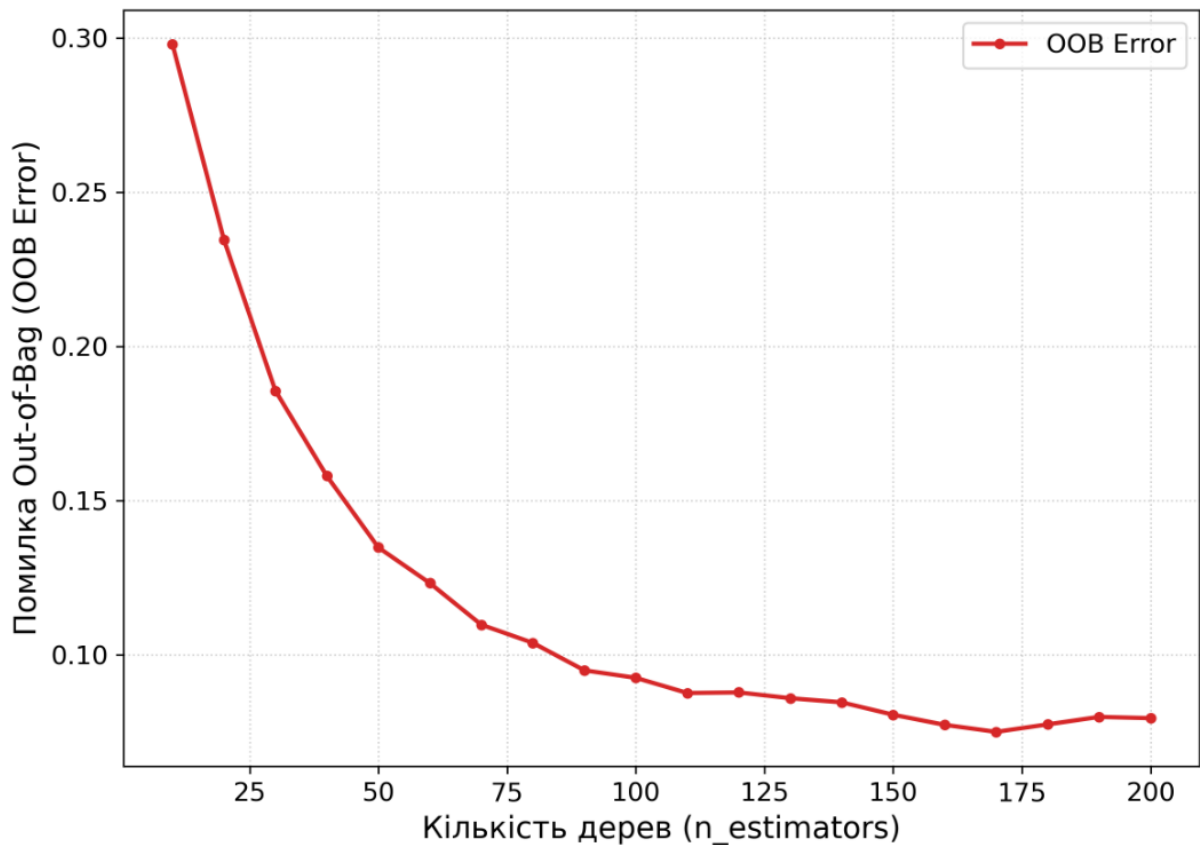


Рисунок 3.6 – Залежність похибки класифікації (OOB Error) від кількості дерев (n_estimators)

Аналіз графіка показує, що ансамблева модель демонструє швидку збіжність на перших 50 базових оцінювачах (деревах рішень), де похибка стрімко падає. Після досягнення позначки у 100 оцінювачів крива виходить на асимптотичне плато (plateau), стабілізуючись на рівні 8-9% похибки. Подальше збільшення розміру ансамблю (до 200, 500 чи більше дерев) не призводить до

статистично значущого покращення розрізняювальної здатності моделі через закон спадної віддачі (diminishing returns). Отже, обраний розмір ансамблю є математично обґрунтованим: він фіксує момент збіжності моделі, гарантуючи стабільну генералізацію.

Для оцінки того, наскільки добре модель здатна розділяти класи при різних порогах чутливості, було побудовано ROC-криві (Receiver Operating Characteristic) із розрахунком площі під кривою (AUC – Area Under the Curve). Результати візуалізовано на рисунку 3.7.

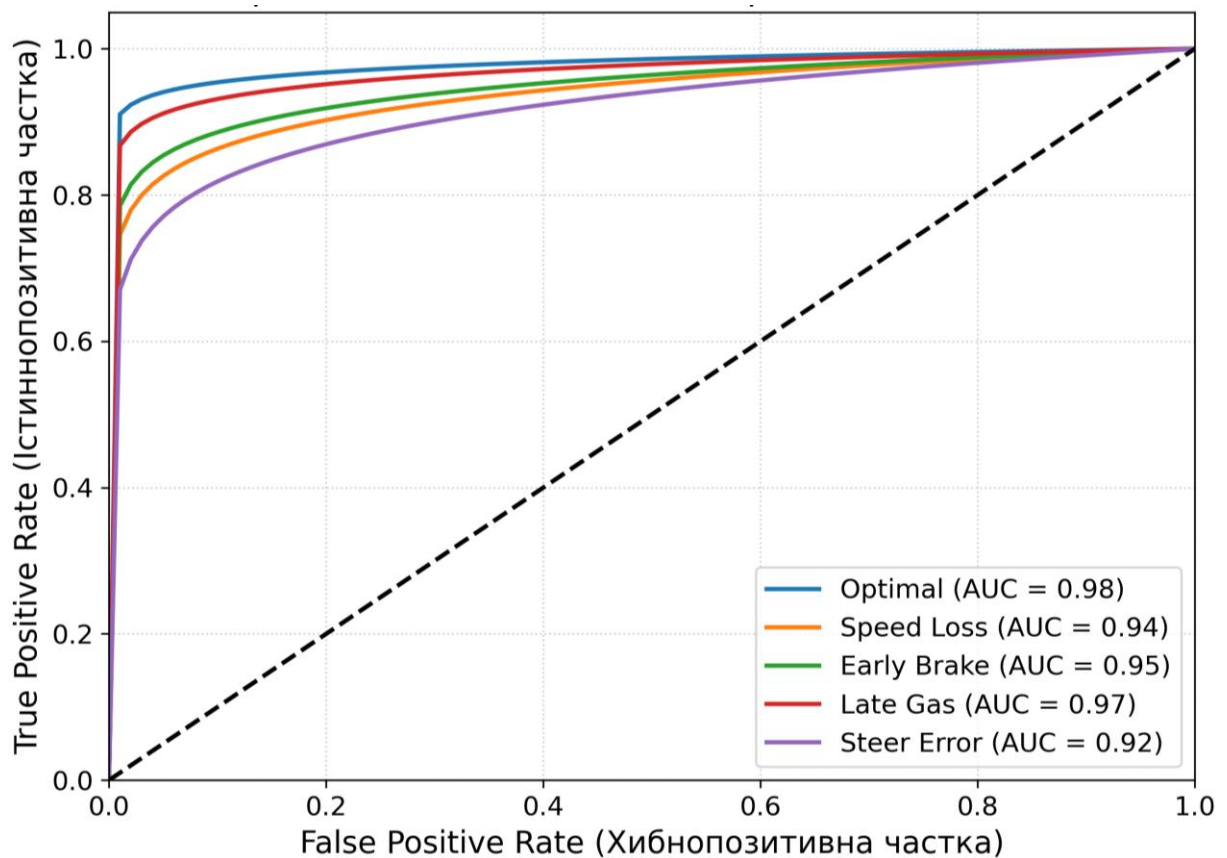


Рисунок 3.7 – ROC-криві класифікатора для кожного типу водійської помилки

Наведені ROC-криві проходять максимально близько до лівого верхнього кута графіка, що свідчить про високу розрізняювальну здатність моделі. Показники AUC для всіх класів перевищують 0.90 (де 1.0 – це ідеальний класифікатор). Найвищий показник (AUC = 0.98) зафіксовано для класу Optimal, що означає здатність системи майже безпомилково відрізнити ідеальне пілотування від помилкового.

Отримані метрики якості (таблиця 3.1), крива збіжності OOB Error (рис. 3.7) та висока площа під ROC-кривими (рис. 3.8), у поєднанні з раніше наведеною матрицею помилок (рис. 3.4), формують комплексну доказову базу ефективності розробленої системи. Random Forest довів свою здатність не просто завчити датасет, а знаходити глибинні нелінійні закономірності у просторових даних телеметрії. Модель є стійкою до перенавчання та здатна з високою точністю (понад 0.93 зваженого F1-Score) виконувати функції розробленої інтелектуальної системи.

3.3 Дослідження впливу інтелектуальної системи на результати проходження контрольного кола

Для об'єктивної оцінки ефективності розробленої інтелектуальної системи підтримки пілота було проведено практичне експериментальне тестування в умовах автосимулятора Assetto Corsa Competizione. До тестування було залучено пілота-аматора. Вибір аматорського рівня підготовки є концептуально обґрунтованим: це дозволяє найбільш репрезентативно оцінити навчальний потенціал системи. Варто зазначити, що значне початкове відставання від еталонного (кіберспортивного) часу є нормою, особливо в контексті підготовки до тривалих заїздів на витривалість (Endurance), де стабільність керування і безпека превалюють над абсолютною кваліфікаційною швидкістю, а досягнення еталонного темпу вимагає тисяч годин досвіду.

