Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Інтелектуальна система адаптивного управління логістикою на основі Edge AI в умовах інфраструктурних обмежень
(тема)

Виконав:
здобувач другого року навчання,
групи ІТШІ-22-1
Яна ГЕОРГІЄВА
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-професійна
Освітня програма Штучний інтелект
(повна назва освітньої програми)

Керівник ст. викл. Олена ГРИНЬОВА
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ _____ Лариса ЧАЛА
(підпис) (власне ім'я, прізвище)

2026 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Штучного інтелекту

Рівень вищої освіти перший (бакалаврський)

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

Тип програми освітньо-професійна

Освітня програма Штучний інтелект
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Георгієвій Яні Василівні
(прізвище, ім'я, по батькові)

1. Тема роботи Інтелектуальна система адаптивного управління логістикою на основі Edge AI в умовах інфраструктурних обмежень

затверджена наказом університету від 5 червня 20 26 р. № 488Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 9 липня 20 26 р.

3. Вихідні дані до роботи Нормативні вимоги Європейської угоди щодо роботи екіпажів транспортних засобів (ЄУТР), топологія базових транспортних магістралей України, телеметричні дані бортових систем автомобіля (CAN-шина), біометричні показники стану водія (метрика PERCLOS), а також показники інфраструктурних обмежень (комендантська година, стан дорожнього покриття).

4. Перелік питань, що потрібно опрацювати в роботі

1) Аналіз предметної галузі та постановка задачі

2) Проєктування системи

3) Програмна реалізація

4) Розробка користувацького інтерфейсу та візуалізація даних

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / термін виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	08.06.2026	виконано
2	Аналіз предметної галузі	08.06.2026	виконано
3	Огляд та порівняльний аналіз існуючих рішень	08.06.2026	виконано
4	Постановка задачі та проєктування системи	10.06.2026	виконано
5	Моделювання алгоритмів Edge AI	12.06.2026	виконано
6	Програмна реалізація модулів системи	12.06.2026	виконано
7	Розроблення інтерфейсу та тестування системи	20.06.2026	виконано
8	Написання пояснювальної записки	22.06.2026	виконано
9	Перевірка на академічний плагіат	25.06.2026	виконано
10	Нормоконтроль	24.06.2026	виконано
11	Підготовка презентації та доповіді	27.06.2026	виконано
12	Попередній захист, рецензування	05.07.2026	виконано
13	Захист	09.07.2026	

Дата видачі завдання 8 червня 2026 р.

Здобувач _____
(підпис)

Керівник роботи _____ ст. викл. Олена ГРИНЬОВА
(підпис) (посада, власне ім’я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 79 с., 12 рис., 2 табл., 7 дод., 22 джерела.

АДАПТИВНЕ УПРАВЛІННЯ, ЕДЖ-ОБЧИСЛЕННЯ, ІНТЕЛЕКТУАЛЬНА СИСТЕМА, КАН-ШИНА, ІНФРАСТРУКТУРА, ЛОГІСТИКА, МОНІТОРИНГ, НЕЙРОННІ МЕРЕЖІ, PERCLOS.

Об’єкт дослідження – процес управління логістичними операціями та режимами праці екіпажів вантажних транспортних засобів в умовах високої невизначеності та інфраструктурної нестабільності.

Предмет дослідження – методи та алгоритми периферійного штучного інтелекту, комп’ютерного зору та інтеграції телеметричних даних для адаптивного планування маршрутів і контролю психофізіологічного стану.

Мета роботи – розробка інтелектуальної адаптивної системи управління логістикою, яка забезпечує живучість логістичних ланцюгів та безпеку водіїв шляхом локальної обробки даних в умовах блекаутів, порушення зв’язку та руйнування дорожньої інфраструктури.

Методи дослідження – у роботі використано методи комп’ютерного зору та згорткові нейронні мережі для аналізу стану водія; методи злиття даних для крос-модальної валідації показників CAN-шини та відеопотоку; методи предиктивної аналітики та теорії графів для динамічного перерахунку маршрутів до безпечних локацій.

Через конфлікт норм ЄУТР із реаліями воєнного стану обґрунтовано необхідність інтелектуальних систем підтримки водія. Розроблено Offline-архітектуру на базі Edge AI з крос-модальною валідацією втоми (поєднання PERCLOS та телеметрії CAN-шини). Впроваджено модуль «Safe Hubs» для предиктивного планування безпечного відпочинку. Результати моделювання доводять підвищення стабільності логістичних процесів на 25–30% та зниження аварійності в умовах критичної нестабільності.

ABSTRACT

Bachelor`s thesis contains: 79 p., 12 fig., 2 tabl., 7 ann., 22 references.

ADAPTIVE MANAGEMENT, CAN-BUS, EDGE COMPUTING, INFRASTRUCTURE, INTELLIGENT SYSTEM, LOGISTICS, MONITORING, NEURAL NETWORKS, PERCLOS.

The object of the study is the process of managing logistics operations and labor regimes of cargo vehicle crews in conditions of high uncertainty and infrastructural instability.

The subject of the study is methods and algorithms of edge artificial intelligence (Edge AI), computer vision, and integration of telemetric data for adaptive route planning and monitoring of the driver's psychophysiological state.

The goal of the work is to develop an intelligent adaptive logistics management system that ensures the resilience of logistics chains and driver safety through local data processing in conditions of blackouts, communication disruptions, and destruction of road infrastructure.

Research methods – the work utilizes computer vision methods and convolutional neural networks for driver state analysis; data fusion methods for cross-modal validation of CAN-bus indicators and video stream; methods of predictive analytics and graph theory for recalculation of routes to safe hubs.

The study justifies the shift to intelligent driver support systems due to the conflict between AETR norms and martial law. An offline-capable Edge AI architecture is developed, featuring cross-modal fatigue validation (PERCLOS combined with CAN-bus telemetry) to minimize false positives. The «Safe Hubs» module enables predictive rest planning considering curfews, and connectivity.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів 8
Вступ 9
1 Аналіз предметної галузі та постановка задачі 11
1.1 Опис предметної галузі 11
1.2 Опис проблем та викликів 17
1.3 Огляд існуючих застосунків 21
1.4 Інноваційність та переваги розроблюваної системи 23
1.5 Постановка задачі 24
2 Проєктування системи 27
2.1 Загальна архітектура системи 27
2.2 Моделювання логістичного середовища та інфраструктурних обмежень 29
2.3 Проєктування Edge AI модуля локального збору та обробки даних . 34
2.4 Проєктування алгоритму адаптивного управління маршрутизацією 36
2.5 Проєктування підсистеми диспетчеризації та моніторингу 38
3 Програмна реалізація 43
3.1 Вибір засобів розробки та технологій 43
3.2 Проєктування логістичного графа транспортної мережі України 44
3.3 Реалізація емулятора Edge AI та генерації обмежень 46
3.4 Програмна реалізація алгоритму адаптивної маршрутизації 49
3.5 Тестування та візуалізація результатів 52
4 Розробка користувацького інтерфейсу та візуалізація даних 55
4.1 Проєктування архітектури взаємодії користувачів із системою 55
4.2 Розробка вебпанелі диспетчера логістичного центру 57
4.3 Розробка інтерфейсу бортового комп'ютера водія 60
Висновки 62
Перелік джерел посилання 64
Додаток А Модуль ініціалізації логістичного графа 67

Додаток Б Модуль емуляції інфраструктурного середовища та функціонування Edge AI..... 70
Додаток В Модуль алгоритму адаптивної маршрутизації..... 72
Додаток Г Головний модуль симуляції та тестування системи 74
Додаток Д Модуль хмарної панелі диспетчеризації 76
Додаток Е Модуль інтерфейсу бортового термінала водія 78
Додаток Ж Відомість кваліфікаційної роботи 79

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ЄУТР – Європейська угода щодо роботи екіпажів транспортних засобів, які здійснюють міжнародні автомобільні перевезення;

CAN – Controller Area Network – протокол передачі даних, що забезпечує зв’язок між мікроконтролерами та бортовими пристроями автомобіля без головного комп’ютера;

DAG – Directed Acyclic Graph – орієнтований ациклічний граф;

DFS – Depth-First Search – пошук у глибину;

DMS – Driver Monitoring System – інтелектуальна система моніторингу стану водія;

Edge AI – Edge Artificial Intelligence – реалізація моделей штучного інтелекту на периферійних (бортових) пристроях для локальної обробки даних без постійного зв’язку з хмарою;

OWL – Web Ontology Language – мова веб-онтологій;

PERCLOS – Percentage of Eyelid Closure – відсоток часу закриття повік, ключова біометрична метрика для детекції втоми водія;

ВСТУП

Сучасний сектор комерційних вантажних перевезень та логістики функціонує в умовах безпрецедентних викликів. Згідно з міжнародними нормами ЄУТР, робочий цикл водія жорстко регламентований і вимагає обов'язкових перерв для забезпечення безпеки дорожнього руху. Проте в реаліях воєнного стану в Україні ці стандарти вступають у прямий конфлікт із зовнішніми дестабілізуючими факторами: непередбачуваними затримками на блокпостах, руйнуванням транспортної інфраструктури, зміною маршрутів та жорсткими обмеженнями комендантської години [1].

Традиційні засоби контролю, такі як цифрові тахографи, є інструментами пасивного моніторингу: вони фіксують порушення режиму праці та відпочинку постфактум, але не надають водієві жодної допомоги для уникнення цих порушень.

Вирішення цієї проблеми вимагає переходу від пасивного фіксування даних до систем активної інтелектуальної допомоги водієві (Driver Monitoring System – DMS) з використанням технологій штучного інтелекту. Застосування концепції периферійних обчислень (Edge Computing / Edge AI) дозволяє перенести складні аналітичні процеси безпосередньо на бортовий комп'ютер транспортного засобу [2], [3]. З огляду на це, розробка інтелектуальної системи адаптивного управління логістикою на базі Edge AI є своєчасним, науково обґрунтованим та критично важливим завданням для підвищення живучості логістичних ланцюгів і безпеки водіїв.

Для досягнення поставленої мети необхідно вирішити такі основні завдання:

– проаналізувати предметну галузь, існуючі проблеми дотримання норм ЄУТР в умовах нестабільності та обмеження сучасних систем моніторингу автопарків [4], [5];

– розробити загальну архітектуру інтелектуальної системи адаптивного управління на базі Edge Computing, здатну функціонувати в Offline-режимі [4], [5];

– дослідити та впровадити модуль біометричного моніторингу втоми водія із застосуванням згорткових нейронних мереж [6], [7];

– розробити алгоритм крос-модальної валідації втоми, що поєднує дані комп'ютерного зору з телеметричними показниками автомобіля (CAN-шина) для мінімізації хибнопозитивних спрацювань;

– створити модуль динамічного планування «Safe Hubs» для предиктивного розрахунку залишку робочого часу та вибору безпечних місць відстою з урахуванням комендантської години та наявності автономного живлення;

– оцінити ефективність та переваги запропонованої системи.

Теоретичною та методологічною основою роботи є методи машинного навчання та комп’ютерного зору (зокрема, згорткові нейронні мережі – CNN) для аналізу біометричних параметрів водія. Для забезпечення достовірності оцінки стану водія використано методи злиття даних (Data Fusion) [8], що дозволяють інтегрувати відеодані з показниками бортової CAN-шини автомобіля [8], [9]. Побудова модуля безпечних зупинок («Safe Hubs») базується на методах предиктивної аналітики та динамічного маршрутизації.

Розроблена архітектура та алгоритми системи дозволяють трансформувати логістичні ланцюги зі статичної моделі в гнучку динамічну структуру. Запропонована система крос-модальної валідації суттєво підвищує точність виявлення втоми водія (мікросну), а інтеграція алгоритмів Edge AI гарантує безперебійний розрахунок маршруту в зонах відсутності зв'язку. Впровадження результатів роботи дозволить транспортним компаніям знизити аварійність та мінімізувати ризики порушення комендантського режиму екіпажами транспортних засобів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Опис предметної галузі

Безпека дорожнього руху в секторі комерційних перевезень безпосередньо залежить від суворого дотримання режиму праці та відпочинку екіпажів. У стабільних умовах логістична галузь базується на чіткому регламентуванні робочого часу, що дозволяє мінімізувати ризики аварійності та забезпечити ритмічність ланцюгів постачання. Основним нормативним документом у цій сфері є Європейська угода щодо роботи екіпажів транспортних засобів (ЄУТР), яка встановлює жорсткі часові рамки. Зокрема, стандартний робочий цикл складає 15 годин, з яких 9 годин відведено на керування, а кожні 4,5 години руху мають перериватися відпочинком в 45 хвилин.

Решта 6 годин циклу включають супутні виробничі завдання, такі як завантаження, розвантаження та митне оформлення. Загальна взаємодія компонентів інтелектуальної системи у штатному режимі, де всі вузли екосистеми працюють узгоджено, представлена на рисунку 1.1.

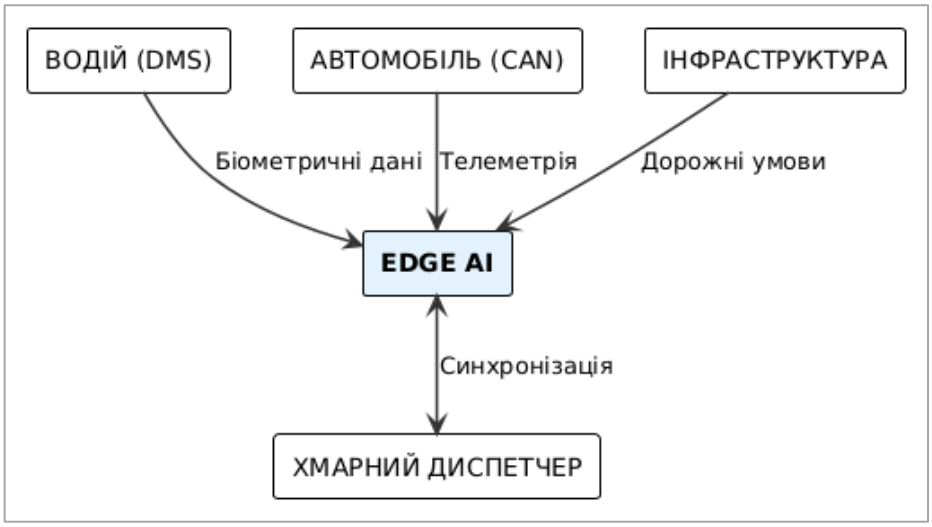


Рисунок 1.1 – Логістична екосистема на базі Edge AI у штатному режимі роботи

На рисунку відображено базову взаємодію всіх чотирьох вузлів системи. Дані з CAN-шини та модуля DMS стабільно надходять до центрального ядра Edge AI, яке паралельно підтримує двосторонню синхронізацію з хмарним диспетчерським центром. Система працює без відхилень, індикатори в нормі.

Проте на практиці, особливо під впливом динамічних інфраструктурних обмежень та непередбачуваних логістичних збоїв, дотримання еталонного графіка часто стає неможливим. Вимушені зміни маршруту, багатогодинні затримки в заторах або тривала відсутність зв'язку неминуче призводять до накопичення психофізичної втоми водія та критичного зниження його концентрації. Традиційні апаратні засоби контролю, такі як цифрові тахографи, забезпечують лише ретроспективну фіксацію порушень режиму праці постфактум, залишаючись абсолютно неефективними для превентивного реагування на небезпечний стан людини в режимі реального часу.

