

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
(повна назва)
Кафедра _____ Системотехніки _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти _____ другий (магістерський) _____
Моделювання процесів логістичних перевезень із використанням
методів машинного навчання _____
(тема)

Виконав:
здобувач 2 курсу, групи ІТПм-24-1 _____
Падалка Артем Борисович _____
(прізвище, власне ім'я, по батькові)
Спеціальність 122 Комп'ютерні науки _____
(код і повна назва спеціальності)
Тип програми освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)
Освітня програма Інформаційні технології
проектування _____
(повна назва освітньої програми)
Керівник проф. Ситніков Д.Е. _____
(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. кафедри _____ Гребеннік І.В. _____
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
Кафедра Системотехніки
Рівень вищої освіти другий (магістерський)
Спеціальність 122 Комп'ютерні науки
(код і повна назва)
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма Інформаційні технології проєктування
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)
«8» грудня 2025 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Падалка Артему Борисовичу
(прізвище, власне ім'я, по батькові)

- Тема роботи Моделювання процесів логістичних перевезень із використанням методів машинного навчання
затверджена наказом університету від «24» листопада 2025р. № 1058 Ст
- Термін подання студентом роботи до екзаменаційної комісії «24» грудня 2025 р.
- Вихідні дані до роботи історичні дані про виконання вантажних перевезень (GPS-треки, журнали замовлень), параметри автопарку, дорожня мережа; вхідні дані: параметри замовлень клієнтів, обмеження вантажопідйомності, погодні умови; ПК з 16 ГБ оперативної пам'яті та Windows 10/11 Pro, Python 3.9+, FastAPI, бібліотеки DEAP, XGBoost, TensorFlow, HTML/CSS/JavaScript (Leaflet.js).
- Перелік питань, що потрібно опрацювати в роботі 1. Вступ; 2. Аналіз предметної області; 2.1 Інформаційні системи управління логістикою: структура, класифікація та вимоги; 2.2 Аналіз функціональних можливостей аналогічних систем; 2.3 Визначення основних показників ефективності перевезень; 2.4 Постановка задачі дослідження; 3. Дослідження існуючих підходів та методів вирішення задач логістичного моделювання; 3.1 Застосування генетичних алгоритмів для задачі маршрутизації; 3.2 Прогнозування часу доставки методами градієнтного бустингу; 3.3 Аналіз та прогнозування попиту за допомогою рекурентних нейронних мереж; 4. Розробка та дослідження методів інтелектуального управління логістичними процесами; 4.1 Гібридний підхід та архітектура системи; 4.2 Розробка моделі прогнозування часу доставки; 4.3 Розробка моделі прогнозування попиту; 5. Порівняльний аналіз ефективності розроблених моделей; 5.1 Формування контрольної вибірки даних та розрахунок вхідних параметрів;

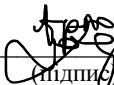
5.2 Розробка веб-застосунку СППР; 5.3 Результати моделювання прийняття рішень; 5.4 Аналіз чутливості моделей до зміни значень; 6. Висновки.

5. Скріншоти інтерфейсу розробленої системи (карти маршрутів, форми введення); Графіки навчання нейронної мережі (loss curves) та збіжності генетичного алгоритму; Діаграми порівняння точності моделей (MAE, RMSE); Скріншоти коду ключових алгоритмів; Таблиці з результатами сценарного моделювання перевезень.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Проведення аналізу предметної області	11.11.2025 – 14.11.2025	Виконано
2	Розробка методу комплексного оцінювання надійності техніки	14.11.2025 – 18.11.2025	Виконано
3	Розробка та аналіз методу на основі жорстких порогів	18.11.2025 – 24.11.2025	Виконано
4	Порівняльний аналіз ефективності розроблених моделей	24.11.2025 – 02.12.2025	Виконано
5	Захист кваліфікаційної роботи в екзаменаційній комісії	24.12.2025	Виконано

Дата видачі завдання « 24 » листопада 2025 р.