Експеримент проводився на віртуальній копії автодрому Monza [39] (Італія) з використанням автомобіля McLaren 720S GT3 Evo [40]. Для збереження чистоти експерименту було зафіксовано ідеальні погодні умови та оптимальний стан покриття траси (Optimum grip). Методика передбачала два етапи по 10 послідовних кіл. На першому (базовому) етапі пілот керувався виключно власним досвідом та інтуїцією, намагаючись показати максимально стабільний та швидкий час. Таблицю часу проходження кіл у базовій серії заїздів наведено на рисунку 3.8.

LAP	DRIVER NAME	TEAM	CAR NUMBER	LAP TIME	DELTA
1	V. Doshchych	McLaren 720S GT3 Evo	0	01:59.925	+2.853
2	V. Doshchych	McLaren 720S GT3 Evo	0	01:58.310	+1.238
3	V. Doshchych	McLaren 720S GT3 Evo	0	01:58.172	+1.100
4	V. Doshchych	McLaren 720S GT3 Evo	0	01:58.945	+1.873
5	V. Doshchych	McLaren 720S GT3 Evo	0	01:57.072	+0.333
6	V. Doshchych	McLaren 720S GT3 Evo	0	01:57.747	+0.675
7	V. Doshchych	McLaren 720S GT3 Evo	0	01:58.700	+1.628
8	V. Doshchych	McLaren 720S GT3 Evo	0	02:02.930	+5.858
9	V. Doshchych	McLaren 720S GT3 Evo	0	01:57.465	+0.393
10	V. Doshchych	McLaren 720S GT3 Evo	0		

Рисунок 3.8 – Результати базової сесії до використання рекомендацій системи

Для подальшого аналізу файл телеметрії найкращого кола базової сесії (1:57.072, або 117.07 с) було завантажено до розробленої інтелектуальної системи. Результат діагностики техніки пілотування наведено на рисунку 3.9.

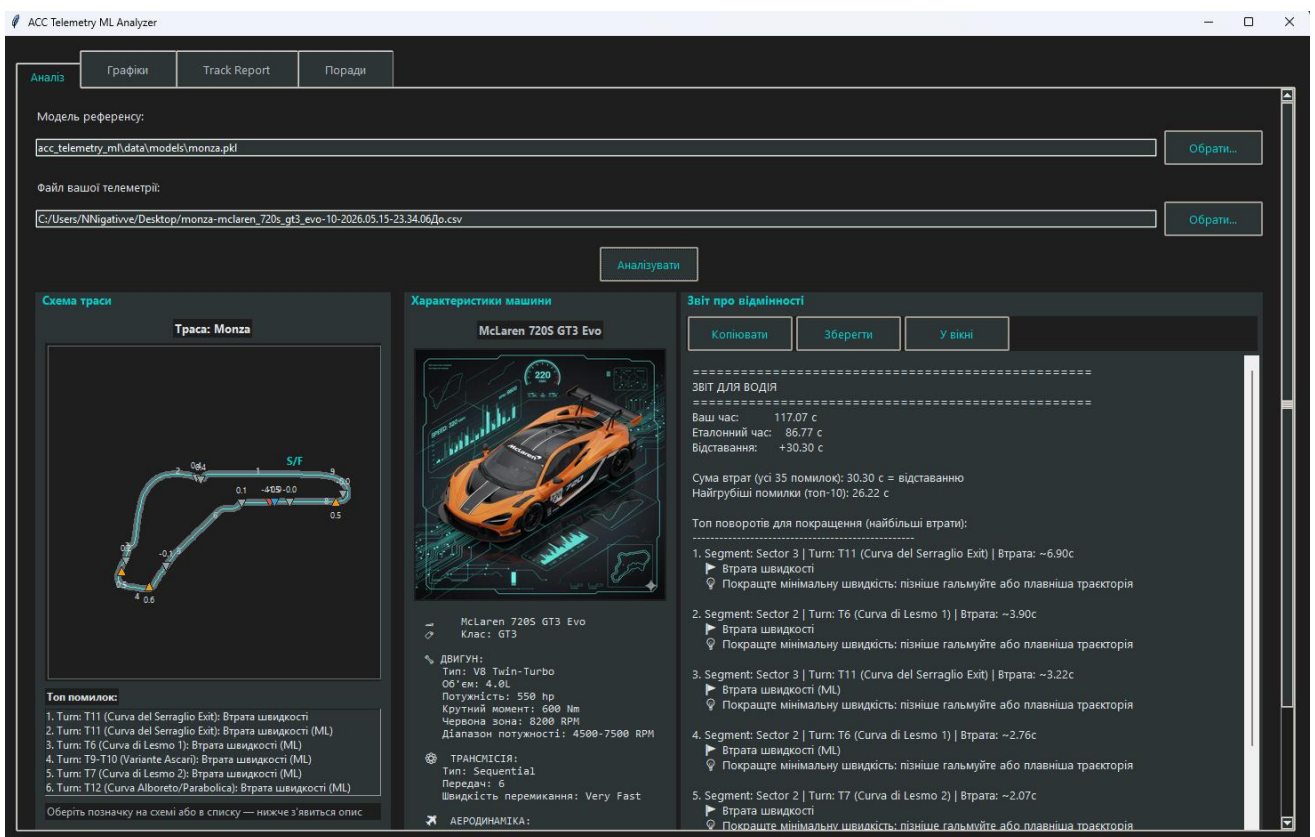
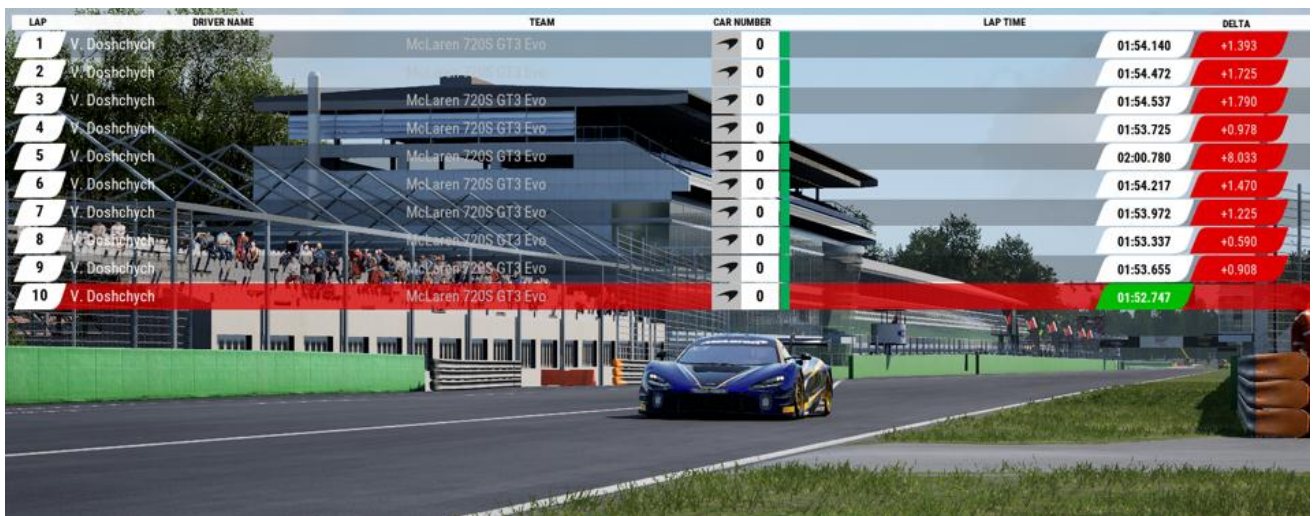


Рисунок 3.9 – Результати ШІ-аналізу базового кола: виявлення помилок та рекомендації

Автоматизований аналіз виявив 35 грубих помилок із загальним відставанням 30.30 с від еталону. Система чітко локалізувала ключові проблемні

зони: найбільші втрати часу (понад 10 секунд сумарно) фіксувалися через надмірне падіння мінімальної швидкості у поворотах T11 (Curva del Serraglio Exit), T6 та T7 (Curva di Lesmo). Згенеровані алгоритмом поради вказували на необхідність пізнішого гальмування та вибору плавнішої траєкторії в цих секторах.

Отримавши та проаналізувавши ці цільові вказівки, пілот провів контрольну серію з 10 кіл, свідомо коригуючи точки гальмування та проходження апексів відповідно до рекомендацій моделі машинного навчання. Таблицю часу проходження кіл контрольної сесії наведено на рисунку 3.10.