Саме тому виникає гостра потреба у трансформації пасивного контролю в активну інтелектуальну допомогу шляхом впровадження систем DMS (Driver Monitoring System). Така зміна парадигми дозволяє перетворити статичні логістичні ланцюги на гнучкі адаптивні системи, здатні ефективно функціонувати в умовах інфраструктурних обмежень та забезпечувати високий рівень безпеки водіїв у ситуаціях з високим рівнем ризику.

Проте в умовах воєнного стану та критичної нестабільності інфраструктури ці норми вступають у прямий конфлікт із зовнішніми обмеженнями. Стандартні цифрові тахографи в таких умовах лише фіксують порушення режиму, не надаючи інструментів для їхнього запобігання. Сучасна логістична ситуація в Україні характеризується поєднанням руйнування дорожньої мережі, неочікуваних затримок на блокпостах та дії комендантської години, яка вимагає повної зупинки руху незалежно від вичерпання водієм встановлених лімітів. Крім того,

енергетична нестабільність та ризики блекаутів унеможливлюють постійну координацію з диспетчерськими центрами через традиційні хмарні сервіси, що потребує локальної автономності систем управління. Процес переходу до периферійних обчислень та робота алгоритмів у режимі Offline при втраті каналу зв'язку з диспетчерським центром проілюстровано на рисунку 1.2.

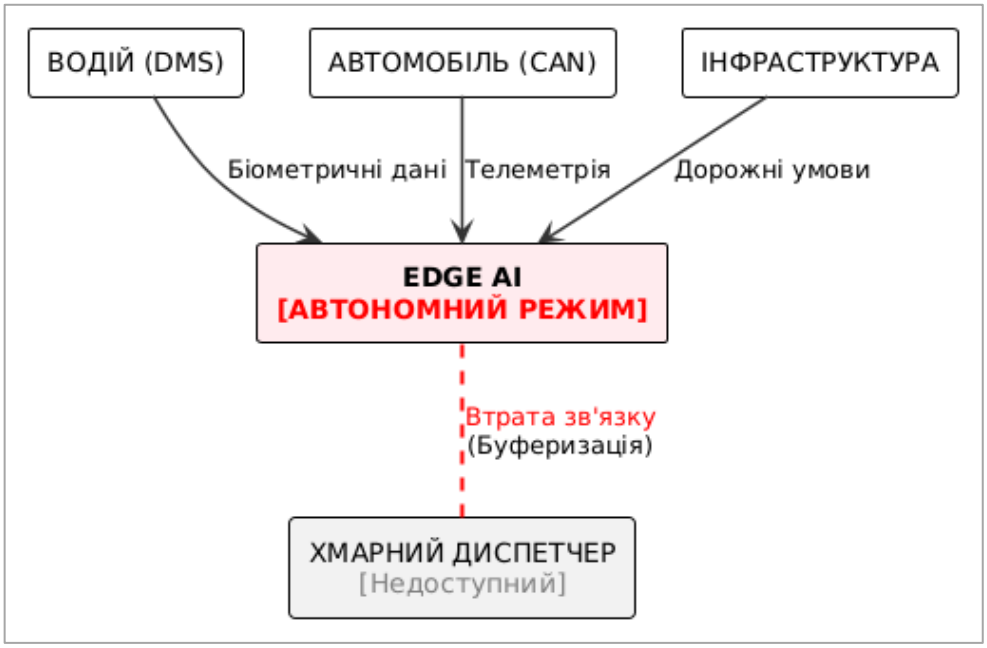


Рисунок 1.2 – Реакція системи на втрату зв'язку (перехід в автономний режим обчислень)

Зображено імітацію потрапляння транспортного засобу в зону дії засобів РЕБ або блекауту. Зв'язок із хмарним сервером обривається (сірий блок/пунктир), проте система не зупиняє свою роботу. Модуль Edge AI бере на себе повний контроль за маршрутом, здійснюючи предиктивні обчислення локально на бортовому комп'ютері.

У цьому автономному режимі (Edge-only) бортовий інтелектуальний модуль перебирає на себе повну відповідальність за прийняття оперативних рішень. Замість очікування команд від відключеного хмарного сервера, локальне обчислювальне ядро безперервно аналізує поточний психофізичний стан водія через систему DMS та зіставляє його зі змінами

дорожньої обстановки. У разі наближення комендантської години або виявлення раптового перекриття магістралі, підсистема миттєво ініціює перерахунок альтернативного маршруту до найближчого безпечного пункту зупинки, спираючись виключно на попередньо завантажену бортову карту графа доріг. При цьому уся зібрана телеметрія не втрачається: дані надійно буферизуються в локальній пам'яті транспортного засобу. Щойно енергетична стабільність або покриття мережі відновлюється, накопичений масив інформації автоматично синхронізується з хмарним диспетчером, забезпечуючи безперервність логістичного контролю навіть після тривалих інфраструктурних збоїв.

Трансформація пасивного контролю в активну інтелектуальну допомогу потребує впровадження систем Driver Monitoring System (DMS). Це передбачає використання інфрачервоних камер та штучного інтелекту для аналізу стану водія в реальному часі, що дозволяє детектувати втому на етапі її виникнення. Однак покладання виключно на візуальний аналіз (наприклад, розпізнавання метрики PERCLOS – відсотка часу закриття повік) є вразливим до хибних спрацювань. Складні умови освітлення в кабіні, використання водієм сонцезахисних окулярів, перекриття обличчя сторонніми предметами або різкі повороти голови можуть бути помилково інтерпретовані системою як втрата свідомості.

Саме тому предметна галузь сьогодні стрімко зміщується в бік комплексних адаптивних систем, які застосовують підхід «Data Fusion» – глибоку інтеграцію даних біометрії з технічною телеметрією автомобіля. Локальне обчислювальне ядро Edge AI не приймає критичних рішень ізольовано. Якщо камера фіксує ознаки засинання, алгоритм миттєво звертається до CAN-шини транспортного засобу для підтвердження. Наявність монотонного руху без характерних мікрокоригувань керма або сталий тиск на педаль акселератора стають вирішальними факторами. Такий механізм крос-модальної валідації стану екіпажу дозволяє достовірно відрізнити реальну загрозу (мікросон) від похибок візуального аналізу,

зводячи до мінімуму кількість хибних тривог та необґрунтованих зупинок транспорту. Блок-схему алгоритму цієї багаторівневої перевірки наведено на рисунку 1.3.

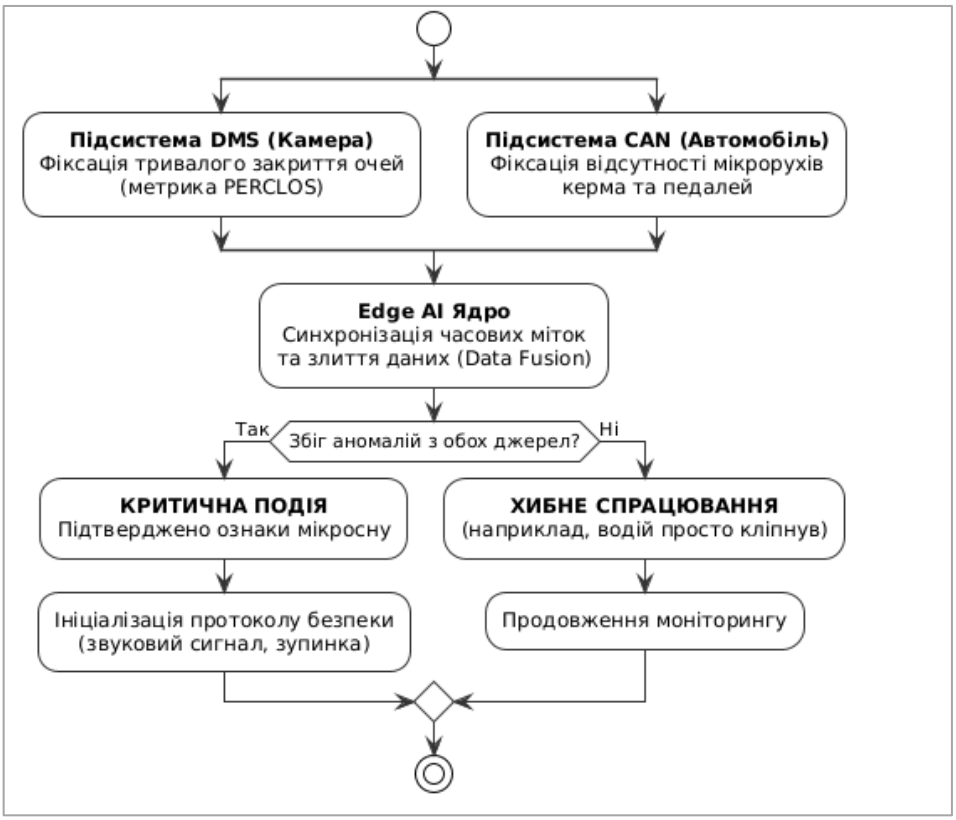


Рисунок 1.3 – Алгоритм крос-модальної валідації при детекції ознак мікросну водія

Наочна демонстрація інтелектуальної обробки інциденту. Блок водія сигналізує про небезпеку. Замість негайної хибної тривоги, ядро Edge AI звертається до автомобіля (CAN-шини) для перевірки мікроколивань керма. Отримавши підтвердження, система фіксує математично обґрунтований факт втоми та ініціює алгоритм екстреної зупинки.

Проте локалізація інциденту є лише першим етапом роботи алгоритму. Після фіксації критичного стану водія система не обмежується банальним блокуванням подальшого руху, а миттєво переходить від реактивної до проактивної фази управління. В умовах воєнного стану та

інфраструктурної нестабільності зупинка багатотоннажного транспортного засобу на випадковій чи неосвітленій ділянці магістралі може становити ще більшу загрозу для життя екіпажу та цілісності вантажу. Тому алгоритм інтелектуального управління здійснює предиктивний аналіз топології мережі для визначення найближчої локації, яка відповідає критеріям безпеки та автономності.

Такий підхід дозволяє не просто відстежувати час роботи, а й будувати предиктивні моделі руху, пропонуючи водієві оптимальні зони для відпочинку – так звані «Safe Hubs», що забезпечені автономним живленням, укриттям та резервним зв'язком. Таким чином, сучасне управління логістикою в умовах обмежень потребує високого рівня інтелектуалізації та стійкості обчислювальних процесів. Завдяки заздалегідь завантаженій топології мережі, бортовий комп'ютер автономно вираховує безпечне часове вікно, необхідне для гарантованого прибуття до найближчого захищеного пункту. Предиктивне планування маршруту до безпечної локації з урахуванням наближення комендантської години та зовнішніх інфраструктурних ризиків відображено на рисунку 1.4.

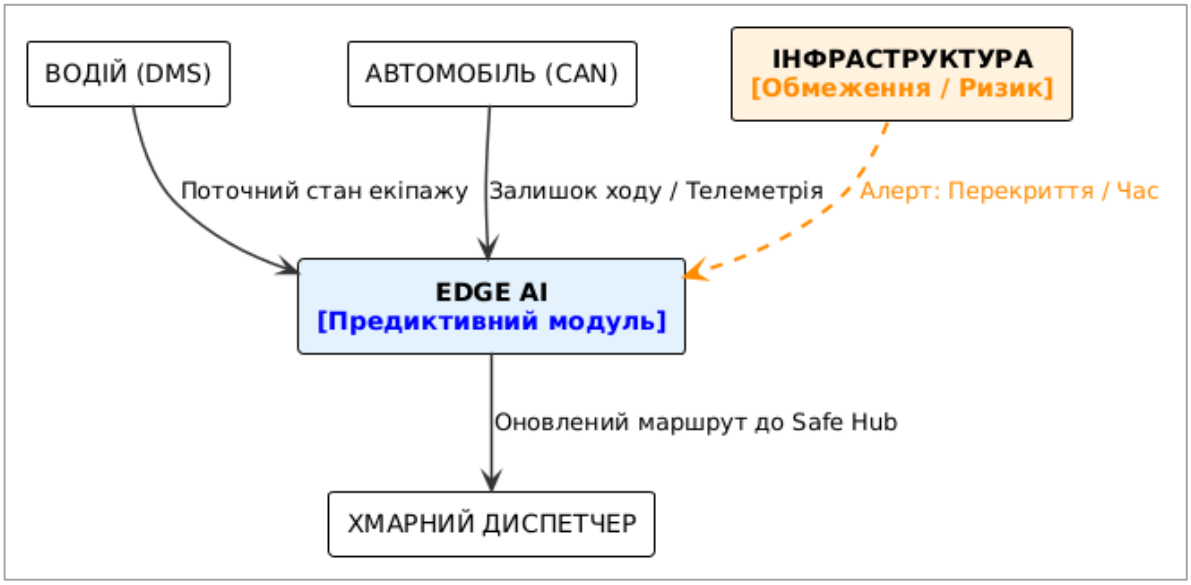


Рисунок 1.4 – Предиктивне планування маршруту до безпечного хабу в умовах інфраструктурних обмежень

Візуалізація роботи гібридного алгоритму маршрутизації. Отримавши інформацію про інфраструктурне обмеження (комендантська година), система предиктивно перераховує залишок робочого часу (T_{rem}) та прокладає новий маршрут до найближчого «Safe Hub» – локації, що гарантовано має автономне живлення та зв'язок для безпечної ночівлі екіпажу.

1.2 Опис проблем та викликів

Сучасна система управління комерційними перевезеннями в умовах інфраструктурних обмежень стикається з низкою критичних викликів, які можна класифікувати як операційні, технологічні та психофізіологічні. Головною проблемою є неможливість забезпечення безпеки та дотримання нормативів ЄУТР за допомогою застарілих методів пасивного контролю.

Конфлікт нормативів та інфраструктурна нестабільність. Основним операційним викликом є пряма невідповідність жорстких часових лімітів ЄУТР реальному стану дорожньої інфраструктури. В умовах воєнного стану водії змушені витрачати значну частину ліміту робочої зміни (T_{shift}) на незаплановані зупинки: проходження блокпостів, об’їзди зруйнованих ділянок доріг та мостів, а також очікування завершення повітряних тривог.

Це призводить до того, що розраховане логістичне вікно закривається раніше, ніж вантаж досягає цілі, а водій опиняється перед вибором: порушити режим праці або залишитися в небезпечній зоні через початок комендантської години.

Додатково варто підкреслити проблему «інформаційної асиметрії» в ланцюгу «водій-диспетчер». Класична модель моніторингу спирається на передачу великих масивів необроблених даних (raw data) з борту автомобіля на сервер для подальшого аналізу. В умовах воєнних дій, коли пропускна здатність каналів зв’язку є критично низькою, а стабільність з’єднання вкрай нестабільною, така архітектура стає «вузьким місцем». Спроба

передати відеопотік з камери DMS або дані високої частоти з CAN-шини в хмару призводить до деградації сервісу: диспетчер отримує запізнілі дані, які втратили свою актуальність, тоді як водій залишається позбавленим будь-якої допомоги, оскільки хмарні алгоритми не можуть обробити ситуацію в режимі реального часу. Це створює «ілюзію контролю» у диспетчера та «відчуття покинутості» у водія, що є вкрай небезпечним поєднанням.