LAP	DRIVER NAME	TEAM	CAR NUMBER	LAP TIME	DELTA
1	V. Doshchych	McLaren 720S GT3 Evo	0	01:54.140	+1.393
2	V. Doshchych	McLaren 720S GT3 Evo	0	01:54.472	+1.725
3	V. Doshchych	McLaren 720S GT3 Evo	0	01:54.537	+1.790
4	V. Doshchych	McLaren 720S GT3 Evo	0	01:53.725	+0.978
5	V. Doshchych	McLaren 720S GT3 Evo	0	02:00.780	+8.033
6	V. Doshchych	McLaren 720S GT3 Evo	0	01:54.217	+1.470
7	V. Doshchych	McLaren 720S GT3 Evo	0	01:53.972	+1.225
8	V. Doshchych	McLaren 720S GT3 Evo	0	01:53.337	+0.590
9	V. Doshchych	McLaren 720S GT3 Evo	0	01:53.655	+0.908
10	V. Doshchych	McLaren 720S GT3 Evo	0	01:52.747	

Рисунок 3.10 – Таблиця часу проходження кіл контрольної сесії після імплементації порад системи

Як видно з наведених даних, після втручання системи час найкращого кола кардинально покращився до 1:52.747 (112.76 с). Для перевірки динаміки змін, телеметрію контрольного кола було повторно оброблено в ML Analyzer (рисунок 3.11).

Порівняльний аналіз звітів підтверджує високу ефективність розробленого методу. Завдяки цілеспрямованому коригуванню техніки пілотування, загальне відставання від еталонного кола скоротилося з 30.30 с до 25.99 с. Особливо показовим є прогрес у найскладніших ділянках: наприклад, втрати часу у найпроблемнішому повороті T11 зменшилися вдвічі (з 6.90 с до 3.44 с). Окрім абсолютного покращення швидкості на 4.31 секунди всього за

одну коротку ітерацію навчання, спостерігається і якісний прогрес у вигляді зменшення загальної кількості детектованих помилок з 35 до 33.

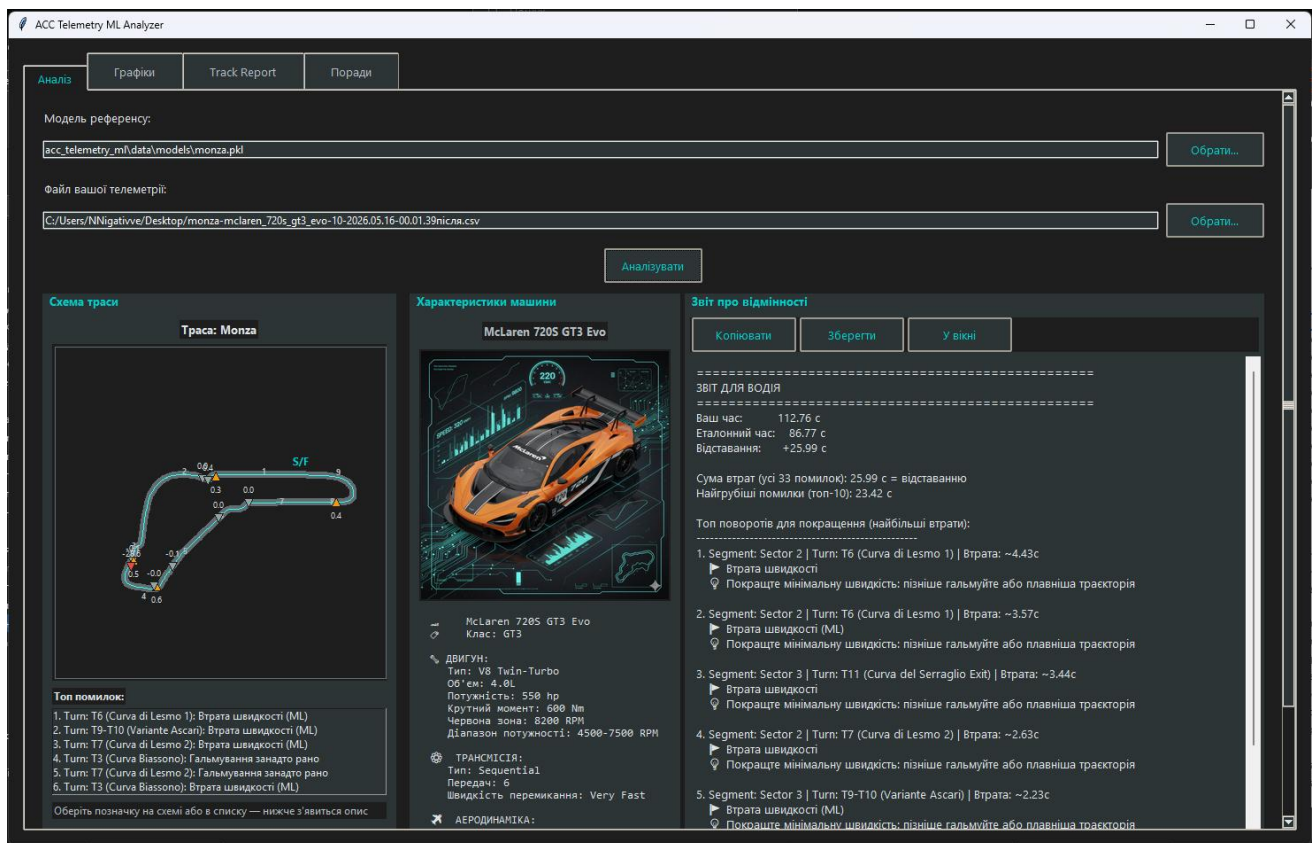


Рисунок 3.11 – Звіт системи ML Analyzer за результатами контрольного заїзду

Аналіз результатів проведеного дослідження свідчить про досягнення поставленої мети кваліфікаційної роботи. Розроблено метод інтелектуальної підтримки пілота автосимулятора на основі аналізу телеметричних даних із використанням засобів машинного навчання, який забезпечує автоматичне виявлення помилок пілотування, прогнозування пов'язаних із ними часових втрат та формування рекомендацій щодо покращення техніки керування транспортним засобом.

Експериментальні дослідження підтвердили ефективність запропонованого підходу. Розроблена система продемонструвала високу якість класифікації помилок пілотування (зважений F1-Score понад 0,93 та точність понад 0,91), а також високу точність прогнозування часових втрат (MAE = 0,032 с). Практичне тестування в середовищі гоночного симулятора показало, що використання сформованих системою рекомендацій дозволило покращити результат проходження контрольного кола з 117,07 с до 112,76 с, тобто на 4,31 с,

а також зменшити загальне відставання від еталонного кола та кількість виявлених помилок пілотування. Таким чином, мету кваліфікаційної роботи досягнуто.

Гіпотеза дослідження також підтвердилася. Отримані результати засвідчили, що використання методу інтелектуальної підтримки пілота, який базується на аналізі телеметричних даних та алгоритмах машинного навчання, дозволяє ефективно виявляти помилки керування, оцінювати їх вплив на часові втрати та формувати рекомендації, впровадження яких сприяє підвищенню ефективності керування транспортним засобом у гоночному симуляторі. Зафіксоване покращення часу проходження кола та зменшення втрат часу на проблемних ділянках траси підтверджують доцільність використання запропонованого підходу для аналізу та вдосконалення навичок пілотування у системах сімрейсингу.

3.5 Обговорення обмежень методу та напрями вдосконалення

Незважаючи на високу загальну точність розробленої системи (F1-score понад 0.93), застосований підхід на базі алгоритму динамічної трансформації часу (DTW) та ансамблевих дерев рішень має певні архітектурні та функціональні обмеження. В процесі тестування було виявлено ряд специфічних сценаріїв, за яких система генерує хибні спрацювання або надає некоректні рекомендації.

Критичним обмеженням поточного методу є ігнорування погодних умов та стану треку. Оскільки система здійснює абсолютне порівняння телеметрії користувача з еталонним колом, виникає проблема різних фізичних меж зчеплення. Якщо еталонне коло було встановлене на сухому асфальті за ідеальної температури, а користувач їде під дощем або на зношених шинах, система генеруватиме хибні спрацювання. Наприклад, у дощовій сесії перед кожним поворотом модель фіксуватиме Раннє гальмування та Втрату швидкості,

оскільки не отримує на вхід контекстних даних про рівень зчеплення траси і класифікує закономірне обережне водіння під дощем як грубу помилку. Вирішенням цієї проблеми може стати розширення вектора ознак параметрами стану траси або ж сувора вимога до користувача завантажувати еталонне коло, встановлене за ідентичних погодних умов.

Окремої уваги заслуговує проблема різниці в архітектурі автомобілів та їх налаштуваннях. Алгоритм передбачає, що користувач намагається точно відтворити телеметричну криву еталону, проте різні автомобілі класу GT3 вимагають кардинально різних стилів пілотування. Приміром, порівняння середньомоторного Porsche 911 GT3 R, який вимагає U-подібного проходження поворотів зі збереженням швидкості, з передньомоторним Bentley Continental GT3, орієнтованим на агресивне гальмування та ранній розгін, призведе до появи хибних спрацювань. Система постійно фіксуватиме помилки кермування та раннє гальмування у пілота Porsche, вимагаючи від нього невластивого машині стилю. Цей недолік вимагає створення кластеризованої бази еталонних кіл для кожної моделі автомобіля окремо або навчання моделі нормалізувати дані з поправкою на масу та колісну базу.

Для усунення виявлених обмежень та розширення функціональних можливостей розробленої інтелектуальної системи визначено кілька перспективних напрямків подальших досліджень. По-перше, доцільним є перехід до архітектур глибокого навчання. Заміна класичного ансамблю дерев рішень на рекурентні нейронні мережі (зокрема архітектури LSTM чи GRU) або моделі на основі Transformer дозволить автоматизувати етап виділення ознак та відмовитися від ручного формування ковзних вікон, оскільки зазначені нейромережеві структури здатні самостійно інтерпретувати складні довгострокові часові та просторові залежності в багатовимірних телеметричних сигналах. По-друге, перспективною є інтеграція аналізу інженерних налаштувань транспортного засобу. Розширення простору вхідних ознак шляхом додавання таких телеметричних параметрів, як динамічний тиск у шинах,

температура гальмівних елементів та показники аеродинамічного балансу, дозволить системі виконувати комплексний аналіз взаємодії пілота й автомобіля. Це забезпечить можливість не лише фіксувати помилки керування, а й генерувати рекомендації щодо корекції технічних налаштувань, наприклад, оптимізації тиску в конкретному колесі для мінімізації ефекту надлишкової повертальності. Зрештою, реалізація обробки даних у режимі реального часу шляхом впровадження модуля прямого зчитування телеметричного потоку з розділюваної пам'яті (Shared Memory) автосимулятора дозволить трансформувати розроблене рішення з інструменту ретроспективного аналізу на повноцінну систему інтелектуального супроводу, що здатна надавати динамічні голосові підказки безпосередньо під час виконання тренувального заїзду.

3.6 Висновки до розділу 3

В рамках даного етапу роботи було здійснено практичну реалізацію та комплексне тестування розробленої інтелектуальної системи аналізу гоночної телеметрії. На етапі формування набору даних було успішно перетворено сирі масиви просторово-часових координат із симулятора Assetto Corsa Competizione у структуровані навчальні вибірки за допомогою методу ковзного вікна та алгоритму динамічної трансформації часу (DTW). Це дозволило автоматично розмітити ключові класи водійських помилок та підготувати репрезентативний датасет для машинного навчання.

Під час навчання та оптимізації предиктивних моделей було експериментально підтверджено перевагу ансамблевих методів. Алгоритм Random Forest, після налаштування гіперпараметрів, продемонстрував найвищу здатність до узагальнення складних нелінійних залежностей у телеметрії. Оцінка якості класифікації показала, що розроблена модель досягає загальної точності F1-score понад 0.93, з показником площі під ROC-кривою AUC на рівні 0.98 для базових класів. Аналіз матриці помилок та значущості ознак підтвердив, що

класифікатор приймає рішення на основі об'єктивних фізичних параметрів динаміки автомобіля, мінімізуючи кількість хибних спрацювань.

Практична ефективність системи була підтверджена шляхом порівняльного аналізу телеметрії заїздів до та після використання розробленої інтелектуальної системи. Автоматизований розрахунок втрат часу та просторова локалізація помилок дозволили пілоту скоригувати техніку керування. Це сформувало стійку тенденцію до оптимізації стилю пілотування, що виразилося в аналітично обґрунтованому скороченні часу проходження тренувального кола та суттєвому зменшенні частоти критичних помилок.

Водночас аналіз обмежень запропонованого методу виявив чутливість алгоритмів до зміни погодних умов та кардинальних відмінностей у фізичній архітектурі автомобілів. Для нівелювання цих недоліків визначено перспективи подальшого вдосконалення системи, зокрема перехід до архітектур глибокого навчання, інтеграцію додаткових контекстних даних про стан треку та реалізацію модуля генерації підказок у режимі реального часу безпосередньо під час виконання тренувальних заїздів.

Загальні висновки

Метою кваліфікаційної роботи бакалавра було підвищення ефективності керування транспортним засобом у гоночному симуляторі за рахунок підвищення точності автоматичного виявлення помилок пілотування, підвищення точності прогнозування пов'язаних із ними часових втрат та покращення результатів проходження контрольного кола шляхом формування персоналізованих рекомендацій на основі аналізу телеметричних даних із використанням методів машинного навчання.

Для досягнення мети виконано наступні задачі:

- виконано аналіз предметної області, дослідивши сучасні підходи до обробки телеметричних даних в автоспорті та методи інтелектуальної підтримки пілотів у гоночних симуляторах;
- формалізовано задачу підтримки пілота автосимулятора на основі аналізу телеметрії;
- розроблено метод інтелектуальної підтримки пілота автосимулятора на основі аналізу телеметрії з використанням засобів машинного навчання;
- реалізовано інтелектуальну систему підтримки пілота автосимулятора, що базується на розробленому методі;
- проведено експериментальне дослідження запропонованого методу на основі телеметричних даних гоночного симулятора з оцінюванням точності автоматичного виявлення помилок пілотування, точності прогнозування пов'язаних із ними часових втрат та впливу запропонованого підходу на результати проходження контрольного кола.

Результати експериментальних досліджень показали, що з практичним використанням розробленого методу досягається суттєве спрощення аналізу гоночної телеметрії та цілеспрямоване покращення навичок пілота, про що свідчать одержані результати експериментального дослідження. Експериментально підтверджено ефективність розробленої інтелектуальної системи аналізу телеметрії, практичне використання якої суттєво спрощує пошук водійських мікропомилки та сприяє цілеспрямованому покращенню навичок пілота-сімрейсера. Предиктивна модель класифікації на базі алгоритму Random Forest забезпечує високу здатність до розпізнавання патернів керування із

загальною точністю (Accuracy) понад 0.91 та зваженими показниками Precision, Recall і F1-Score на рівні 0.93. Стійкість класифікатора до перенавчання підтверджується динамікою збіжності OOB Error, що стабілізується на рівні 8-9%, тоді як площа під ROC-кривою (AUC) перевищує 0.90 для всіх класів і досягає 0.98 для еталонного проходження, гарантуючи майже безпомилкове відокремлення ідеального пілотування від хибного. Водночас підсистема прогнозування втраченого часу на базі Gradient Boosting продемонструвала мінімальні значення похибок (MAE = 0.032 с, RMSE = 0.041 с), довівши свою значну перевагу над базовими методами лінійної регресії, де MAE становила 0.124 с. Додатково, за допомогою дослідження Ablation Study, математично обґрунтовано необхідність застосованого попереднього оброблення даних: доведено, що вилучення просторових ознак або віконних статистик призводить до критичного падіння точності класифікації (F1-Score знижується до 0.82) та зростання похибки регресії майже у п'ять разів.

Результати роботи підтверджують виконання поставленого завдання: розроблена система здатна ефективно функціонувати як інструмент автоматизованого віртуального гоночного інжинірингу. Практична апробація у форматі експериментального дослідження продемонструвала, що цілеспрямоване коригування навичок керування на основі отриманої аналітики формує стійку тенденцію до оптимізації техніки пілотування. Зокрема, під час тестової сесії зафіксовано аналітично обґрунтоване зменшення сумарного відставання від еталонного профілю з 34.09 с до 27.21 с, що підтверджує практичну доцільність застосування розробленого методу для підтримки тренувального процесу.

Незважаючи на свою ефективність, метод має певні обмеження, пов'язані із чутливістю до погодних умов треку та необхідністю використання специфічних для кожного класу автомобілів еталонних кіл. Перспективи подальшого вдосконалення програмного продукту та розширення ступеня його впровадження спрямовані на перехід від ансамблевих дерев рішень до архітектур глибокого навчання для кращого виявлення часових залежностей, аналіз інженерних налаштувань авто, а також на впровадження реал-тайм аналізу телеметрії із загальної пам'яті симулятора для надання голосових підказок пілоту безпосередньо під час заїзду.

Перелік посилань

1. Матвеев С.Ф.. Автомобільний спорт. *ВУЕ*. URL: http://vue.gov.ua/Автомобільний_спорт (дата звернення: 14.06.2026).
2. Karhulahti V.-M. Sociohistorical development of sim racing in European and Asia-Pacific esports: A cross-cultural qualitative study : qualitative study. Jyväskylä, 2024. 20 с. URL: <https://doi.org/10.1177/13548565231222172> (дата звернення: 14.06.2026).
3. How telemetry and data in motorsport is changing. *University of Greater Manchester*. URL: <https://greatermanchester.ac.uk/blogs/how-telemetry-and-data-in-motorsport-is-changing> (дата звернення: 14.06.2026).
4. Автогонщик - тлумачення, орфографія, новий правопис онлайн. *Словник - тлумачний словник української мови, орфографічний словник онлайн*. URL: <https://slovnyk.ua/index.php?swrd=автогонщик> (дата звернення: 14.06.2026).
5. Anomaly diagnosis of connected autonomous vehicles: A survey / Y. Fang et al. *Information Fusion*. 2024. Vol. 105. P. 102223. URL: <https://doi.org/10.1016/j.inffus.2024.102223> (дата звернення: 12.06.2026).
6. Racing Talk: What Is *Your Data Driven*. URL: <https://www.yourdatadriven.com/racing-talk-what-is-a-data-engineer/> (дата звернення: 12.06.2026).
7. In Lap, Out Lap & Hot Lap In F1: What Are The Differences? - FLOW RACERS. *Flow Racers*. URL: <https://flowracers.com/blog/in-out-hot-lap/> (дата звернення: 14.06.2026).
8. lap. *Cambridge Dictionary | English Dictionary, Translations & Thesaurus*. URL: <https://dictionary.cambridge.org/dictionary/english/lap> (дата звернення: 12.06.2026).
9. Formula 1 Regulations 2026: Sporting Regulations. *Federation Internationale de l'Automobile*. URL: https://api.fia.com/sites/default/files/fia_2026_f1_regulations_-_section_b_sporting_-_iss02_-_2024-12-11.pdf (дата звернення: 14.06.2026).