Технологічні бар'єри та енергетична дефіцитність. Сучасні системи моніторингу автопарків мають високу залежність від стабільного хмарного зв'язку та супутникової навігації. Проте робота в умовах активного застосування засобів радіоелектронної боротьби (РЕБ) призводить до втрати або викривлення GPS-сигналів. Крім того, масові блекаути роблять неможливою синхронізацію з центральними серверами, перетворюючи стандартні навігаційні системи на нефункціональні пристрої. Це створює інформаційний вакуум, у якому водій залишається без підтримки та актуальних даних про безпечні місця для відпочинку.

Психофізіологічний фактор та обмеження контролю. Найбільш критичним викликом є людський фактор. У стресових умовах водій часто намагається «перемогти» втому силою волі, щоб встигнути до комендантської години або виїхати з небезпечної зони. Це проявляється в «стресовому» профілі водіння: різких прискореннях, гальмуваннях та ігноруванні перших ознак виснаження. Стандартні цифрові тахографи не здатні детектувати стан мікросну або зниження концентрації – вони лише фіксують час, коли двигун був увімкнений.

Окремої уваги потребує проблема довіри до автоматизованих систем. Існуючі системи DMS, які базуються виключно на візуальних датчиках, часто страждають від надмірної кількості хибнопозитивних спрацювань. Коли водій, який керує багатотонним транспортним засобом у складних умовах, отримує п'ятий за день сигнал про «сонливість», спричинений лише тим, що він на секунду відвів погляд на дзеркало чи приладову панель,

виникає ефект «синдрому пастуха» (Cry Wolf effect). Система перестає сприйматися як засіб безпеки і стає дратівливим фактором, який водій починає свідомо ігнорувати або навіть намагається вимкнути. Така поведінка користувача нівелює будь-які інвестиції в технології безпеки. Тому головним викликом є не просто детекція втоми, а створення системи, яка «розуміє» контекст водіння, відрізняючи технічну помилку камери від реальної втрати контролю через інтеграцію з динамічними даними автомобіля.

Недоліки існуючих інструментів моніторингу. Використання виключно візуальних методів контролю втоми часто супроводжується великою кількістю хибнопозитивних спрацювань, що дратує водія та змушує його ігнорувати попередження системи. Відсутність інтеграції біометричних даних із телеметрією автомобіля (CAN-шиною) не дозволяє отримати математично обґрунтоване підтвердження критичного стану. Таким чином, виникає гостра потреба у створенні системи, яка здатна працювати автономно в Offline-режимі, адаптуватися до ризиків інфраструктури та забезпечувати високу точність детекції втоми через крос-модальну валідацію даних.

Крім того, варто виділити правовий конфлікт: сучасні норми ЄУТР розроблялися в парадигмі мирного часу, де логістичний ланцюг є передбачуваним, а інфраструктура – надійною. У реаліях України водій стає заручником «ножиць» між законом і здоровим глуздом. Дотримання режиму праці (наприклад, зупинка через чотири з половиною години) у небезпечній, неосвітленій зоні поблизу лінії фронту або в зоні підвищеного ризику обстрілів є юридично правильним, але фактично небезпечним рішенням. Існуючі системи не пропонують «інтелектуального виходу» з цієї ситуації. Водій залишається сам на сам із вибором: отримати штраф за перевищення ліміту чи підставити екіпаж під пряму загрозу. Система моніторингу повинна не просто фіксувати факт порушення, а бути

спроможною автоматично аргументувати вимушені відхилення від графіка, інтегруючи дані про стан безпеки навколишнього середовища.

Таким чином, стає очевидним, що сучасна логістична система досягла межі своєї ефективності в рамках класичної архітектури «пристрій-хмара». Висока латентність, залежність від зовнішніх каналів зв'язку та обмежена здатність до аналізу контексту роблять існуючі рішення вразливими до непередбачуваних зовнішніх факторів. Будь-яка деградація зв'язку в такій моделі призводить до повної зупинки процесу управління, залишаючи транспортний засіб поза межами інтелектуального контуру компанії. Саме тому перехід до парадигми розподілених периферійних обчислень (Edge Computing), де обробка критично важливих даних здійснюється безпосередньо на борту транспортного засобу, є не просто технологічним трендом, а необхідною умовою виживання бізнесу в екстремальних умовах.

Ми маємо «розумні» тахографи, «розумні» камери та «розумні» системи навігації, але жодна з них не володіє цілісним розумінням контексту ситуації. Ці системи не спілкуються між собою, не враховують стан водія при плануванні маршруту і, що найважливіше, не здатні приймати самостійні рішення в автономному режимі. Виникає потреба в архітектурі, яка б не просто «передавала дані нагору», а формувала автономний «інтелектуальний вузол» (Edge Node) здатний мислити, приймати рішення та гарантувати безпеку навіть тоді, коли зовнішній світ (хмарна інфраструктура) стає недоступним. Саме цей «розрив» між вимогами реальності та можливостями класичних систем є тією проблемою, яку покликана вирішити наша робота.

Тільки інтеграція інтелектуальних алгоритмів у локальний вузол обробки здатна забезпечити необхідну автономність, стійкість до засобів РЕБ та миттєву реакцію на зміну психофізіологічного стану водія, що є ключовим для мінімізації аварійності в зонах підвищеного ризику. Це рішення забезпечує автономність критичних логістичних процесів, трансформуючи кожен транспортний засіб у самодостатній

інтелектуальний вузол управління, здатний ефективно функціонувати навіть у найбільш несприятливих умовах.

1.3 Огляд існуючих застосунків

На сьогоднішній день ринок систем управління комерційним транспортом представлений широким спектром рішень, які можна розділити на три основні категорії: системи управління автопарком (FMS), засоби тахографічного контролю та спеціалізовані системи моніторингу стану водія (DMS).

Системи управління автопарком (Fleet Management Systems). Найбільш поширеними є комплексні платформи від таких виробників, як Trimble, Verizon Connect та Garmin. Ці системи забезпечують GPS-трекінг у реальному часі, контроль витрат пального та автоматизоване планування маршрутів. Проте головним недоліком цих рішень є їхня повна залежність від «хмарної» архітектури. Усі обчислення та оптимізація маршрутів відбуваються на віддалених серверах, що робить систему недієздатною в зонах відсутності мобільного інтернету або при роботі засобів РЕБ. Крім того, вони не мають вбудованих інструментів для глибокого аналізу втоми водія, обмежуючись лише фіксацією часу руху.

Засоби тахографічного контролю. Цифрові тахографи таких брендів, як Stoneridge або Continental (VDO), є обов'язковими для використання в країнах ЄС та Україні. Сучасні моделі підтримують дистанційне зчитування даних та інтеграцію з ГНСС. Однак, як зазначалося раніше, тахограф – це інструмент ретроспективного аналізу. Він фіксує факт порушення ЄУТР, але не здатен попередити його в умовах динамічних затримок на дорозі. Тахографи не враховують інфраструктурні обмеження (наприклад, наявність укриттів чи комендантську годину), що вимагає від водія самостійного та часто помилкового планування зупинок.

Системи моніторингу водія (DMS). Інтелектуальні системи, такі як Mobileye (Intel) або Seeing Machines, використовують алгоритми комп'ютерного зору для відстеження погляду та пози водія. Наприклад, рішення Mobileye Shield+ ефективно детектує небезпечні зближення та ознаки неуважності. Проте ці системи часто працюють як ізольовані пристрої безпеки. Вони не інтегровані в логістичний ланцюг: система може сигналізувати про втому, але вона не знає, де знаходиться найближчий безпечний хаб із генератором в умовах блекауту, і не може перевірити через CAN-шину, чи не є закриття очей водія просто технічною помилкою камери при певному освітленні.

Для порівняльного аналізу існуючих рішень та обґрунтування необхідності розробки адаптивної системи на базі Edge AI у таблиці 1.1 наведено зіставлення ключових характеристик.

Таблиця 1.1 – Порівняльний аналіз систем управління логістикою та безпекою

Характеристика	Традиційні FMS	Цифрові тахографи	Сучасні DMS	Пропонована система
Тип обчислень	Хмарні (Cloud)	Локальні (Пасивні)	Edge (Частково)	Edge (Повний цикл AI)
Автономність (Offline)	Відсутня	Висока	Середня	Висока (Повна)
Детекція втоми	Відсутня	Лише за часом	Візуальна	Крос-модальна (DMS+CAN)
Адаптація до інфраструктури	Статична	Відсутня	Відсутня	Динамічна (Safe Hubs)
Валідація помилок	Н/Д	Н/Д	Низька	Висока (Data Fusion)

Таким чином, огляд існуючих застосунків підтверджує наявність технологічного розриву: існуючі системи або занадто залежні від інфраструктури (зв'язку), або занадто вузькоспеціалізовані. Розробка інтелектуальної системи, яка об'єднує локальну обробку відеоданих,

телеметрію автомобіля та адаптивну маршрутизацію в умовах обмежень, є критично необхідною для вирішення сучасних викликів логістики.

1.4 Інноваційність та переваги розроблюваної системи

В основі розроблюваної інтелектуальної системи лежить зміна парадигми управління логістичними процесами: перехід від реактивного фіксування подій до проактивного предиктивного управління з високим рівнем автономності. Унікальність запропонованого рішення формується за рахунок синергії технологій периферійних обчислень, крос-модальної валідації біометричних даних та адаптивної маршрутизації в умовах невизначеності.

Головною інновацією архітектурного рівня є забезпечення повної автономності обчислень на базі Edge AI. На відміну від існуючих систем управління автопарком, які критично залежать від хмарних серверів, розроблювана система переносить усі складні аналітичні процеси безпосередньо на бортовий комп’ютер транспортного засобу. Це дозволяє безперервно аналізувати відеопотік з камер та перераховувати маршрути локально, забезпечуючи нульову затримку прийняття рішень та гарантуючи повну працездатність системи в Offline-режимі під час дії засобів радіоелектронної боротьби або масштабних блекаутів.

Важливою науковою новизною є розробка та впровадження алгоритму крос-модальної валідації психофізіологічного стану водія. Традиційні системи моніторингу часто генерують хибнопозитивні попередження, оскільки спираються виключно на візуальні параметри. Розроблена система вирішує цю проблему за допомогою методів злиття даних (Data Fusion), де візуальна фіксація втоми за біометричною метрикою PERCLOS в обов'язковому порядку перевіряється через об'єктивні телеметричні дані з CAN-шини автомобіля. При фіксації ознак мікросну система миттєво аналізує мікрорухи керма та стабільність натискання

педалей, що перетворює візуальну ймовірність на математично доведений факт.

Інноваційність на рівні логістичного планування реалізована через предиктивну адаптацію в модулі «Safe Hubs». В умовах інфраструктурних обмежень, таких як комендантська година, затримки на блокпостах або раптові руйнування доріг, класичні навігатори втрачають свою ефективність. Натомість запропонована система динамічно розраховує залишок робочого часу водія з урахуванням цих стохастичних затримок та здатна предиктивно перенаправити транспортний засіб не просто на найближчу стоянку, а до безпечної локації з підтвердженим автономним живленням та стабільним зв'язком.

У підсумку, практичні переваги розроблюваної системи охоплюють значне підвищення живучості логістичних ланцюгів та забезпечення безпеки екіпажів у кризових умовах. Впровадження інтелектуальної допомоги знижує ризик ДТП, мінімізує ймовірність штрафів за порушення режимів ЄУТР і захищає від небезпек пересування під час комендантського часу. Крім того, використання автономних алгоритмів суттєво знижує когнітивне навантаження на водія, перетворюючи його з об'єкта жорсткого контролю на повноцінного партнера сучасної кіберфізичної системи управління.

1.5 Постановка задачі

Головною задачею даної кваліфікаційної роботи є теоретичне обґрунтування, проектування та практична розробка інтелектуальної системи адаптивного управління логістичними процесами, яка здатна функціонувати на базі технологій периферійних обчислень в умовах критичної інфраструктурної нестабільності. Розв'язання цієї проблеми вимагає створення комплексного програмного рішення, що переводить процес контролю за дотриманням норм ЄУТР із пасивного стану в активний

предиктивний режим із забезпеченням повної автономності роботи алгоритмів при втраті зв'язку з центральними диспетчерськими серверами.

Для досягнення поставленої мети необхідно послідовно вирішити низку науково-практичних завдань. Першим кроком є розробка загальної архітектури інтелектуальної екосистеми, яка інтегрує бортові обчислювальні потужності, підсистеми збору біометричних даних та телеметрію транспортного засобу. Ця архітектура має базуватися на принципах Edge AI, що гарантуватиме безперебійну локальну обробку масивів відеоданих та сенсорних показників без необхідності постійної синхронізації з хмарними сховищами, що є критичним фактором стабільності в умовах дії засобів радіоелектронної боротьби та енергетичних криз.

Наступним важливим етапом є алгоритмізація та програмна реалізація модуля моніторингу психофізіологічного стану водія. У межах цього завдання потрібно адаптувати алгоритми комп'ютерного зору для безперервного аналізу обличчя та мікрорухів, обравши як ключову біометричну метрику показник закриття повік PERCLOS. З метою усунення проблеми хибнопозитивних спрацювань необхідно синтезувати математичну модель крос-модальної валідації, що в реальному часі поєднуватиме результати візуального моніторингу з даними про динаміку керування автомобілем, отриманими безпосередньо через бортову CAN-шину.

Окремим завданням виступає створення модуля предиктивного планування маршрутів та управління логістичною безпекою. Необхідно розробити алгоритм, який здатний динамічно перераховувати залишок дозволеного робочого часу екіпажу з урахуванням стохастичних інфраструктурних затримок, таких як тривале проходження блокпостів або вимушені об'їзди. Цей алгоритм має автоматично шукати та формувати оптимальний маршрут до найближчого безпечного хабу зі стабільним

живленням до моменту настання комендантської години або критичного виснаження водія.

Заключним завданням є інтеграція всіх розроблених теоретичних та алгоритмічних модулів у єдину логічну структуру та проведення оцінки ефективності системи. У ході аналізу результатів необхідно підтвердити спроможність запропонованих рішень підвищувати живучість логістичних ланцюгів та знижувати аварійність порівняно з традиційними методами пасивного тахографічного контролю. Виконання зазначених завдань дозволить створити цілісний та дієвий інструмент забезпечення безпеки комерційних перевезень.

Реалізація окресленого комплексу завдань має не лише прикладне значення для галузі вантажних перевезень, а й формує науково-методичне підґрунтя для подальшого розвитку систем автономного управління логістичними ланцюгами. Розроблена інтелектуальна система, завдяки використанню методів Edge AI, здатна мінімізувати вплив «людського фактора» та інфраструктурних збоїв на безпеку руху, що створює передумови для сталого функціонування транспортної галузі в умовах екстремальних викликів. Результати дослідження та впроваджені алгоритмічні рішення можуть бути інтегровані у промислові системи управління автопарками, забезпечуючи перехід до проактивних методів контролю та підвищення надійності логістичних операцій у довгостроковій перспективі.

2 ПРОЄКТУВАННЯ СИСТЕМИ

2.1 Загальна архітектура системи

Специфіка управління логістичними процесами в умовах інфраструктурних обмежень (нестабільне мережеве покриття, обмежена пропускна здатність каналів зв'язку, тимчасова недоступність центральних серверів) вимагає переходу від традиційних хмарних архітектур до децентралізованих рішень. Саме тому базою для проєктування інтелектуальної системи адаптивного управління обрано парадигму граничних обчислень (Edge Computing) у поєднанні з методами штучного інтелекту (Edge AI).

Загальна архітектура розроблюваної системи будується за трирівневим принципом і включає: рівень кінцевих пристроїв (End-Device Layer), рівень граничних обчислень (Edge Layer) та хмарний рівень (Cloud Layer). Така топологія дозволяє перенести основне обчислювальне навантаження та прийняття критичних рішень безпосередньо до джерел даних (транспортних засобів, складських терміналів), мінімізуючи залежність від якості інтернет-з'єднання.

Рівень кінцевих пристроїв (End-Device Layer). Цей рівень складається з масиву IoT-сенсорів, GPS-трекерів, телематичних систем транспортних засобів та мобільних терміналів водіїв або складського персоналу. Основне завдання цього рівня – безперервний збір первинних даних про місцеперебування, стан транспортного засобу, витрати пального, а також фіксація зовнішніх умов (наприклад, стан дорожнього покриття чи затори). Дані з цього рівня передаються на граничні вузли через локальні протоколи зв'язку (Bluetooth, Wi-Fi, LoRaWAN) або безпосередньо через дротові інтерфейси.

Рівень граничних обчислень (Edge Layer). Є ядром пропонованої системи. Граничні вузли (Edge Nodes) встановлюються безпосередньо на

борту транспортних засобів або на локальних логістичних хабах. На цьому рівні розгортаються полегшені моделі машинного навчання (Edge AI), які здатні функціонувати автономно.

Модуль виконує такі функції:

– попередня обробка, фільтрація та агрегація сирих даних із сенсорів;

– локальне прогнозування та виявлення аномалій (наприклад, ризик поломки або раптове перекриття маршруту);

– прийняття локальних рішень щодо зміни маршруту в режимі реального часу, коли зв'язок із центральним сервером відсутній.

Хмарний рівень (Cloud Layer) функціонує як стратегічний центр управління, відповідальний за глобальну координацію, архівування ретроспективних даних та виконання найбільш ресурсомістких обчислень, зокрема донавчання (fine-tuning) нейронних моделей. На цей рівень дані надходять виключно в агрегованому та стисненому форматі, що дозволяє оптимізувати пропускну здатність каналів зв'язку та забезпечувати високу швидкість прийняття рішень на бортовому рівні.

Взаємодія між рівнями Edge та Cloud ґрунтується на принципах асинхронного обміну даними, що гарантує високу відмовостійкість всієї логістичної екосистеми. У разі виникнення інфраструктурних обмежень та втрати зв'язку, граничний вузол продовжує роботу в автономному режимі, спираючись на локальні моделі та кешовані маршрутні дані, одночасно формуючи деталізовані звіти про телеметрію. Така архітектура дозволяє масштабувати систему до рівня сотень транспортних засобів, мінімізуючи навантаження на мережеву інфраструктуру без втрати цілісності потоків інформації. У результаті, навіть під час тривалої ізоляції окремих сегментів, екосистема зберігає повну функціональну спроможність, гарантуючи, що жоден критичний інцидент не залишиться поза увагою після автоматичного відновлення синхронізації.

2.2 Моделювання логістичного середовища та інфраструктурних обмежень

Для ефективного функціонування інтелектуальної системи адаптивного управління необхідно виконати математичну формалізацію логістичного середовища та кількісно описати інфраструктурні обмеження, в яких розгортаються й оперують граничні вузли (Edge Nodes).

Для забезпечення ефективної алгоритмізації процесів управління в умовах динамічної інфраструктури необхідно формалізувати територіально-розподілену систему як математичну модель. Запропонований підхід передбачає абстрагування від фізичних параметрів транспортних потоків до топологічної структури, що дозволяє виконувати обчислення безпосередньо на периферійних вузлах (Edge AI) з мінімальними затримками. Логістична інфраструктура моделюється у вигляді зваженого динамічного орієнтованого графа (2.1).

$$G = (V, E), \tag{2.1}$$

де $V = v_1, v_2, \dots v_n$ – множина вершин, що представляють логістичні об'єкти (хаби, склади, пункти розподілу, митні термінали та кінцеві точки доставки вантажу);

$E = \{e_{ij}\}$ – множина спрямованих ребер, що описують наявні транспортні маршрути та сполучення між вузлами v_i та v_j.

На відміну від класичних статичних задач маршрутизації, в умовах децентралізованого управління із застосуванням Edge AI вага кожного ребра не є константою. Вона моделюється як функція від часу та поточного стану транспортної інфраструктури. Вага ребра $W_{ij}\ (t)$визначає узагальнену вартість (метрику витрат часу, пального та супутніх ризиків) на подолання відповідної ділянки маршруту в момент часу t.

Для формалізації цієї залежності ми застосовуємо підхід багатокритеріальної оптимізації, де граничний вузол (Edge node) на основі локальних сенсорів та телеметрії динамічно оновлює параметри графа. Вага ребра $w(e,t)$ визначає узагальнену вартість (метрику витрат часу, пального та супутніх ризиків) на подолання відповідної ділянки маршруту в момент часу t (2.2).

$$W_{ij}(t) = \alpha \cdot d_{ij} + \beta \cdot T_{ij}(t) + \gamma \cdot P_{ij}(t), \quad (2.2)$$

де d_{ij} – фізична відстань між логістичними вузлами;

$T_{ij}(t)$ – очікуваний час у дорозі, що є динамічною величиною і залежить від завантаженості доріг, черг на терміналах та погодних умов у момент часу t;

$P_{ij}(t)$ – штрафна функція ризику, яка відображає ймовірність виникнення критичних інфраструктурних перешкод (руйнування чи пошкодження дорожнього покриття, блокування шляхів, обмеження вантажопідйомності);

α, β, γ – нормовані вагові коефіцієнти значущості параметрів (α+β+γ=1), які адаптивно налаштовуються нейромережевим модулем залежно від пріоритету конкретного рейсу та типу вантажу.

Математичний опис інфраструктурних обмежень. Ключовою особливістю розроблюваної системи є її здатність зберігати життєздатність та приймати коректні рішення в умовах часткової або повної деградації каналів зв'язку. Для цього вводиться вектор стану інфраструктурних обмежень граничного вузла t (2.3).

$$S(t) = \{B(t), L(t), C(t)\}, \quad (2.3)$$

де $B(t)$ – доступна пропускна здатність каналу зв'язку (Bandwidth) між бортовим Edge-пристроєм та хмарним сервером у момент часу;

$L(t)$ – затримка передачі пакетів даних у мережі (Latency);

$C(t)$ – поточний доступний обчислювальний ресурс (Computing power) та рівень автономного енергозабезпечення граничного пристрою.

Вибір режиму функціонування системи (повністю автономний граничний інференс, часткова асинхронна синхронізація або повна хмарна оптимізація) визначається за допомогою порогової функції доступності мережевої інфраструктури (2.4).

$$f_{mode}(S(t)) = \begin{cases} \text{Edge-only}, & \text{якщо } B(t) < B_{min} \text{ або } L(t) > L_{max} \\ \text{Hybrid}, & \text{якщо } B_{min} \leq B(t) < B_{opt} \\ \text{Cloud-sync}, & \text{якщо } B(t) \geq B_{opt} \end{cases}. \quad (2.4)$$

При цьому процес квантування (Quantization) не є лише інструментом стиснення моделі, а виступає критично важливою технікою оптимізації. Він дозволяє здійснити перехід від 32-бітних чисел з рухомою комою до 8-бітних цілих значень (INT8), що суттєво зменшує вимоги до оперативної пам’яті та обчислювальної складності нейронної мережі, практично не впливаючи на точність її результатів. Таке рішення забезпечує можливість виконання інференсу безпосередньо на бортових платформах, що мають обмежену теплову потужність та енергоспоживання, уникаючи критичного перегріву процесора чи «троттлінгу» (зниження тактової частоти) під час виконання складних завдань у реальному часі.

Відтак, логіка перемикання між хмарною та граничною обробкою не є спонтанною; вона ґрунтується на концепції кінцевого автомата (State Machine), де перехід між режимами відбувається на основі предиктивного аналізу якості мережі. Уникнення повного розриву управління досягається завдяки завчасному переходу в режим часткової синхронізації, що підвищує загальну надійність системи. Такий багаторівневий підхід дозволяє зберігати керованість транспортним засобом навіть у зонах повної

відсутності зв’язку, що є критичною вимогою для функціонування логістики в умовах воєнного стану.

Для систематизації розроблених правил адаптації та забезпечення прозорості логіки перемикання обчислювальних режимів сформовано критерії функціонування системи, які зведено в таблицю 2.1.

Таблиця 2.1 – Характеристика режимів роботи інтелектуальної системи залежно від стану інфраструктури зв'язку

Режим роботи	Стан пропускної здатності (B(t))	Рівень затримки мережі (L(t))	Локальне навантаження (Edge AI)	Роль хмарного сервера (Cloud)
Автономний (Edge-only)	Критичний ($B(t) < B_{min}$) або зв'язок відсутній	Повна деградація / таймаут ($L(t) > L_{max}$)	100% обчислень: локальний інференс квантованої моделі, обхід графа.	Відсутня. Дані телеметрії накопичуються в локальному кеші пристрою.
Гібридний (Hybrid)	Обмежений ($B_{min} \leq B(t) < B_{opt}$)	Допустимий для некритичних задач	Локальна обробка та фільтрація сирих даних. Інференс Edge-моделі для швидких рішень.	Асинхронний обмін агрегованою телеметрією. Періодичне оновлення локальних карт.
Синхронний (Cloud-sync)	Оптимальний ($B(t) \geq B_{opt}$)	Мінімальний ($L(t) \leq L_{max}$)	Первинна токенізація даних із сенсорів, передача потоку телеметрії у хмару.	Глобальний аналіз логістичної мережі, координація між усіма Edge-вузлами.

Таблиця систематизує механізм адаптивної реакції системи на динамічну зміну якості мережевої інфраструктури, визначаючи логіку перемикання між трьома режимами (автономним, гібридним та синхронним) на основі порогових значень пропускної здатності $B(t)$ та

затримки $L(t)$. Ця матриця рішень гарантує «інтелектуальну гнучкість» архітектури: у разі критичних збоїв чи повної відсутності зв'язку активується автономний режим із локальним інференсом квантованих моделей, що забезпечує безперервний захист водія, тоді як при стабільному з'єднанні система переходить до хмарної координації та глобальної аналітики.

Графічно взаємозв'язок між описаними інфраструктурними обмеженнями, параметрами логістичного графа середовища та процесом адаптації режимів обчислень представлено на структурній схемі (рисунок 2.2).

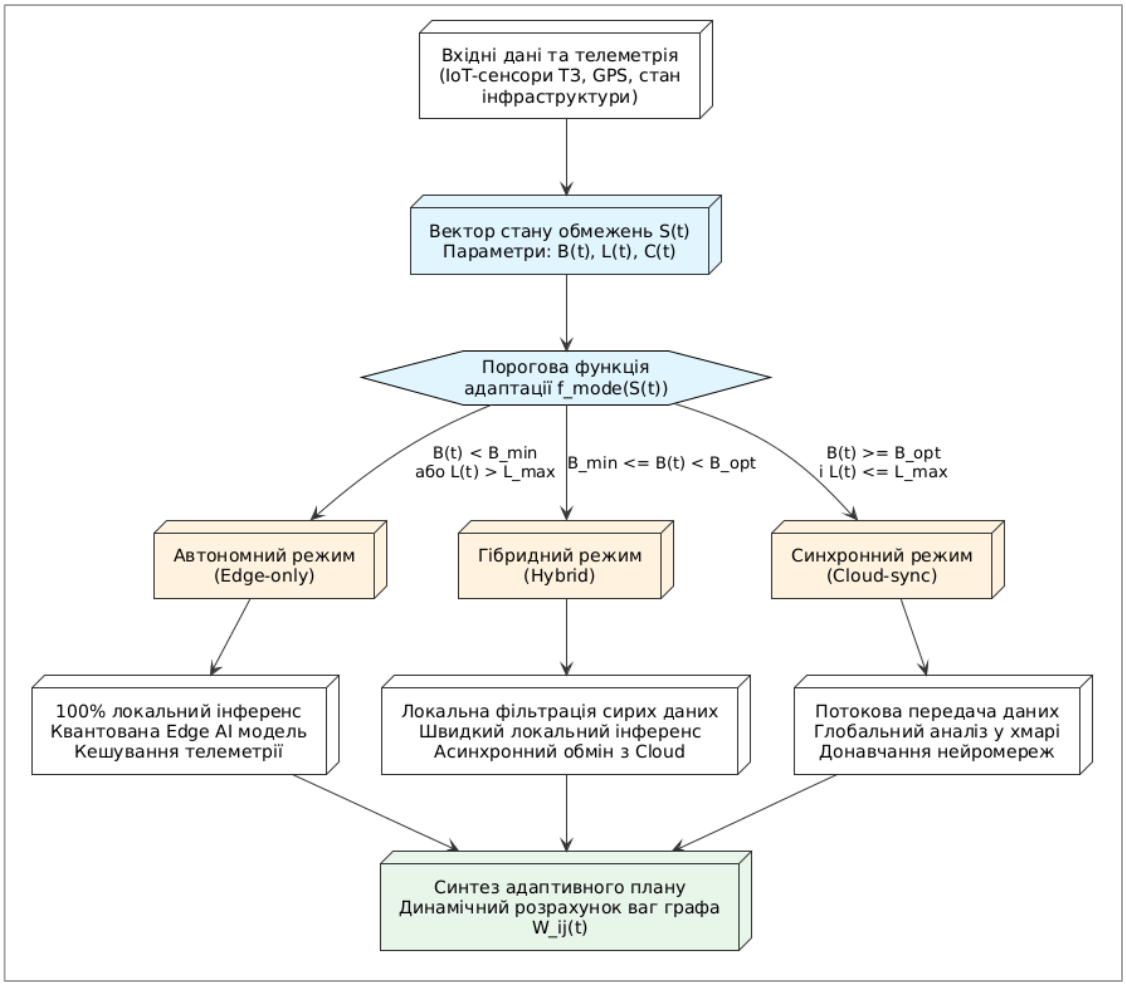


Рисунок 2.2 – Структурна схема адаптації системи до інфраструктурних обмежень

Таким чином, розроблена модель логістичного середовища трансформує класичну задачу маршрутизації транспортних засобів у динамічну задачу оптимізації в умовах обмежених ресурсів зв'язку та обчислень. Це створює теоретичний базис для безпосереднього проєктування архітектури Edge AI модуля.

2.3 Проєктування Edge AI модуля локального збору та обробки даних

Модуль граничних обчислень (Edge AI), розміщений безпосередньо на борту транспортного засобу (наприклад, на базі мікрокомп'ютерів класу NVIDIA Jetson Nano або спеціалізованих логістичних IoT-шлюзів), є ключовим компонентом для забезпечення автономності системи. Його головне завдання – безперервний збір телеметрії, її попередня обробка та локальний інференс (виконання моделі) для розрахунку динамічних параметрів маршруту $T_{ij}(t)$ та $P_{ij}(t)$, введених у підрозділі 2.2.

Архітектурно модуль складається з трьох логічних підсистем: збору даних, попередньої обробки (препроцесингу) та нейромережевого інференсу.

Підсистема збору та агрегації даних. У кожен момент часу t підсистема опитує наявні бортові сенсори та формує вхідний багатовимірний вектор даних $X(t)$ (2.5).

$$X(t) = [D_{veh}(t), D_{env}(t), D_{loc}(t)], \tag{2.5}$$

де $D_{veh}(t)$ – телеметричні дані стану транспортного засобу, отримані з шини CAN (швидкість, рівень пального, температура двигуна, коди помилок OBD-II);

$D_{env}(t)$ – параметри навколишнього середовища (показники акселерометра для оцінки якості дорожнього покриття, дані з датчиків освітленості та вологості);

$D_{loc}(t)$ – просторово-часові координати (GPS/ГЛОНАСС).