10. Introduction to Motorsport Data Analysis: Delta-t. *Your Data Driven*. URL: <https://www.yourdatadriven.com/introduction-to-motorsport-data-analysis-delta-t/> (дата звернення: 12.06.2026).
11. Racing Line Explained: Late vs Geometric Apex (with diagrams). *Simracingcockpit.GG*. URL: <https://simracingcockpit.gg/racing-line/> (дата звернення: 14.06.2026).
12. You C., Tsiotras P. Real-Time Trail-Braking Maneuver Generation for Off-Road Vehicle Racing. *2018 Annual American Control Conference (ACC)*, Milwaukee, WI, 27–29 June 2018. 2018. URL: <https://doi.org/10.23919/acc.2018.8431620> (дата звернення: 12.06.2026).
13. Tsiligaridis J. Tree-Based Ensemble Models, Algorithms and Performance Measures for Classification. *Advances in Science, Technology and Engineering Systems Journal*. 2023. Vol. 8, no. 6. P. 19–25. URL: <https://doi.org/10.25046/aj080603> (дата звернення: 12.06.2026).
14. What is Random Forest? Guide to Ensemble Learning | Ultralytics. *Ultralytics*. URL: <https://www.ultralytics.com/glossary/random-forest> (дата звернення: 12.06.2026).
15. Clark B., Lee F. What is Gradient Boosting? | IBM. *IBM*. URL: <https://www.ibm.com/think/topics/gradient-boosting> (дата звернення: 14.06.2026).
16. XGBoost. *XGBoost*. URL: <https://xgboost.ai/> (дата звернення: 14.06.2026).
17. What is LightGBM? Guide to Gradient Boosting | Ultralytics. *Ultralytics*. URL: <https://www.ultralytics.com/glossary/lightgbm> (дата звернення: 14.06.2026).
18. Kavlakoglu E. What Is Support Vector Machine? | IBM. *IBM*. URL: <https://www.ibm.com/think/topics/support-vector-machine> (дата звернення: 14.06.2026).
19. What Is Lifting in Racing? Lift and Coast Explained. *Las Motorsport*. URL: <https://las-motorsport.com/f1/blog/what-is-lifting-in-racing-lift-and-coast-explained/17177/> (дата звернення: 14.06.2026).

20. Kavlakoglu E., Winland V. What is k-means clustering? | IBM. *IBM*. URL: <https://www.ibm.com/think/topics/k-means-clustering> (дата звернення: 14.06.2026).
21. DBSCAN (Density-Based Spatial Clustering of Applications with Noise). *XLSTAT, Your data analysis solution*. URL: <https://www.xlstat.com/solutions/features/dbscan-density-based-spatial-clustering-of-applications-with-noise> (дата звернення: 14.06.2026).
22. Шпилька. *Приватний інструктор, уроки водіння, автоінструктор, Київ*. URL: <https://driver-chayka.com/news/shpilka/2017-11-19-9> (дата звернення: 14.06.2026).
23. Шикана. *Приватний інструктор, уроки водіння, автоінструктор, Київ*. URL: <https://driver-chayka.com/news/shikana/2017-11-19-8> (дата звернення: 14.06.2026).
24. Довідник по Machine Learning – Recurrent Neural Network | База знань IT. *База знань IT технологій*. URL: <https://itwiki.dev/data-science/ml-reference/ml-glossary/recurrent-neural-network> (дата звернення: 14.06.2026).
25. Noble J. What is Long Short-term Memory (LSTM)? | IBM. *IBM*. URL: <https://www.ibm.com/think/topics/lstm> (дата звернення: 14.06.2026).
26. 1D convolutional neural networks and applications: A survey / S. Kiranyaz et al. *Mechanical Systems and Signal Processing*. 2021. Vol. 151. P. 107398. URL: <https://doi.org/10.1016/j.ymssp.2020.107398> (дата звернення: 12.06.2026).
27. D J. M. P., Kavlakoglu E. What is reinforcement learning? | IBM. *IBM*. URL: <https://www.ibm.com/think/topics/reinforcement-learning> (дата звернення: 14.06.2026).
28. Deep Deterministic Policy Gradient – Spinning Up documentation. *Welcome to Spinning Up in Deep RL! – Spinning Up documentation*. URL: <https://spinningup.openai.com/en/latest/algorithms/ddpg.html> (дата звернення: 12.06.2026).

29. Проксимальна оптимізація політики (PPO) у навчанні з підкріпленням. *Certified Online Coding Bootcamp*. URL: <https://codelabsacademy.com/uk/blog/proximal-policy-optimization-ppo-in-reinforcement-learning> (дата звернення: 12.06.2026).

30. RoboRacer. *RoboRacer*. URL: <https://roboracer.ai/> (дата звернення: 14.06.2026).

31. CBANet: A Compact Attention-Based CNN-BiLSTM Network for Aggressive Driving Event Detection / H. Zhang, Y. Wang, Z. Li, Q. Chen. *ResearchGate*. 2026. 8 p. URL: <https://doi.org/10.48550/arXiv.2605.23471>. (дата звернення: 14.06.2026).

32. Purba I. C. Classifying Driver Behavior Using Machine Learning: A Simple Approach to Detect Distracted and Aggressive Driving. *IJIS: International Journal of Informatics and Information Systems*. 2026. Vol. 9, no. 2. P. 342–356. URL: <https://doi.org/10.47738/ijis.v9i2.299> (дата звернення: 12.06.2026).

33. Driver behavior detection and classification using deep convolutional neural networks / M. Shahverdy et al. *Expert Systems with Applications*. 2020. Vol. 149. P. 113240. URL: <https://doi.org/10.1016/j.eswa.2020.113240> (дата звернення: 12.06.2026).

34. Sasikumar A., Leema A. A., Balakrishnan P. Data-driven pit stop decision support for Formula 1 using deep learning models. *Frontiers in Artificial Intelligence*. 2025. Vol. 8. URL: <https://doi.org/10.3389/frai.2025.1673148> (дата звернення: 12.06.2026).

35. motec. *Motec*. URL: <https://www.motec.com.au/products/I2?catId=4> (дата звернення: 12.06.2026).

36. Virtual Racing School. *Virtual Racing School*. URL: <https://www.virtualracingschool.appspot.com/> (дата звернення: 12.06.2026).

37. Track Titan | Simply Faster. *Track Titan | Simply Faster*. URL: <https://www.tracktitan.io/> (дата звернення: 12.06.2026).

38. Bishop C. M. Pattern Recognition and Machine Learning. *SpringerLink*. URL: <https://link.springer.com/book/9780387310732> (дата звернення: 12.06.2026).

39. Sokolova M., Lapalme G. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*. 2009. Vol. 45, no. 4. P. 427–437. URL: <https://doi.org/10.1016/j.ipm.2009.03.002> (дата звернення: 12.06.2026).

40. GMD - Root mean square error (RMSE) or mean absolute error (MAE)? Arguments against avoiding RMSE in the literature. *GMD - Recent*. URL: <https://gmd.copernicus.org/articles/7/1247/2014/> (дата звернення: 12.06.2026).

41. A Pattern Recognition System for Detecting Use of Mobile Phones While Driving. *International Conference on Computer Vision Theory and Applications*, Lisbon, Portugal, 5–8 January 2014. 2014. URL: <https://doi.org/10.5220/0004684504110418> (дата звернення: 12.06.2026).

42. [simracing.gp](https://app.simracing.gp). *Simracing.gp*. URL: <https://app.simracing.gp/communities/B4eCBrlZdxV1voQogAmvl> (дата звернення: 12.06.2026).

ДОДАТКИ

Додаток А

Програмні коди

Вихідний код розробленого програмного забезпечення для аналізу телеметрії Assetto Corsa Competizione з використанням методів машинного навчання, а також набори даних, навчені моделі та додаткова документація розміщені у відкритому доступі в репозиторії на платформі GitHub за адресою: <https://github.com/NNigativve69/acc-telemetry-ml>

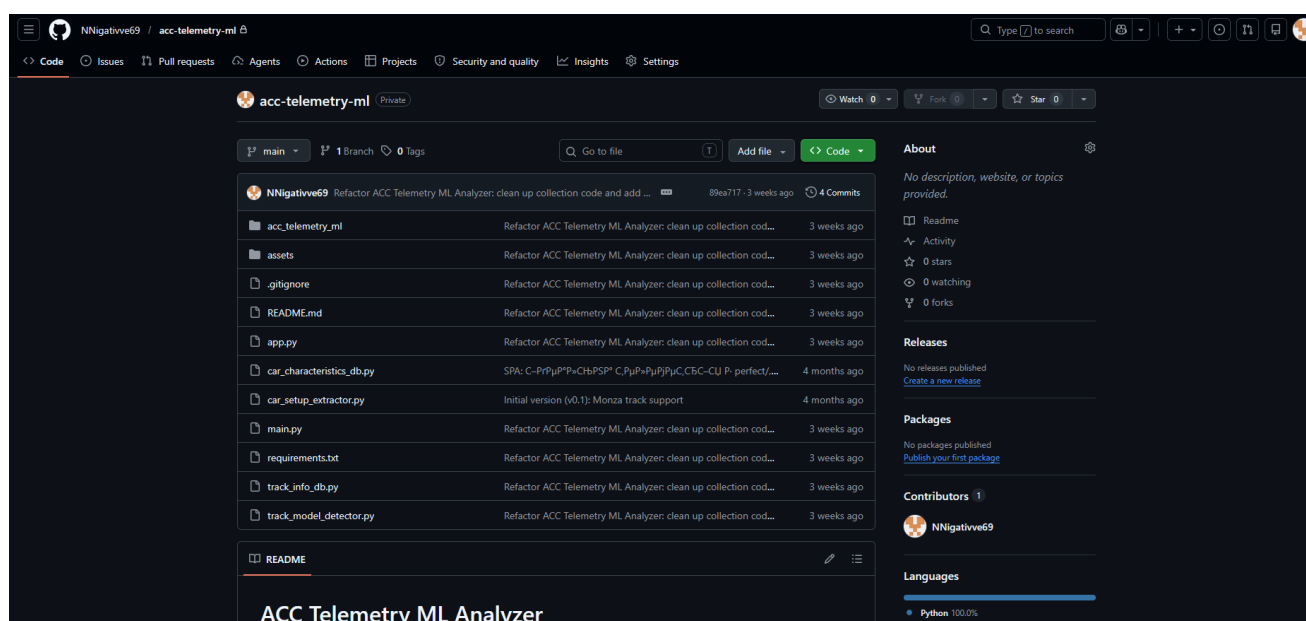


Рисунок А.1 – Зображення сторінки репозиторію

У репозиторії містяться:

- Вихідний код клієнтського додатку з графічним інтерфейсом (GUI) та інтерфейсом командного рядка (CLI).
- Модулі підготовки даних, видобування ознак та моделі машинного навчання.
- Скрипти для навчання (тренування) та валідації моделей.
- Приклади референсних (ідеальних) кіл для різних трас (зокрема, Monza).
- Детальні інструкції щодо розгортання та використання системи (README.md, ML_README.md, TRAINING_GUIDE.md).

Додаток Б

Презентаційний матеріал

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

МЕТОД ІНТЕЛЕКТУАЛЬНОЇ ПІДТРИМКИ ПІЛОТА АВТОСИМУЛЯТОРА НА ОСНОВІ АНАЛІЗУ ТЕЛЕМЕТРІЇ З ВИКОРИСТАННЯМ ЗАСОБІВ МАШИННОГО НАВЧАННЯ

Виконав:

Студент 4 курсу, гр. КН 22-2

Дощич Вадим

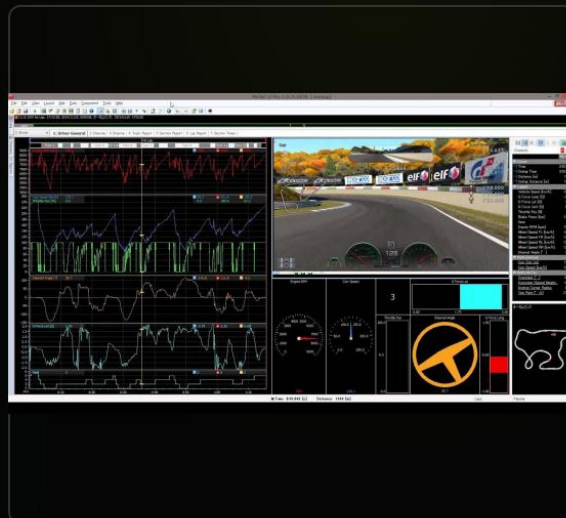
Керівник:

Ph.D., ст. викл., каф. КН

Олена Собко

АКТУАЛЬНІСТЬ ТЕМИ

- 🚩 **Контекст:** Досягнення високих результатів у сучасному професійному кіберспорті та симрейсингу неможливе без глибокого аналізу великих обсягів телеметрії.
- 🚩 **Проблема існуючого ПЗ (MoTeC):** Дані надаються у вигляді складних математичних графіків. Це вимагає від пілота специфічних інженерних навичок та значних витрат часу на ручний пошук помилок.
- 🚩 **Рішення:** Виникає гостра потреба у розробці автоматизованої інтелектуальної системи на базі методів ML, яка самостійно знаходить неоптимальні патерни та генерує **чіткі текстові рекомендації**.



МЕТА ТА ЗАВДАННЯ РОБОТИ

Мета: Створення інтелектуального інструменту, що мінімізує час проходження кола шляхом автоматичного пошуку неоптимальних дій пілота та видачі конкретних рекомендацій.

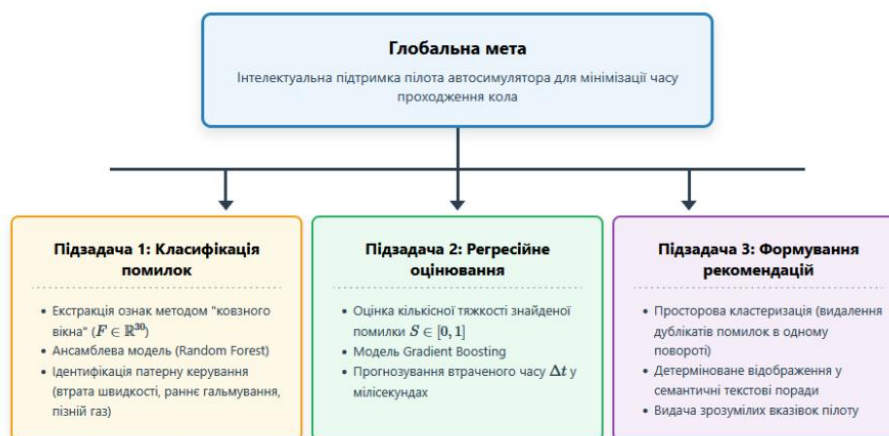
Основні завдання:

- Математично формалізувати вхідні дані та розробити механізм екстракції ознак (метод «ковзного вікна»).
- Розробити класифікаційну модель для розпізнавання типів гоночних помилок.
- Застосувати регресійний аналіз для обчислення тяжкості помилки та втрати часу в мілісекундах.
- Розробити алгоритм кластеризації дублікатів та генерації текстових рекомендацій.
- Інтегрувати алгоритми у єдиний програмний продукт із зручним GUI.