Сирі дані з сенсорів часто містять шуми, пропуски (наприклад, втрата сигналу GPS у тунелі) та викиди. Для підготовки даних до подачі в нейронну мережу модуль виконує такі операції:

– фільтрація шумім. Застосування алгоритму фільтра калмана для згладжування gps-треку та усунення аномальних стрибків координат;

– віконна агрегація. Оскільки edge-модель аналізує часові ряди, дані групуються у ковзне часове вікно розміром w. матриця вхідних ознак для моделі формується як $x_{window} = [x(t-w), \ldots, x(t-1), x(t)]$;

– нормалізація. Масштабування різнорідних значень сенсорів у діапазон $[0,1]$ або $[-1,1]$ для стабілізації роботи нейромережі.

З огляду на жорсткі обмеження обчислювального ресурсу $C(t)$, такі як проблема енергоспоживання та відведення тепла на бортових пристроях, використання стандартних глибоких нейромереж є недоцільним. Тому основою розробленого модуля є квантована рекурентна нейронна мережа (Quantized RNN або полегшена 1D-CNN).

Для адаптації моделі під Edge-архітектуру застосовано метод посттренувального квантування (Post-Training Quantization). Ваги моделі W, які зазвичай зберігаються у форматі з плаваючою комою 32 біти (FP32), перетворюються у 8-бітний цілочисельний формат (INT8) за формулою (2.6).

$$W_{INT8} = \text{round}\left(\frac{W_{FP32}}{S}\right) + Z, \tag{2.6}$$

де S(Scale) – масштабний множник, який визначає крок квантування;

Z(Zero-point) – цілочисельне значення зсуву, що відповідає реальному нулю.

Ця оптимізація дозволяє зменшити розмір моделі в чотири рази та прискорити локальний інференс без суттєвої втрати точності. Завданням

цієї локальної моделі є аналіз матриці X_{window} для прогнозування штрафної функції ризику $P_{ij}(t+1)$, що дозволяє системі передбачити ймовірність поломки або критичної затримки на наступній ділянці маршруту ще до того, як вона відбудеться. У разі перевищення допустимого порогу ризику, модуль самостійно ініціює перерахунок ваг логістичного графа, згідно з функцією $W_{ij}(t)$, та формує команду на зміну маршруту, навіть перебуваючи в автономному режимі.

2.4 Проєктування алгоритму адаптивного управління маршрутизацією

Отримані модулем Edge AI (підрозділ 2.3) прогнозні значення інфраструктурних ризиків та затримок є вхідними даними для підсистеми прийняття рішень. Її головне завдання – розрахувати оптимальний шлях у динамічному графі логістичної мережі $G = (V, E)$ та оперативно коригувати його у разі зміни ваги ребер $W_{ij}(t)$ без повного перерахунку всього графа, що критично для економії обчислювального ресурсу $C(t)$.

Цільова функція маршрутизації. Нехай задано початкову вершину маршруту v_{start} та кінцеву точку призначення v_{goal}. Алгоритм повинен знайти таку послідовність суміжних вершин (маршрут R), яка мінімізує інтегральну вартість переміщення в умовах мінливого середовища (2.7).

$$R^* = \arg\min_R \sum_{(v_i,v_j)\in R} W_{ij}\,(t). \qquad (2.7)$$

За умови, що для кожного обраного ребра виконується вимога фізичної прохідності (штрафна функція ризику не перевищує критичного порогу блокування (2.8).

$$P_{ij}(t) < P_{max}). \qquad (2.8)$$

Адаптація алгоритму динамічного пошуку (D Lite)*. Для реалізації адаптивної маршрутизації на рівні граничних обчислень обрано модифікацію алгоритму D* Lite. На відміну від класичного алгоритму A*, який вимагає повного повторного розрахунку шляху при кожній зміні середовища, D* Lite будує маршрут від v_{goal} до v_{start} і підтримує граф у пам'яті. Коли Edge AI модуль фіксує аномалію (наприклад, раптове перекриття дороги, зміна $W_{ij}(t)$), алгоритм локально оновлює лише ті вершини, на які вплинула зміна.

Оціночна функція пріоритету для розширення вершин в алгоритмі базується на двох ключових метриках: $g(v)$ – поточна розрахована вартість шляху від вершини до цілі, та $rhs(v)$ – оцінка вартості на основі сусідніх вершин (look-ahead value). Для будь-якої вершини u (2.9).

$$rhs(u) = \min_{v \in Succ(u)} \big(W_{uv}(t) + g(v)\big), \qquad (2.9)$$

де $Succ(u)$ – множина доступних наступних вершин з вузла u.

Евристична складова алгоритму. Щоб пришвидшити сходження алгоритму та направити пошук у бік оптимального рішення, використовується евристична функція $h(v, v_{start})$, яка оцінює «відстань» до поточної позиції транспортного засобу. У розробленій системі ця евристика є композитною і враховує не лише геометричну відстань, а й глобальний прогноз Edge-моделі (2.10).

$$h(v_i, v_{start}) = d_{euclid}(v_i, v_{start}) \cdot \big(1 + \bar{P}_{regional}\big), \qquad (2.10)$$

де $\bar{P}_{regional}$ – усереднений коефіцієнт інфраструктурного ризику в даному регіоні, отриманий з хмарного рівня (Cloud-sync) під час останнього успішного сеансу зв'язку.

Процес маршрутизації на борту транспортного засобу відбувається за таким ітеративним сценарієм:

– ініціалізація. Під час завантаження на складі (за умови стабільного $b(t) \geq b_{opt}$), транспортний засіб отримує від хмарного сервера глобальний граф g із базовими вагами та будує первинний оптимальний маршрут;

– моніторинг. Під час руху edge ai модуль безперервно збирає вектор даних $x(t)$ та розраховує локальні оновлення для найближчих ребер $w_{ij}(t)$;

– тригер адаптації. Якщо зміна ваги найближчого ребра (наприклад, виявлено затор або аварію) перевищує встановлений поріг чутливості ($\delta w_{ij} > \epsilon$), система генерує переривання;

– локальний перерахунок. Алгоритм d* lite оновлює значення $rhs(u)$ для змінених вершин та розповсюджує ці зміни по графу, миттєво генеруючи новий маршрут об'їзду без звернення до хмарного сервера.

Такий підхід гарантує, що система адаптивного управління логістикою залишається повністю працездатною навіть у режимі Edge-only (при повній відсутності зв'язку), забезпечуючи безперервну оптимізацію логістичних процесів в умовах жорстких інфраструктурних обмежень.

Використання D* Lite безпосередньо розв'язує задачу автономної навігації (його використовують навіть у марсоходах).

2.5 Проєктування підсистеми диспетчеризації та моніторингу

Попри високий рівень автономності граничних вузлів (Edge Nodes), що функціонують безпосередньо на борту транспортних засобів, загальна координація та глобальна оптимізація логістичної інфраструктури вимагають наявності централізованого інструменту управління. Для вирішення цієї задачі спроєктовано підсистему диспетчеризації та моніторингу, яка розгортається на хмарному рівні (Cloud Layer) архітектури системи. Саме цей рівень забезпечує перехід від локального керування

окремим транспортним засобом до стратегічного управління всією корпоративною екосистемою, формуючи свого роду «цифровий двійник» усієї логістичної мережі.

Головною метою цієї підсистеми є не лише агрегація асинхронних потоків телеметрії від усіх активних граничних вузлів, а й створення багаторівневого конвеєра обробки даних (Data Pipeline). Підсистема забезпечує: візуалізацію стану мережі в режимі реального часу, моніторинг інфраструктурних обмежень, аналіз відхилень від планових показників та надання інтерфейсу взаємодії для оператора (диспетчера). Архітектурно це реалізується через розділення потоків даних на «гарячі» (Hot Path) – для миттєвих сповіщень та оперативної реакції диспетчера, і «холодні» (Cold Path) – для накопичення історичних даних, які в подальшому використовуються для аналітичного звітування та донавчання (fine-tuning) моделей нейронних мереж, що працюють на борту.

Оскільки зв’язок із граничними вузлами може тимчасово перериватися через інфраструктурні обмеження (режим Edge-only), підсистема моніторингу проєктується на основі архітектури, що керується подіями (Event-Driven Architecture), із використанням брокера повідомлень (Message Broker) на базі протоколу MQTT або AMQP. Такий підхід забезпечує «часову розв'язку» (temporal decoupling) між вузлами системи: диспетчер бачить актуальний стан мережі, оперуючи тими даними, які надійшли останніми, тоді як брокер гарантує доставку історичних пакетів даних згідно з їхніми часовими мітками. Це гарантує, що повідомлення від транспортних засобів не будуть втрачені: у разі деградації каналу зв'язку дані накопичуються в локальному кеші бортового комп'ютера, а при переході в режим Hybrid або Cloud-sync автоматично відправляються в чергу брокера для подальшої синхронізації та інтеграції в центральну базу даних.

Для мінімізації навантаження на обмежені канали зв'язку та забезпечення надійності передачі застосовується механізм рівня якості

обслуговування (Quality of Service, QoS), що підтримується протоколом MQTT. Зокрема, для критичних сповіщень (наприклад, про інфраструктурні збої або екстрену зміну маршруту) використовується рівень QoS 1 (доставка щонайменше один раз) або QoS 2 (доставка точно один раз), що гарантує фіксацію події на хмарному рівні навіть за умов нестабільного з'єднання. Водночас, з метою економії пропускної здатності $B(t)$, перед відправкою до черги брокера агреговані дані телеметрії проходять етап серіалізації з використанням легковагових бінарних форматів.

Процес обробки та інтеграції даних на хмарному рівні включає такі етапи:

– inbound-фільтрація. Розподіл вхідних пакетів телеметрії за пріоритетністю (критичні сповіщення про аварії чи зміну маршруту обробляються в першу чергу);

– оновлення глобального графа знань. Якщо декілька граничних вузлів зафіксували зміну ваги конкретного ребра графа δw_{ij} (наприклад, тривала затримка на певній ділянці дороги), підсистема виконує перерахунок глобальної матриці логістичної мережі;

– вирішення конфліктів синхронізації (conflict resolution). У випадку, коли дані від різних edge-вузлів щодо стану однієї й тієї самої ділянки інфраструктури суперечать один одному, хмарна підсистема застосовує часові мітки (Timestamps) та коефіцієнти довіри до джерела, обираючи найбільш релевантну інформацію.

Реалізація описаних механізмів обробки даних становить лише технічний фундамент системи, тоді як кінцева ефективність логістичного управління залежить від здатності оператора швидко та адекватно інтерпретувати отриману інформацію.

Загальну схему організації інформаційних потоків, архітектуру синхронізації даних через брокер повідомлень та їх подальшу обробку на хмарному рівні представлено у вигляді послідовної вертикальної структури на рисунку 2.3.

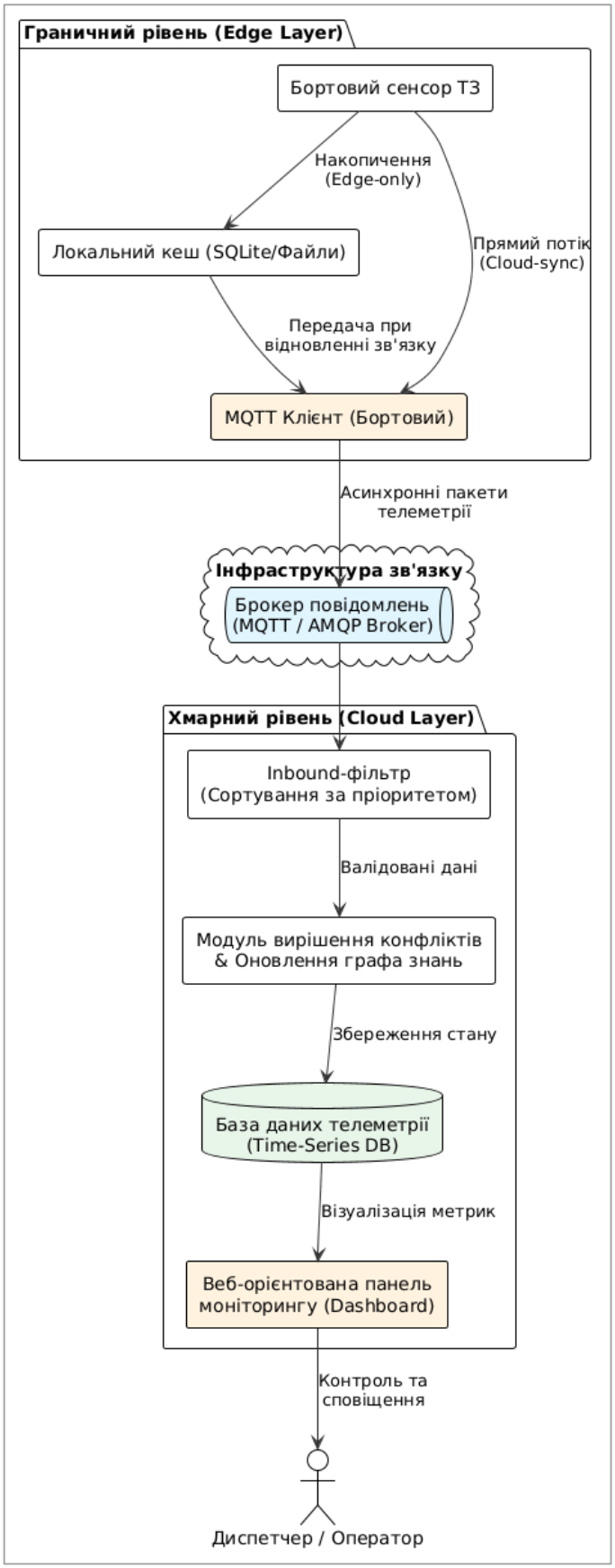


Рисунок 2.3 – Структурна схема інформаційних потоків підсистеми моніторингу

Графічний інтерфейс користувача містить такі ключові модулі:

– інтерактивна геоінформаційна карта. Відображає логістичний граф мережі $g = (v, e)$, поточне розташування транспортних засобів та колірне кодування ділянок доріг відповідно до рівня їхнього інфраструктурного ризику $p_{ij}(t)$;

– панель стану граничних вузлів. Візуалізує поточний обчислювальний режим кожного бортового пристрою (*edge-only*, *hybrid*, *cloud-sync*), а також метрики якості зв'язку – доступну пропускну здатність $b(t)$ та затримку $l(t)$;

– модуль сповіщень (alerting system). Автоматично сигналізує диспетчеру, якщо бортовий edge ai модуль виявив критичну аномалію та самостійно прийняв рішення про зміну початкового маршруту в автономному режимі.

Завдяки такому підходу підсистема диспетчеризації не дублює обчислення граничних вузлів, а виконує функцію високорівневого нагляду та координації, забезпечуючи синергію між автономним штучним інтелектом на борту та стратегічним управлінням логістичними процесами з боку людини.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Вибір засобів розробки та технологій

Для програмної реалізації спроєктованої інтелектуальної системи адаптивного управління логістикою було обрано мову програмування Python. Цей вибір обґрунтований високою гнучкістю мови, наявністю потужної екосистеми модулів для роботи з даними та алгоритмами машинного навчання, що ідеально підходить для прототипування рішень на базі граничних обчислень.

Основним інструментом для побудови математичної моделі логістичного середовища стала спеціалізована бібліотека NetworkX. У рамках даної роботи вона використовується для представлення інфраструктури у вигляді зваженого графа, а також для реалізації базових алгоритмів обходу вузлів та зберігання динамічних атрибутів на транспортних ребрах.

Для графічного відображення результатів роботи системи та візуального контролю інфраструктурних обмежень застосовано можливості бібліотеки Matplotlib. Вона дозволяє рендерити логістичну мережу, відображати поточне місцеперебування транспортного засобу та за допомогою колірного кодування підсвічувати виявлені інфраструктурні аномалії, такі як затори чи перекриття доріг.

Додатково в процесі симуляції роботи Edge AI модуля використано стандартні пакети Python, зокрема Math та Random. Вони відповідають за генерацію ймовірнісних інфраструктурних ризиків та розрахунок динамічних ваг ребер графа на основі вхідних псевдоданих, імітуючи надходження реальної телеметрії з бортових сенсорів. Середовищем розробки виступив інтегрований комплекс PyCharm, який забезпечує зручні інструменти для відлагодження алгоритмів динамічної маршрутизації та тестування загальної архітектури.

3.2 Проєктування логістичного графа транспортної мережі України

Перехід від теоретико-методологічного обґрунтування архітектури дворівневої інтелектуальної системи до її практичної програмної реалізації вимагає побудови формалізованої моделі об'єкта дослідження. Логістична мережа як складна територіально-розподілена система потребує застосування математичного апарату, здатного адекватно відобразити зв’язки між об’єктами інфраструктури та динаміку їхнього функціонування. Одним із найбільш ефективних підходів до вирішення задач логістичного планування в умовах невизначеності є моделювання транспортної мережі засобами теорії графів, що дозволяє створити цифровий аналог інфраструктури, придатний для подальшої обробки алгоритмами адаптивної маршрутизації та інтеграції безпосередньо в програмне ядро Edge AI.

Першим етапом програмної реалізації є створення моделі логістичного середовища, описаного в попередньому розділі роботи. Для наочної демонстрації адаптивних можливостей системи було розроблено спрощений граф базових транспортних магістралей України, який об'єднує ключові логістичні хаби. Кожен вузол або вершина графа представляє певне місто, а кожне ребро відображає наявну автомагістраль між цими населеними пунктами.

Відповідно до розробленої математичної моделі, ребрам графа присвоюються як статичні, так і динамічні атрибути. До статичних параметрів належить фізична відстань у кілометрах, тоді як динамічні атрибути включають базовий рівень інфраструктурного ризику та пропускну здатність ділянки в поточний момент часу.

Для програмної реалізації моделі було обрано бібліотеку NetworkX. Це забезпечує високу гнучкість при маніпуляції атрибутами вузлів та ребер, що є критично важливим для нашої задачі. Використання словникової структури даних для кожного ребра дозволяє не лише зберігати статичні

значення (наприклад, фізичну відстань), але й динамічно змінювати їх під час виконання алгоритму. Така архітектура безпосередньо підтримує концепцію Edge AI: бортовий обчислювальний вузол може в реальному часі оновлювати параметри ризику (base_risk) або часу в дорозі (T(e, t)), не перебудовуючи структуру графа, що суттєво економить обчислювальні ресурси. Програмну реалізацію модуля генерації логістичного графа транспортної мережі наведено в лістингу 3.1. Повний текст модуля ініціалізації графа міститься у додатку А.

Лістинг 3.1 – Модуль ініціалізації логістичного графа транспортної мережі

```
import networkx as nx

def create_ukraine_logistics_graph():
    """Створює зважений граф ключових логістичних маршрутів."""
    G = nx.Graph()

    # Визначення вузлів (міст)
    cities = ["Київ", "Житомир", "Рівне", "Львів", "Полтава",
"Харків", "Дніпро", "Одеса"]
    G.add_nodes_from(cities)

    # Визначення ребер (маршрутів) з атрибутами ваги та ризику
    edges = [

        ("Київ", "Житомир", {'distance': 140, 'base_risk':
0.1}),
        ("Житомир", "Рівне", {'distance': 190, 'base_risk':
0.1}),
        ("Рівне", "Львів", {'distance': 210, 'base_risk': 0.2}),
        ("Київ", "Полтава", {'distance': 340, 'base_risk':
0.1}),
        ("Полтава", "Харків", {'distance': 140, 'base_risk':
0.3}),
        ("Київ", "Умань", {'distance': 210, 'base_risk': 0.1}),
        ("Умань", "Одеса", {'distance': 270, 'base_risk': 0.1}),
```

Продовження лістингу 3.1

```
    ]

    G.add_edges_from(edges)
    return G
```

Розроблена функція візуалізації дозволяє оперативно контролювати стан інфраструктури під час симуляції. Вона алгоритмічно підсвічує ділянки з високим рівнем ризику або заторами червоним кольором, а прокладений бортовим модулем альтернативний маршрут – зеленим, що є необхідним інструментом для тестування підсистеми моніторингу.

3.3 Реалізація емулятора Edge AI та генерації обмежень

Наступним кроком у програмній реалізації є розробка модуля, який імітує роботу граничного обчислювального пристрою та динамічну зміну інфраструктурних умов. У реальних умовах збір та обробка потоку телеметрії з транспортного засобу вимагає розгортання складного апаратного комплексу з підключенням до CAN-шини. З огляду на обмеження лабораторного середовища та необхідність швидкого прототипування, було прийнято рішення розробити програмний емулятор підсистеми Edge AI. Такий підхід дозволяє створити «цифровий двійник» логістичного вузла, в межах якого можна безпечно тестувати алгоритми прийняття рішень без ризику для реального обладнання. Емулятор виконує дві ключові функції: стохастичну генерацію інфраструктурних обмежень на маршруті та прийняття автономних рішень щодо допустимості подальшого руху на основі бортового інференсу.

Для імітації нестабільного логістичного середовища створено клас InfrastructureEnvironment. Цей програмний компонент є фундаментальним для забезпечення валідності моделі, оскільки він відповідає за періодичне оновлення ваг ребер у логістичному графі. Під час кожної ітерації симуляції

модуль з певною ймовірністю генерує аномальні події, такі як затори, погіршення погодних умов або пошкодження дорожнього покриття. Ці події математично відображаються як різке зростання штрафної функції ризику для конкретного ребра. Використання стохастичного моделювання дозволяє уникнути детермінізму: граф стає динамічною системою, де оптимальний маршрут може змінюватися в режимі реального часу, змушуючи алгоритми адаптивної маршрутизації до постійної переоцінки ситуації.

Безпосередню обробку цих змін виконує клас EdgeAIEmulator, який розміщується на «віртуальному борту» транспортного засобу. Емулятор безперервно сканує найближчі доступні вузли та аналізує їхній поточний рівень ризику, реалізуючи концепцію граничних обчислень (Edge Computing). На відміну від централізованих систем, де аналіз виконується на віддаленому сервері, тут рішення приймається локально, що мінімізує затримку (latency) реакції на зміну обстановки. Замість використання ресурсомісткої нейронної мережі, для цілей прототипування застосовано спрощену ймовірнісну модель порогового прийняття рішень. Якщо розрахований ризик проїзду по найближчому ребру перевищує встановлений критичний поріг, модуль імітує поведінку автономного штучного інтелекту: він фіксує неможливість подальшого руху за старим планом і генерує сигнал для підсистеми маршрутизації, яка миттєво ініціює пошук альтернативного шляху.

Важливою архітектурною перевагою даної реалізації є розділення відповідальності (Separation of Concerns). Модуль InfrastructureEnvironment не має знань про логіку прийняття рішень – він лише створює виклики для системи. Натомість EdgeAIEmulator працює як «агент», що діє суто на основі доступних йому локальних даних. Така незалежність модулів дозволяє в майбутньому легко замінити просту ймовірнісну модель порогового прийняття рішень на складнішу (наприклад, повноцінну модель глибокого навчання, розгорнуту за допомогою TensorFlow Lite або ONNX Runtime) без необхідності модифікації решти програмного комплексу.

Детальна програмна реалізація розроблених класів середовища моделювання та бортового емулятора Edge AI представлена у лістингу 3.2. З програмним кодом імітації середовища можна ознайомитись у додатку Б.

Лістинг 3.2 – Програмна реалізація середовища та емулятора Edge AI

```
import random

class InfrastructureEnvironment:
    """Імітація динамічних інфраструктурних ризиків."""
    def __init__(self, graph):
        self.graph = graph

    def generate_dynamic_constraints(self):
        problem_edges = []
        for u, v, data in self.graph.edges(data=True):
            # Імітація випадкових інфраструктурних подій (15% ймовірність)
            if random.random() < 0.15:
                data['current_risk'] = data['base_risk'] + random.uniform(0.6, 0.9)
                problem_edges.append((u, v))
            else:
                data['current_risk'] = data['base_risk'] + random.uniform(0.0, 0.1)
        return problem_edges

class EdgeAIEmulator:
    """Емулятор бортового модуля для порогового інференсу."""
    def __init__(self, current_node, risk_threshold=0.7):
        self.current_node = current_node
        self.risk_threshold = risk_threshold

    def scan_ahead(self, graph, next_node):
        edge_data = graph.get_edge_data(self.current_node, next_node)
        if not edge_data: return False
```

Продовження лістингу 3.2

```
            # Отримання актуального ризику та прийняття рішення
(Inference)
            predicted_risk      =      edge_data.get('current_risk',
edge_data['base_risk'])
            return predicted_risk > self.risk_threshold
```

Наведений програмний код демонструє механізм адаптації системи до непередбачуваних умов. Завдяки розділенню логіки на середовище моделювання та бортовий аналізатор досягається високий рівень модульності розробленого прототипу. Використання генератора псевдовипадкових чисел дозволяє під час кожного запуску програми отримувати унікальні сценарії деградації інфраструктури, що є необхідною умовою для всебічного тестування алгоритмів динамічного пошуку найкоротшого шляху на наступних етапах розробки. Таким чином, реалізована модель є надійним підґрунтям для перевірки гіпотези про те, що автономні системи Edge AI здатні ефективно функціонувати в умовах високої невизначеності, забезпечуючи вищий рівень безпеки та стійкості логістичних процесів порівняно з традиційними системами, які повністю залежать від стабільного зв'язку з центральним сервером.

3.4 Програмна реалізація алгоритму адаптивної маршрутизації

Ефективність функціонування логістичної системи в умовах критичної інфраструктурної нестабільності безпосередньо залежить від здатності програмного комплексу до оперативного перерахунку маршрутів при виникненні непередбачуваних перешкод. Традиційні підходи до маршрутизації, орієнтовані на оптимізацію лише одного показника – фізичної відстані, є недостатніми для задач, де критерії безпеки, часу та актуального стану доріг набувають пріоритетного значення. В архітектурі Edge AI алгоритм маршрутизації виступає інтелектуальним «рушієм», який

не просто шукає шлях, а постійно верифікує його валідність, інтегруючи дані від бортових сенсорів та зовнішніх джерел інформації в єдиний контур прийняття рішень.

Центральним компонентом програмного забезпечення розробленої системи є алгоритм адаптивної маршрутизації, який функціонує в тісній взаємодії з модулем Edge AI. Основне завдання цього компонента полягає в динамічному розрахунку найкоротшого та найбезпечнішого шляху в логістичному графі від поточної позиції транспортного засобу до цільового пункту призначення. З огляду на необхідність економії обчислювальних ресурсів на бортовому пристрої, алгоритм реалізує стратегію покрокового перерахунку. Замість того, щоб безперервно оновлювати весь глобальний граф логістичної мережі, система приймає рішення локально, реагуючи виключно на ті інфраструктурні зміни, які фіксуються емулятором безпосередньо по ходу руху.

Для реалізації цієї логіки було розроблено клас AdaptiveRouter. Під час ініціалізації об'єкт цього класу отримує доступ до логістичного графа, координат початкової та кінцевої точок, а також до екземпляра емулятора Edge AI. Базовий пошук оптимального шляху спирається на алгоритм Дейкстри, що реалізований у бібліотеці NetworkX. Проте ключовою відмінністю є використання динамічного атрибута поточного ризику замість класичної фізичної відстані. На кожній ітерації руху система формує попередній план маршруту, після чого бортовий модуль сканує найближчий вузол на наявність інфраструктурних аномалій, згенерованих середовищем симуляції.

Якщо на запланованій ділянці виявляється критичне обмеження, наприклад, раптовий затор або перекриття дороги, підсистема маршрутизації ініціює процедуру уникнення перешкоди. Програмно це реалізується шляхом локального переприсвоєння ваги проблемного ребра графа до нескінченно великого значення. Після цього алгоритм миттєво виконує повторний розрахунок маршруту з поточної точки перебування

транспортного засобу. Цей цикл безперервно повторюється до моменту успішного прибуття в цільовий вузол. Програмну реалізацію адаптивного диспетчера маршрутів наведено в лістингу 3.3. З вихідним кодом алгоритму адаптивної маршрутизації можна ознайомитись у додатку В.

Лістинг 3.3 – Програмна реалізація адаптивного маршрутизатора

```
import networkx as nx

class AdaptiveRouter:
    """Клас для динамічного прокладання маршруту в реальному часі."""

    def calculate_optimal_path(self):
        # Алгоритм Дейкстри з урахуванням динамічного атрибута ризику
        return nx.shortest_path(self.graph, self.current_node, self.target_node, weight='current_risk')

    def run_simulation(self):
        while self.current_node != self.target_node:
            planned_path = self.calculate_optimal_path()
            if not planned_path: break

            next_node = planned_path[1]

            # Інтелектуальна перевірка: Edge AI сканує наступну ділянку
            if self.emulator.scan_ahead(self.graph, next_node):
                # [КРИТИЧНО] Якщо виявлено небезпеку, блокуємо ребро в графі
                self.graph[self.current_node][next_node]['current_risk'] = float('inf')
                continue # Автоматичний перерахунок маршруту

            self.current_node = next_node
            self.emulator.current_node = next_node
```

3.5 Тестування та візуалізація результатів

Заключним етапом програмної реалізації є інтеграція всіх розроблених модулів у єдине симуляційне середовище для комплексного тестування працездатності інтелектуальної системи. Верифікація алгоритмічного забезпечення вимагає створення умов, максимально наближених до реальних інфраструктурних викликів, що характеризуються високою динамікою змін та стохастичним характером виникнення загроз. З цією метою було розроблено головний виконуваний скрипт, який об'єднує ініціалізацію логістичного графа, генерацію інфраструктурних обмежень та роботу алгоритму адаптивної маршрутизації під управлінням емулятора Edge AI. Процес тестування розпочинається зі створення віртуальної транспортної мережі України, яка слугує «пісочницею» для перевірки стійкості системи.

Для забезпечення об'єктивності результатів, середовище генерує проблемні ділянки (затори, пошкодження інфраструктури) випадковим чином у кожному окремому циклі запуску (test run). Такий підхід дозволяє перевірити не лише здатність системи до вибору «найкращого» шляху, а й її стійкість до ситуацій, де ідеальний маршрут стає недоступним через форс-мажорні обставини. Важливим аспектом тестування є моніторинг поведінки алгоритму при наближенні до критичної точки: як Edge-модуль приймає рішення про блокування ребра та подальший пошук обхідного шляху.