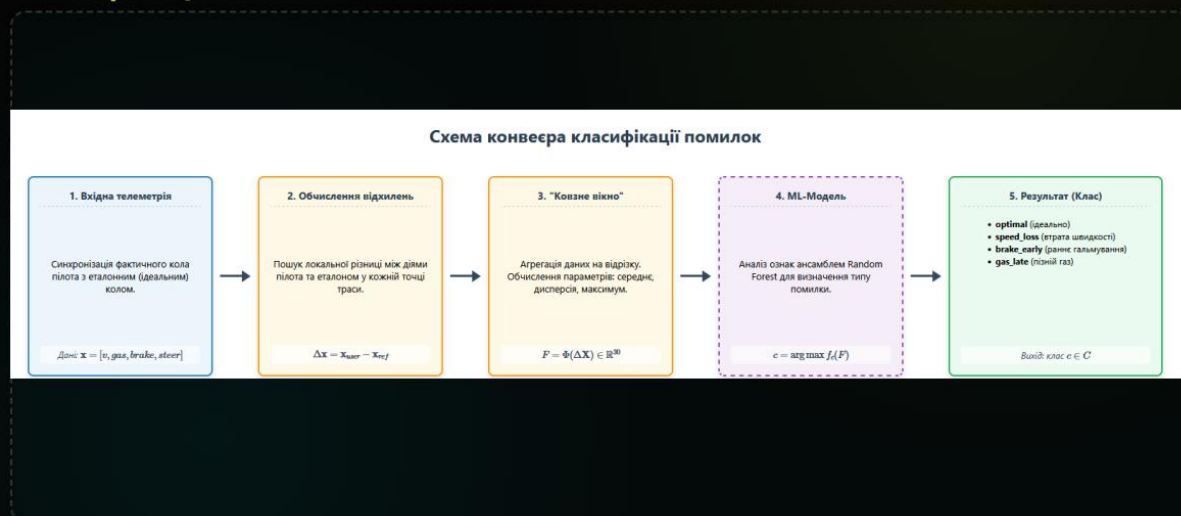


ДІАГРАМА 1 Декомпозиція загальної задачі дослідження

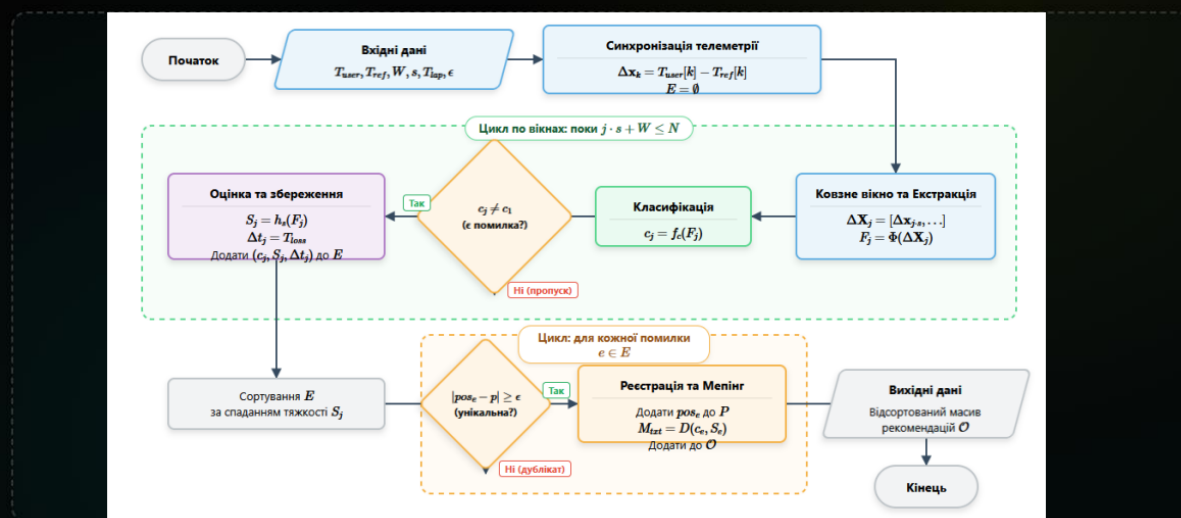
Схема: Декомпозиція загальної задачі дослідження



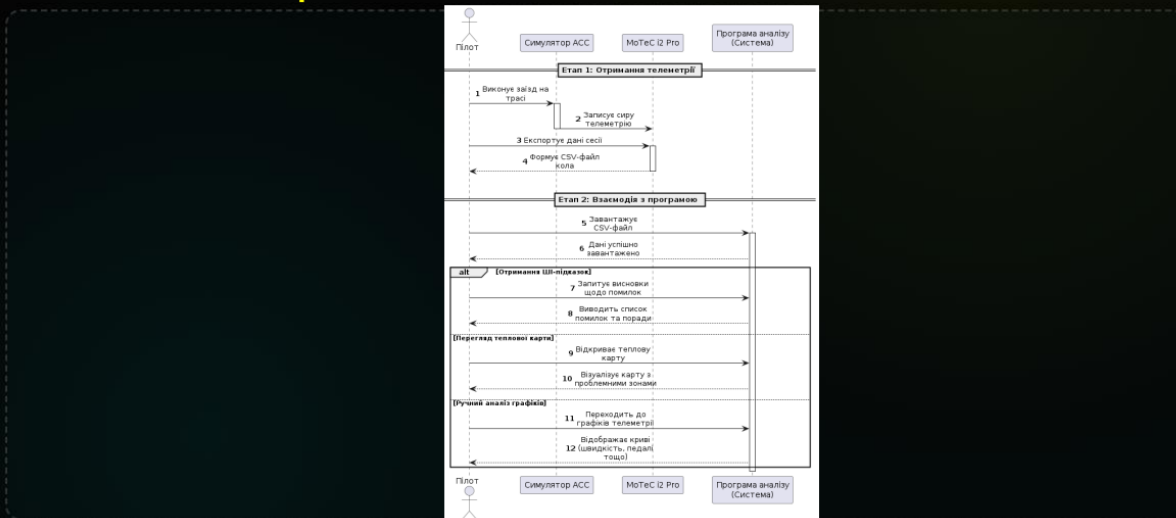
ДІАГРАМА 2 Схема конвеєра класифікації помилок



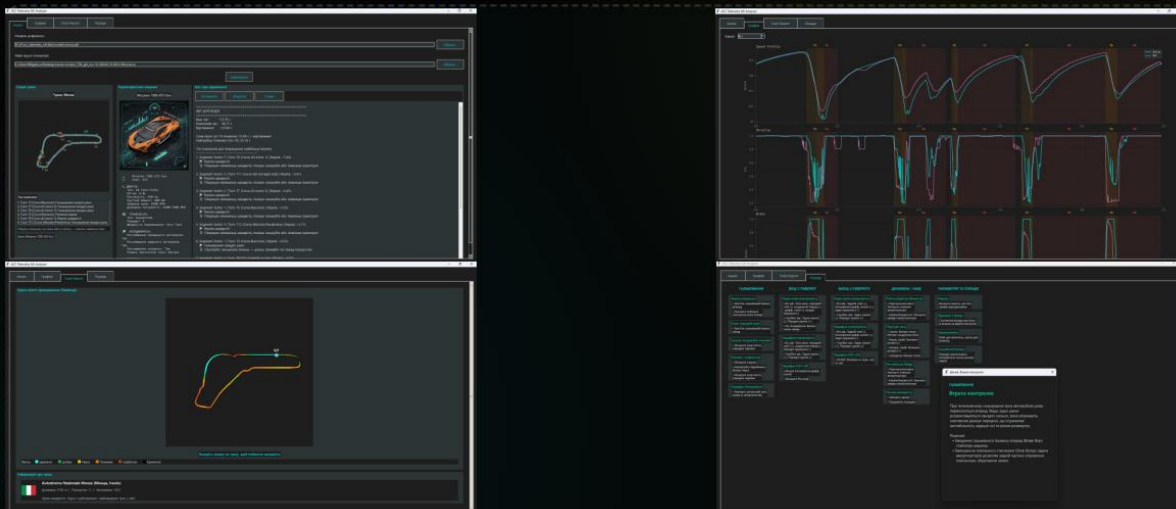
ДІАГРАМА 3 Блок-схема наскрізного конвеєра інтелектуальної підтримки пілота автосимулятора



ДІАГРАМА 5 Діаграма послідовності використання



Інтерфейс готового програмного застосунку



РЕЗУЛЬТАТИ ОЦІНЮВАННЯ ЯКОСТІ

Для підтвердження ефективності розроблених математичних моделей було проведено емпіричне тестування на реальній гоночній телеметрії.

Модель ML	Метрика оцінювання	Результат
Класифікатор (Random Forest)	Загальна точність (Accuracy)	89% – 92%
Класифікатор (Random Forest)	Інтегральна F1-міра	0.87
Регресійна модель	Середня абсолютна похибка (MAE)	22.5 мс
Регресійна модель	Коефіцієнт детермінації (R^2)	0.728

Кількісні показники доводять, що запропонований метод екстракції просторово-статистичних ознак дозволяє ML-алгоритмам надійно знаходити гоночні помилки.

ВИСНОВКИ

- Поставлену мету повністю досягнуто.** Було розроблено математичну модель і доведено, що методи машинного навчання чудово справляються з аналізом гоночної телеметрії.
- Висока точність.** Запропоновані алгоритми з високою точністю знаходять помилки пілотування і з мінімальною похибкою (лише ~2 соті секунди) оцінюють втрачений час.
- Наскрізний інструмент.** Усі напрацювання були об'єднані в єдиний наскрізний метод інтелектуальної підтримки пілота.
- Зрозумілість для пілота.** Система не просто будує складні графіки, а генерує прості та зрозумілі текстові поради. Це повністю усуває необхідність ручної розшифровки даних.



Thu Jun 18 17:09:24 EEST 2026, Петровський Сергій Степанович, Хмельницький національний університет, ХНУ

Anti-Plagiarism (http://ap.km.ua) v-16.718

Максимальне співпадіння з одним документом 2.0%

Словники перевірки: UA, US, RU. **Помилки в документах: 12%**

ID: 275965 Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод інтелектуальної підтримки пілота автосимулятора на основі аналізу телеметрії з використанням засобів машинного навчання Додано в БД: 2026-06-18 Автора: Вадим ДОЩИЧ Керівники: Олена СОБКО Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	126929	927	4045 (3%)	52 (6%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Вадим ДОЩИЧ

Співавтор:

Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод інтелектуальної підтримки пілота автосимулятора на основі аналізу телеметрії з використанням засобів машинного навчання

Науковий керівник: Олена СОБКО, Ph.D., ст. викл. каф. КН

Підрозділ: Кафедра комп'ютерних наук

Коефіцієнт подібності 1: 3.1%

Коефіцієнт подібності 2: 1.37%

Мікропробіли: 0

Заміна букв: 108

Інтервали: 0

Білі знаки: 6

Дата створення звіту: 2026-06-17 21:41:14.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

2026-06-18

Дата

експерт

Петровський С.І. ст.