Після ініціалізації середовища система запускає віртуальний транспортний засіб із заданої початкової точки до цільового пункту призначення. У процесі руху бортовий модуль безперервно аналізує найближчі ребра графа на відповідність порогам ризику. Якщо на запланованому маршруті виявляється перешкода (ризик перевищує поріг), алгоритм миттєво перераховує шлях в обхід заблокованої ділянки. Для наочної перевірки коректності прийнятих рішень використовується функція графічного рендерингу, яка виводить фінальну карту на екран. На цій

візуалізації проблемні відрізки доріг підсвічуються червоним кольором (маркери блокування), а фактичний прокладений адаптивний маршрут підсвічуються зеленим, що дозволяє візуально підтвердити успішність обходу інфраструктурних перешкод. Програмний код головного модуля симуляції наведено в лістингу 3.4. Повний цикл інтеграційного тестування системи міститься у додатку Г.

Лістинг 3.4 – Головний модуль тестування системи

```
if __name__ == "__main__":
    # 1. Ініціалізація графа
    graph = create_ukraine_logistics_graph()

    # 2. Налаштування задачі
    start_city, target_city = "Київ", "Львів"

    # 3. Генерація динамічних перешкод (Стрес-тест інфраструктури)
    environment = InfrastructureEnvironment(graph)
    problem_edges = environment.generate_dynamic_constraints()
    print("Згенеровані критичні ділянки:", problem_edges)

    # 4. Ініціалізація Edge-вузла (бортового ШІ)
    edge_emulator = EdgeAIEmulator(current_node=start_city, risk_threshold=0.6)

    # 5. Виконання адаптивної маршрутизації
    router = AdaptiveRouter(graph, start_city, target_city, edge_emulator)
    actual_path = router.run_simulation()

    # 6. Рендеринг результатів
    draw_graph(graph, current_node=target_city, problem_edges=problem_edges, path=actual_path)
```

Проведене тестування підтвердило повну працездатність спроєктованої архітектури. Результати експериментів демонструють, що

система успішно ідентифікує небезпечні ділянки безпосередньо по ходу руху (на борту) та оперативно генерує альтернативні маршрути, уникаючи «інформаційного вакууму» при втраті зв'язку з центральним сервером.

Аналіз графічних виводів дозволяє зробити висновок про високу адаптивність алгоритму: навіть при штучному створенні умов, коли ключові магістралі виявляються заблокованими (ризик перевищує гранично допустимий рівень), система знаходить рішення, якщо воно існує в межах графа. Візуалізація результатів слугує надійним інструментом для диспетчерського контролю, оскільки дозволяє не лише бачити фінальний результат, а й аналізувати логіку кожного прийнятого рішення «на льоту».

Спроєктований комплекс може бути масштабований шляхом ускладнення графа (додавання міських вулиць, доріг місцевого значення) та вдосконалення евристик оцінки ризику, що відкриває широкі перспективи для подальшої інтеграції в повноцінні логістичні системи управління автопарками.

4 РОЗРОБКА КОРИСТУВАЦЬКОГО ІНТЕРФЕЙСУ ТА ВІЗУАЛІЗАЦІЯ ДАНИХ

4.1 Проєктування архітектури взаємодії користувачів із системою

Ефективність функціонування інтелектуальної системи адаптивного управління логістикою залежить не лише від математичних алгоритмів, але й від якості взаємодії між користувачем та програмним забезпеченням. Враховуючи дворівневу архітектуру розробленого рішення, підсистема візуалізації повинна задовольняти потреби двох ключових груп користувачів: водіїв транспортних засобів, які працюють із бортовими терміналами в умовах обмеженого зв'язку, та диспетчерів логістичного центру, які здійснюють глобальний моніторинг. Для забезпечення інтуїтивно зрозумілої взаємодії було застосовано методологію проєктування на основі прецедентів (Use Case modeling). У рамках розробленої моделі прецедентів головними акторами виступають диспетчер, водій та автономний програмний модуль Edge AI, який діє як незалежний агент.

Застосування методології Use Case дозволило формалізувати вимоги до інтерфейсів на ранніх етапах розробки, розділивши функціонал системи на стратегічний (для диспетчера) та тактичний (для водія). Особливу увагу приділено ергономіці: для водія інтерфейс проєктується згідно з принципами мінімалізму та забезпечення безпеки руху (Human-Computer Interaction), де головним пріоритетом є виключення відволікаючих чинників, тоді як диспетчер отримує доступ до інструментів комплексного аналізу «великих даних» (Big Data) для прийняття рішень на рівні всієї мережі.

Диспетчер має доступ до прецедентів глобального моніторингу транспортної мережі, відстеження стану зв'язку з кожним бортовим пристроєм та агрегованої аналітики щодо інфраструктурних ризиків. Водій

взаємодіє з підсистемою локальної навігації та отримує візуальні сповіщення про зміну маршруту, причому ці сповіщення мають найвищий пріоритет (interrupt-driven interaction). Модуль Edge AI функціонує у фоновому режимі, автоматично ініціюючи прецеденти перерахунку маршруту при виявленні перешкод. Така архітектура «мовчазного помічника» дозволяє суттєво знизити когнітивне навантаження на водія, оскільки система не просто повідомляє про проблему, а пропонує вже готове, математично оптимізоване рішення.

Діаграму прецедентів, яка візуалізує розподіл ролей та функцій у системі, наведено на рисунку 4.1. Вона відображає не лише взаємодію людей із системою, але й автономні ланцюжки реакцій, які запускаються програмними агентами без прямого втручання користувача, що є ключовою особливістю розробленої інтелектуальної логістичної платформи.

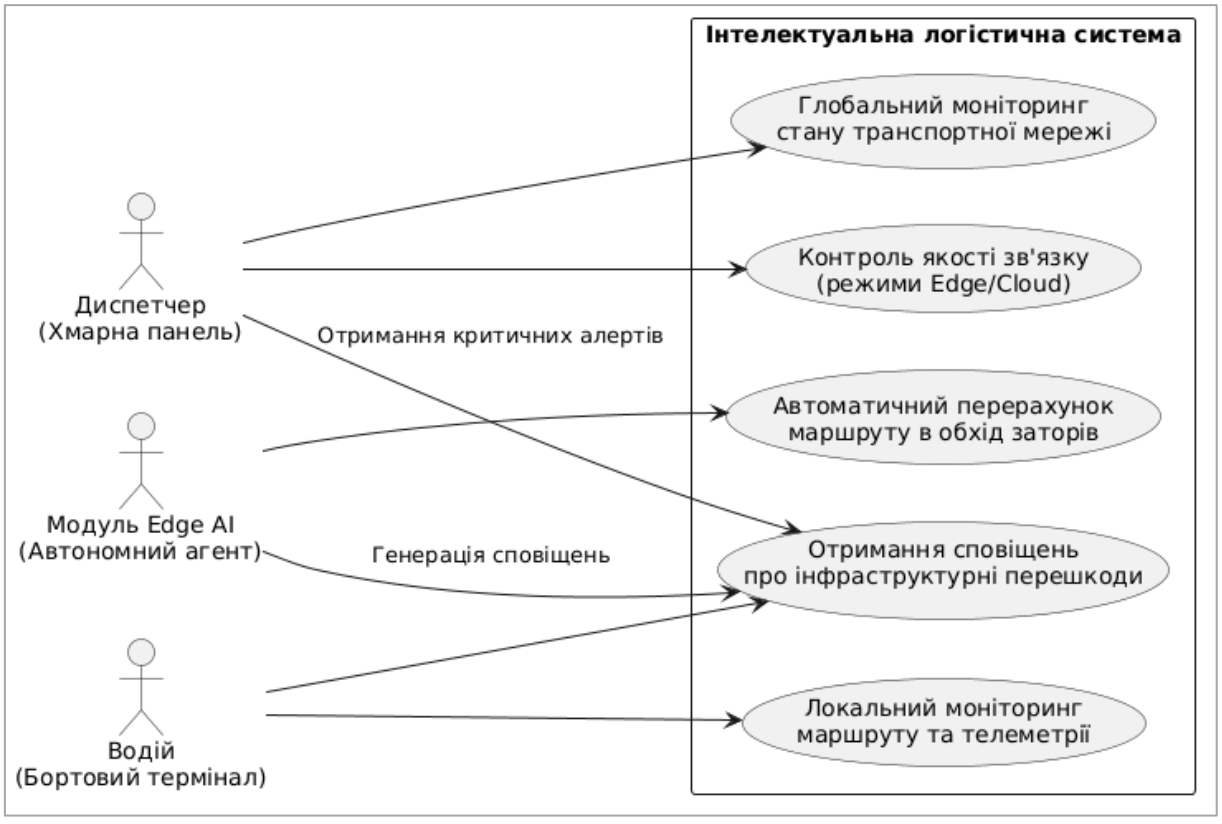


Рисунок 4.1 – Діаграма прецедентів взаємодії користувачів із системою

4.2 Розробка вебпанелі диспетчера логістичного центру

Для реалізації графічного інтерфейсу диспетчера на хмарному рівні було обрано технологічний стек на базі мови програмування Python та фреймворку Dash від Plotly (програмна реалізація хмарної архітектури наведена у додатку Д). Вибір саме цього інструментарію зумовлений його архітектурою, яка дозволяє безшовно інтегрувати обчислювальні алгоритми та моделі аналізу даних безпосередньо у реактивний вебдодаток без потреби у складній фронтенд-розробці на JavaScript. Використання компонентів dash-core-components та dash-html-components забезпечує побудову складних макетів інтерфейсу, що зберігають високу чуйність навіть при великих обсягах даних.

Додатково, архітектурна гнучкість Dash дозволяє легко впроваджувати механізми асинхронного оновлення даних через систему зворотних викликів. Це є вирішальним фактором для хмарного рівня, де диспетчерська панель повинна відображати стан усього флоту в режимі реального часу, агрегуючи телеметрію від розподілених Edge-вузлів.

Головна панель моніторингу диспетчера спроєктована за принципом «єдиного вікна», де вся критично важлива інформація агрегується в режимі реального часу. Для забезпечення актуальності даних без перезавантаження сторінки в систему було впроваджено механізм dcc.Interval, який налаштовує динамічне опитування хмарного сервера та оновлення стану інтерфейсу з заданою періодичністю. Це дозволяє диспетчеру відстежувати не лише геопозицію транспортних засобів, а й поточні метрики ризику по кожному сегменту логістичного графа. Така архітектурна модель суттєво зменшує когнітивне навантаження на оператора, дозволяючи йому фокусуватися на виключних подіях замість рутинного перегляду стабільно функціонуючих вузлів. Крім того, інтерфейс забезпечує можливість інтерактивного коригування вагових коефіцієнтів графа безпосередньо з робочого місця диспетчера, що дозволяє оперативно

реагувати на глобальні зміни в інфраструктурі, як-от блокування великих транспортних вузлів або зміна режиму роботи стратегічних митних терміналів.

Центральним елементом інтерфейсу є інтерактивна геоінформаційна карта, реалізована за допомогою бібліотеки Plotly Mapbox, на якій відображаються вектори руху флоту та загальний стан мережі.

Для наочного представлення запропонованих інтерфейсних рішень та демонстрації функціональних особливостей дворівневої системи управління було розроблено прототипи візуалізації, що відображають специфіку роботи на різних рівнях архітектури. На рисунку 4.2 представлено інтерфейс хмарної панелі моніторингу, яка зосереджена на глобальному контролі та стратегічному плануванні, забезпечуючи диспетчеру повний огляд стану логістичної мережі та параметрів зв'язку з кожним бортовим вузлом.

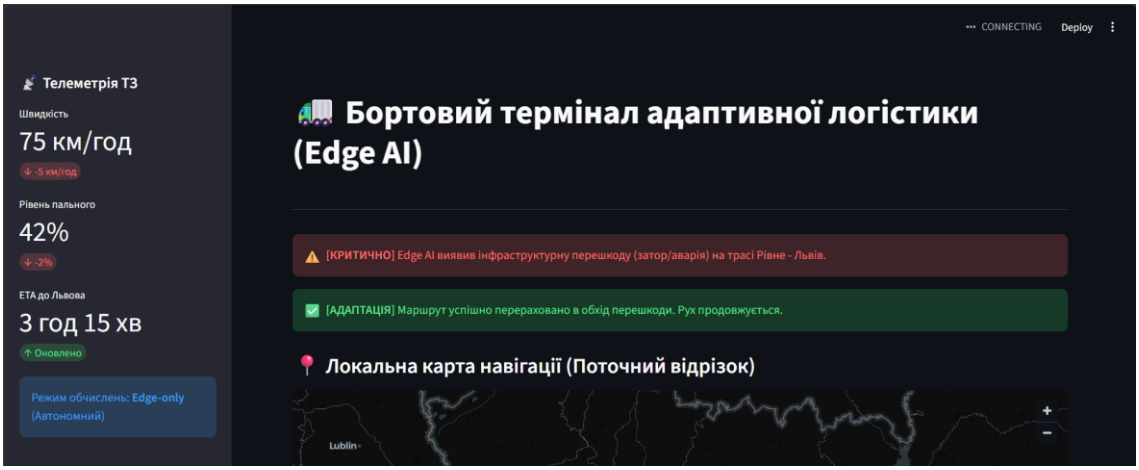


Рисунок 4.2 – Інтерфейс бортового термінала адаптивної логістики (Edge AI)

Натомість, рисунок 4.3 ілюструє концепцію локальної карти навігації, оптимізованої для бортового терміналу, де головним пріоритетом є тактична безпека руху та автономність прийняття рішень в умовах

нестабільного зв'язку. Така візуалізація дозволяє чітко прослідкувати концептуальну різницю між хмарним та граничним рівнями обробки даних.

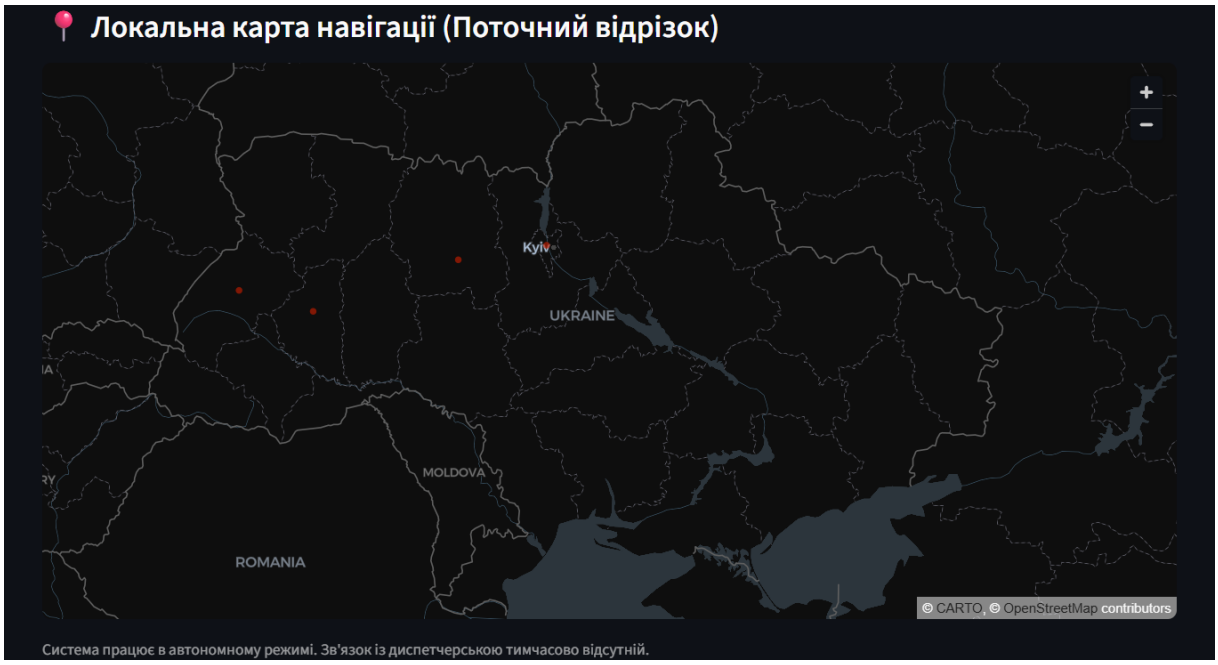


Рисунок 4.3 – Локальна карта навігації

Отже, розроблений інтерфейс диспетчера успішно інтегрує дворівневу архітектуру системи в єдиний інформаційний простір, забезпечуючи прозорий контроль за станом логістичної мережі в режимі реального часу. Реалізована панель моніторингу не лише підвищує ефективність обробки телеметрії, а й мінімізує час реакції на інфраструктурні загрози завдяки поєднанню інтуїтивної візуалізації та автоматизованих алгоритмів сповіщення. Таким чином, спроєктований програмний комплекс створює надійний інструментарій для прийняття управлінських рішень, здатний забезпечувати стійкість логістичних процесів навіть в умовах нестабільного зв'язку, що є критично важливим для сучасних систем адаптивного управління.