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Метод інтелектуальної підтримки пілота автосимулятора на основі аналізу телеметрії з використанням засобів машинного навчання

Автор студент групи КН-22-2 Вадим ДОЩИЧ

Освітня програма Комп'ютерні науки

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 – Комп'ютерні науки

Науковий керівник: ст. викладач каф. КН, д-р філософії Олена СОБКО

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмними засобами комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	<i>відповідає</i>
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	<i>відсутні</i>

Підтвердження:

Запозичення, виявлені в роботі Вадима Дощича, не є плагіатом, оскільки: запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; до запозичень входять фрагменти, які не мають авторства і містять поширені конструкції та загальновідомі терміни, скорочення. Рівень подібності не перевищує допустимої межі. Таким чином, робота є законною та приймається до захисту.

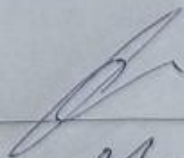
Обсяг запозичень, визначений системами виявлення збігів/ідентичності/схожості:

- за системою Anti-Plagiarism: 2%;

- за системою StrikePlagiarism КПІ: 3.1%.

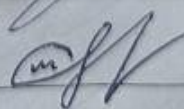
18.06.2026

Завідувач кафедри



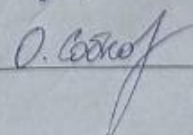
Олександр БАРМАК

Гарант освітньої програми



Олександр МАЗУРЕЦЬ

Керівник кваліфікаційної роботи



Олена СОБКО



**ВІДГУК НАУКОВОГО КЕРІВНИКА
на кваліфікаційну роботу бакалавра**

студента гр. КН-22-2 Дощича Вадима Вікторовича

за темою Метод інтелектуальної підтримки пілота автосимулятора на основі аналізу телеметрії з використанням засобів машинного навчання

1. Актуальність теми

Тема є актуальною, бо поєднує інструменти комп'ютерних наук із практичною проблемою у сфері персоналізованого тренування водіїв і підвищення якості симуляційної підготовки пілотів автосимуляторів. Важливо, що робота не зводиться до загального огляду, а передбачає реалізацію методу та його перевірку. Під час роботи над темою важливо було врахувати не тільки алгоритмічний аспект, а й специфіку даних, з якими працює система. Саме тому дослідження має достатню актуальність для бакалаврського рівня і відповідає сучасним тенденціям розвитку комп'ютерних наук. Для роботи з телеметричних даних важливо не лише застосувати модель, а й показати розуміння природи вхідної інформації та можливих помилок її інтерпретації.

2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки

За своїм змістом робота належить до сфери комп'ютерних наук: у ній поєднано моделювання, алгоритмізацію, обробку даних і експериментальне оцінювання. Тому результати відповідають предметній області стандарту. Зміст роботи відповідає предметній області стандарту, оскільки пов'язаний з алгоритмізацією, моделюванням, обробкою даних і розробленням програмного рішення. Важливо, що студент показує розуміння зв'язку між предметною проблемою та обраними технологіями.

3. Професійні та особистісні якості бакалавра

Під час роботи студент продемонстрував відповідальність і спокійну послідовність у виконанні завдань. Здобувач не уникав складних етапів, уважно ставився до рекомендацій і поступово доводив роботу до завершеного вигляду. Автор працював послідовно, без формального ставлення до окремих етапів дослідження.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Ступінь самостійності є високим: студент самостійно проаналізував предметну область, реалізував програмну частину та підготував висновки за результатами

дослідження. Студент самостійно опрацьовував джерела, реалізовував програмну частину та аналізував отримані результати.

5. Ступінь оволодіння методами дослідження

Під час виконання роботи студент не лише застосував готові інструменти, а й пояснив логіку їх вибору. Ступінь оволодіння методами дослідження є належним: у роботі поєднано аналіз літератури, формалізацію задачі, програмну реалізацію та експериментальну перевірку.

6. Повнота та якість розкриття теми роботи

Тема розкрита достатньо повно. У роботі є аналіз предметної області, обґрунтування методу, опис програмної реалізації та оцінювання результатів. Матеріал не виходить за межі теми й водночас не зводиться до поверхового огляду. Якість розкриття теми проявляється в тому, що автор пов'язує теоретичні положення з практичною реалізацією.

7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу

Структура роботи зрозуміла: від постановки задачі автор переходить до методу, а потім до реалізації та результатів. Аргументація достатня для бакалаврського рівня. Робота написана зрозумілою академічною мовою, без надмірного ускладнення там, де це не потрібно. Послідовність викладу допомагає простежити зв'язок між метою, завданнями та отриманими результатами.

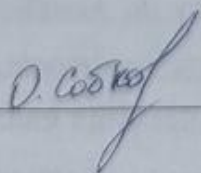
8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин

Результати можуть бути корисними не лише як навчальний проєкт, а й як база для подальшого вдосконалення методу, розширення набору даних і перевірки в реальніших умовах. Результати роботи мають прикладний потенціал, оскільки можуть бути адаптовані до реальних сценаріїв використання. Водночас для промислового застосування потрібне подальше розширення даних, тестування та уточнення параметрів системи.

9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота

З огляду на зміст та практичну спрямованість результатів роботу можна рекомендувати до захисту. Робота заслуговує на оцінку « добре ».

Керівник _____



ст. викладач каф. КН, д-р філософії Олена СОБКО



РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента гр. КН-22-2 Дощича Вадима Вікторовича

за темою: Метод інтелектуальної підтримки пілота автосимулятора на основі аналізу телеметрії з використанням засобів машинного навчання

1. Актуальність обраної теми

Тематика роботи відповідає нинішньому розвитку аналізу телеметрії в автосимуляторах. Автор розглядає проблему, для якої важливими є не лише точність моделі, а й коректність інтерпретації отриманих результатів. Особливо слід підкреслити, що тема має міждисциплінарний характер: у ній поєднуються методи комп'ютерних наук, аналіз даних і практична потреба конкретної сфери застосування. Це робить дослідження не формальним, а таким, що має потенціал подальшого розвитку. Оскільки робота пов'язана з аналізом телеметричних даних, особливої ваги набувають питання достовірності даних і коректності автоматизованих висновків. Автор демонструє розуміння того, що технічне рішення має бути не лише працездатним, а й обґрунтованим з погляду практичного використання.

2. Повнота розкриття мети та завдань роботи

Завдання роботи в цілому виконані. Особливо варто відзначити, що автор не обмежується загальним описом теми, а доводить дослідження до програмного й експериментального рівня. Повнота розкриття проявляється у переході від аналізу джерел до формалізації методу та оцінювання результатів. Така логіка дозволяє побачити, що виконання роботи не обмежувалося компіляцією відомих положень, а передбачало власне опрацювання матеріалу. У роботі достатньо простежується зв'язок між сформульованими завданнями та отриманими результатами. Автор не розриває теоретичну й практичну частини, а показує, як аналіз предметної області переходить у вибір методу та його реалізацію.

3. Зміст кожного розділу роботи

У роботі послідовно розглянуто предметну область, модельну частину і програмну реалізацію. Така побудова є зручною для оцінювання, оскільки видно, як вихідна проблема переходить у конкретне технічне рішення. Зміст розділів подано достатньо розгорнуто для бакалаврського рівня. Позитивним є те, що робота містить не тільки загальний огляд предметної області, а й пояснення особливостей даних, етапів обробки та критеріїв оцінювання. Опис розділів дозволяє зрозуміти, які саме дані використовувалися, які етапи обробки були необхідними та яким чином оцінювалася ефективність рішення. Для бакалаврської роботи це важливо, оскільки демонструє не тільки результат, а й методичну послідовність його отримання.

4. Оцінка розробленої інформаційної системи, її практична цінність

Практична цінність полягає в можливості використання результатів для персоналізованого тренування водіїв і підвищення якості підготовки. Розробка не виглядає відірваною від реального застосування: вона орієнтована на обробку конкретних даних і

отримання інтерпретованого результату. Розроблена система має прикладний характер, оскільки орієнтована на обробку конкретних типів даних і отримання результату, придатного для інтерпретації. Це важливо, бо практична цінність інформаційної системи визначається не тільки фактом реалізації, а й можливістю використання результатів у реальних сценаріях. Практична значущість роботи полягає також у можливості використання її результатів як основи для подальшого експериментального розширення. Запропонований підхід може бути доповнений новими даними, уточненими критеріями оцінювання або інтегрований у ширшу програмну систему.

5. Якість оформлення кваліфікаційної роботи бакалавра

Якість оформлення є належною. В окремих місцях текст міг би бути компактнішим, однак це не порушує загальної логіки подання матеріалу. Матеріал подано достатньо охайно й послідовно, без різких переходів між теоретичними та практичними фрагментами. Текст роботи загалом читається як цілісний академічний матеріал, а не як набір розрізнених технічних описів.

6. Недоліки кваліфікаційної роботи бакалавра

У роботі не завжди однаково детально описано обмеження запропонованого методу. Проте зазначене не знижує значущості отриманих результатів. Крім того, робота виграла б від ширшого аналізу випадків, у яких запропонований метод може давати менш точні або неоднозначні результати. Разом із тим зазначені недоліки не є визначальними, оскільки основна мета роботи досягнута, а запропонований метод продемонстровано на практиці.

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота

З урахуванням актуальності теми, самостійності виконання та практичної спрямованості результатів робота допускається до захисту і заслуговує на оцінку « добре ».

Рецензент к.т.н., доц. Калустян М.В.