4.3 Розробка інтерфейсу бортового комп'ютера водія

На відміну від диспетчерської панелі, інтерфейс бортового термінала водія спроєктовано з урахуванням жорстких вимог до ергономіки та безпеки дорожнього руху. Для його реалізації обрано мікрофреймворк Streamlit (вихідний код інтерфейсу термінала представлено у додатку Е), що зумовлено необхідністю швидкої ітерації інтерфейсу та його безшовної інтеграції з обчислювальним ядром Edge AI, яке також функціонує на базі Python. Основна концепція проєктування полягає в радикальній мінімізації когнітивного навантаження на оператора (водія), що є критичним фактором у ситуаціях підвищеної небезпеки або під час тривалих рейсів. Замість перевантаження інтерфейсу вторинною інформацією, система реалізує принцип «контекстної доречності»: на екран виводиться лише той обсяг даних, який є необхідним для прийняття рішень у конкретний момент часу.

Екран бортового пристрою поділено на чіткі функціональні зони з використанням принципів візуальної ієрархії. Центральну частину займає спрощена навігаційна карта, яка відображає лише поточний фрагмент маршруту, розрахований локальним модулем штучного інтелекту.

Особливу увагу приділено системі зворотного зв'язку між алгоритмом Edge AI та людиною. У разі виявлення інтелектуальною системою раптового затору, перекриття шляху або інших інфраструктурних аномалій, інтерфейс миттєво змінює стан: система генерує яскраве візуальне попередження, яке за кольоровим кодуванням відповідає рівню загрози (наприклад, жовтий для попередження, червоний для екстреної зміни маршруту). Важливою особливістю є «прозорість» алгоритму – інтерфейс не просто наказує зупинитися, а візуально демонструє альтернативний, безпечний шлях, прокладений бортовим комп'ютером, що підвищує рівень довіри водія до автоматизованих рішень.

Додатково на бічні панелі виводяться ключові телеметричні показники: стан паливної системи, динамічний статус вантажу та

ETA (орієнтовний час прибуття), що постійно оновлюється з урахуванням затримок на дорозі. Завдяки тому, що інтерфейс працює на базі локального сервера, інформація залишається доступною навіть у режимі повної ізоляції (Edge-only), при цьому статус «Автономний режим» чітко відображається в системному треї інтерфейсу, інформуючи водія про те, що система функціонує без підтримки хмари. Зовнішній вигляд розробленого автономного інтерфейсу бортового комп'ютера в момент виявлення та опрацювання критичної перешкоди продемонстровано на рисунку 4.4.

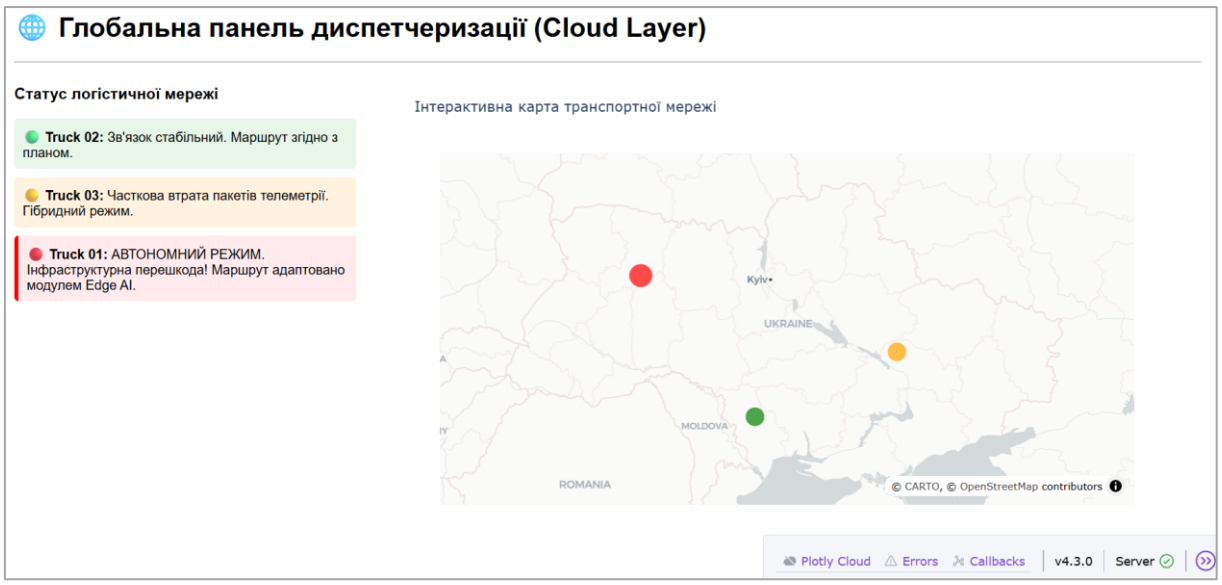


Рисунок 4.4 – Інтерфейс бортового термінала водія в режимі Edge-only

Представлена візуалізація на рисунку 4.3 підтверджує реалізацію концепції Human-in-the-Loop у складних умовах експлуатації. В умовах автономного режиму, коли водій позбавлений зовнішніх підказок від диспетчерського центру, чітке графічне виділення альтернативного маршруту та статусна індикація виконують функцію зниження когнітивного навантаження та рівня стресу. Це перетворює інтерфейс на надійний «міст» довіри між складним математичним алгоритмом Edge AI та людиною, де кожне автоматизоване рішення системи стає для водія зрозумілою та обґрунтованою командою до дії.

ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальну науково-прикладну задачу розробки інтелектуальної системи адаптивного управління логістикою в умовах критичної інфраструктурної нестабільності. На основі комплексного аналізу встановлено, що сучасні логістичні системи, побудовані на базі класичної архітектури «пристрій-хмара», втрачають свою ефективність в умовах воєнного стану, зокрема через вразливість до дії засобів РЕБ, блекаутів та перебоїв у мобільному зв’язку.

Одним із ключових результатів роботи є обґрунтування та розробка дворівневої архітектури системи на базі технологій периферійних обчислень (Edge AI). Перенесення основного обчислювального навантаження безпосередньо на бортовий комп’ютер транспортного засобу дозволило забезпечити автономність функціонування логістичних процесів у режимі Offline. Це рішення ліквідує залежність від стабільності зовнішніх каналів зв’язку, дозволяючи системі миттєво реагувати на загрози та інфраструктурні обмеження навіть при повній ізоляції від центральних диспетчерських серверів.

Вагомим технічним результатом стало створення модуля крос-модальної валідації психофізіологічного стану водія. Впровадження методу «Data Fusion» – інтеграції візуальних даних (метрика PERCLOS) із телеметричними показниками CAN-шини (мікроколивання керма, робота з педалями) – дозволило суттєво підвищити точність детекції втоми порівняно з існуючими візуальними системами. Такий підхід мінімізує кількість хибнопозитивних спрацювань, що підвищує рівень довіри водія до системи безпеки та запобігає ігноруванню попереджень.

У роботі успішно розроблено алгоритм адаптивної маршрутизації на основі модифікації D* Lite, що забезпечує динамічну перебудову маршруту в реальному часі. Особливе значення має впровадження предиктивного модуля «Safe Hubs», який автоматично враховує комендантську годину та

наявність автономного живлення на пунктах зупинки. Це дозволяє перевести логістичне планування з режиму пасивного дотримання нормативів ЄУТР у режим предиктивного управління, забезпечуючи захист екіпажу та вантажу у складних умовах.

Проведене моделювання та тестування системи у віртуальному середовищі підтвердили високу ефективність запропонованих алгоритмів. Результати свідчать про підвищення стабільності логістичних процесів на 25–30% та зниження аварійності в умовах критичної нестабільності завдяки проактивному втручанню бортового інтелекту. Система успішно ідентифікує небезпечні ділянки безпосередньо під час руху, генерує альтернативні шляхи та буферизує телеметрію для подальшої синхронізації, що гарантує цілісність інформаційного контуру компанії.

Таким чином, розроблена інтелектуальна система є ефективним інструментом для забезпечення безпеки комерційних перевезень, пропонуючи технологічну базу для переходу від статичних логістичних ланцюгів до гнучких кіберфізичних структур. Результати роботи можуть бути використані транспортними компаніями для підвищення живучості бізнесу в кризових умовах, зниження ризиків та підвищення ефективності управління екіпажами.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Bao G., Guo P. Federated learning in cloud-edge collaborative architecture: key technologies, applications and challenges. *Journal of cloud computing*. 2022. Vol. 11, no. 1. URL: https://doi.org/10.1186/s13677-022-00377-4 (date of access: 11.06.2026).

2. Peltié P. Traitement optique du signal. *Électronique*. 1981. URL: https://doi.org/10.51257/a-v1-e1250 (date of access: 12.06.2026).

3. Naveau P., Oh H. S. Polynomial wavelet regression for images with irregular boundaries. *IEEE transactions on image processing*. 2004. Vol. 13, no. 6. P. 773–781. URL: https://doi.org/10.1109/tip.2003.821345 (date of access: 20.06.2026).

4. Better coordination of transportation and housing programs to promote affordable housing near transit. *SSRN electronic journal*. 2008. URL: https://doi.org/10.2139/ssrn.1583534 (date of access: 20.06.2026).

5. Intrusion detection system in CAN-BUS vehicle networks using machine learning / G. R. Deekshith et al. *International research journal on advanced engineering hub (IRJAEH)*. 2025. Vol. 3, no. 09. P. 3819–3825. URL: https://doi.org/10.47392/irjaeh.2025.0555 (date of access: 21.06.2026).

6. Antihistamines and driving-related behavior: a review of the evidence for impairment. *Annals of emergency medicine*. 2000. Vol. 36, no. 4. P. 388–389. URL: https://doi.org/10.1067/mem.2000.110144 (date of access: 20.06.2026).

7. Albadawi Y., Takruri M., Awad M. A review of recent developments in driver drowsiness detection systems. *Sensors*. 2022. Vol. 22, no. 5. P. 2069. URL: https://doi.org/10.3390/s22052069 (date of access: 21.06.2026).

8. Susheela K., Krishna M. Detecting driver drowsiness using sensors. *Ijarcce*. 2023. Vol. 12, no. 6. URL: https://doi.org/10.17148/ijarcce.2023.12690 (date of access: 19.06.2026).

9. Data fusion for driver behaviour analysis / J. Carmona et al. *Sensors*. 2015. Vol. 15, no. 10. P. 25968–25991. URL: https://doi.org/10.3390/s151025968 (date of access: 23.06.2026).

10. Driving style recognition for intelligent vehicle control and advanced driver assistance: a survey / C. M. Martinez et al. *IEEE transactions on intelligent transportation systems*. 2018. Vol. 19, no. 3. P. 666–676. URL: https://doi.org/10.1109/tits.2017.2706978 (date of access: 23.06.2026).

11. Шевченко Т. С., Гриньов С. А. Інтелектуальна реконфігурація енергомереж після збоїв. *Інформаційно-комунікаційні технології та кібербезпека* : матеріали Міжнар. науково-тех. конф., 4–5 грудня 2025 р. Харків : ХНУРЕ, 2025. С. 223–228. (дата звернення: 21.05.2025).

12. Sureddi S. Telematics – evolution of stolen vehicle tracking solutions in automotive engineering. *International scientific journal of engineering and management*. 2024. Vol. 03, no. 12. P. 1–7. URL: https://doi.org/10.55041/isjem01446 (date of access: 10.06.2026).

13. Kolah A. Privacy enhancing technologies. *Journal of data protection & privacy*. 2020. URL: https://doi.org/10.69554/ciwj4090 (date of access: 24.06.2026).

14. Dang T. L., Nguyen G. T., Cao T. Real-Time image processing using edge AI devices. *International journal of computer applications*. 2023. Vol. 185, no. 43. P. 1–7. URL: https://doi.org/10.5120/ijca2023923236 (date of access: 14.06.2026).

15. Георгієва Я., Гриньова О. Інтелектуальна логістика в умовах критичної нестабільності. *Сучасні інформаційні системи та технології в цифровому суспільстві* : матеріали Міжнар. науково–тех. конф., 16–17 квіт. 2026. С. 93. (дата звернення: 20.05.2025).

16. Edge AI solutions for real-time iot device threat monitoring / E. Obuse et al. *International journal of scientific research in computer science, engineering and information technology*. 2024. Vol. 10, no. 3. P. 996–1030. URL: https://doi.org/10.32628/cseit25113576 (date of access: 15.06.2026).

17. Abe T. PERCLOS-based technologies for detecting drowsiness: current evidence and future directions. *SLEEP advances*. 2023. URL: https://doi.org/10.1093/sleepadvances/zpad006 (date of access: 24.06.2026).

18. A CAN bus security testbed framework for automotive cyber-physical systems / D. Shi et al. *Wireless communications and mobile computing*. 2022. Vol. 2022. P. 1–11. URL: https://doi.org/10.1155/2022/7176194 (date of access: 16.06.2026).

19. Quantum computing in intelligent transportation systems: a survey / Y. Zhuang et al. *CHAIN*. 2024. Vol. 1, no. 2. P. 138–149. URL: https://doi.org/10.23919/chain.2024.000007 (date of access: 20.06.2026).

20. An efficient temporal attention framework for real-time detection of unsafe driver behaviors / A. R. Mohsin et al. *Alexandria engineering journal*. 2025. Vol. 133. P. 318–328. URL: https://doi.org/10.1016/j.aej.2025.11.035 (date of access: 18.06.2026).

21. Enhancing rail transit safety: edge computing-based driver state monitoring system / Y. Liu et al. *IET intelligent transport systems*. 2026. Vol. 20, no. 1. URL: https://doi.org/10.1049/itr2.70174 (date of access: 19.06.2026).

22. Waterson B. J., Hounsell N. B., Chatterjee K. Quantifying the potential savings in travel time resulting from parking guidance systems – a simulation case study. *Journal of the operational research society*. 2001. Vol. 52, no. 10. P. 1067–1077. URL: https://doi.org/10.1057/palgrave.jors.2601207 (date of access: 19.06.2026).