Здобувач _____

(підпис)

Керівник роботи _____
(підпис)

проф. Ситніков Д.Е.
(посада, власне ім'я, прізвище)

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

8.11.2025



Кваліфікаційна робота не містить відомостей заборонених до відкритого опублікування.

Керівник кваліфікаційної роботи _____ проф. Ситніков Д.Е.

Кваліфікаційна робота виконана у відповідності до стандартів, що діють в Україні.

Керівник кваліфікаційної роботи _____ проф. Ситніков Д.Е.

Попередній захист проведено 19 грудня 2025 р.

Керівник кваліфікаційної роботи _____ проф. Ситніков Д.Е.

РЕФЕРАТ

Кваліфікаційна робота: 94 с., 20 рис., 2 табл., 16 джерел, 2 додатки.

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ, ЛОГІСТИЧНІ ПЕРЕВЕЗЕННЯ, МАШИННЕ НАВЧАННЯ, ОПТИМІЗАЦІЯ МАРШРУТІВ, ПРОГНОЗУВАННЯ ПОПИТУ, НЕЙРОННА МЕРЕЖА, PYTHON, ВЕБ-ЗАСТОСУНОК, ТРАНСПОРТНА ЛОГІСТИКА.

Метою магістерської роботи є підвищення ефективності управління вантажними перевезеннями шляхом розробки та дослідження моделей машинного навчання для вирішення задач прогнозування термінів доставки та оптимізації маршрутів.

Методами дослідження є: системний аналіз, методи машинного навчання (регресійний аналіз, кластеризація), математичне моделювання та об'єктно-орієнтоване програмування.

Магістерська робота містить аналіз предметної області транспортної логістики, опис розроблених моделей на основі алгоритмів машинного навчання, програмну реалізацію модуля прогнозування, порівняльний аналіз точності різних алгоритмів, а також оцінку ефективності отриманих результатів.

Враховуючи динамічність ринку перевезень та зростання обсягів даних, впровадження інтелектуальних методів для автоматизації логістичних процесів є актуальною задачею. Розроблені моделі забезпечать перехід від статичного планування до динамічного прогнозування, що дозволить оптимізувати транспортні витрати, мінімізувати затримки та підвищити надійність ланцюгів постачання.

ABSTRACT

Master's thesis: 94 pp., 20 fig., 2 tabl., 16 sources, 2 attachments.

SUBJECT AREA ANALYSIS, LOGISTICS TRANSPORTATION, MACHINE LEARNING, ROUTE OPTIMIZATION, DEMAND FORECASTING, NEURAL NETWORK, PYTHON, WEB APPLICATION, TRANSPORT LOGISTICS.

The aim of the master's thesis is to improve the efficiency of freight transportation management by developing and researching machine learning models for solving the problems of delivery time forecasting and route optimization.

The research methods include: system analysis, machine learning methods (regression analysis, clustering), mathematical modeling, and object-oriented programming.

The master's thesis contains an analysis of the subject area of transport logistics, a description of the models developed based on machine learning algorithms, the software implementation of the forecasting module, a comparative analysis of the accuracy of different algorithms, and an assessment of the effectiveness of the results obtained.

Given the dynamic nature of the transportation market and the growth in data volumes, the implementation of intelligent methods for automating logistics processes is a pressing task. The developed models will ensure the transition from static planning to dynamic forecasting, which will optimize transportation costs, minimize delays, and increase the reliability of supply chains.

ЗМІСТ

Скорочення та умовні позначки.....	9
Вступ.....	10
1 Аналіз предметної області.....	12
1.1 Інформаційні системи управління логістикою: структура, класифікація та вимоги.....	12
1.2 Аналіз функціональних можливостей аналогічних систем.....	15
1.3 Визначення основних показників ефективності логістичних перевезень та метрик якості прогностичних моделей.....	17
1.4. Постановка задачі дослідження.....	19
2 Дослідження існуючих підходів та методів вирішення задач логістичного моделювання	21
2.1 Загальна характеристика обраних напрямків дослідження	21
2.2 Застосування генетичних алгоритмів для вирішення задачі маршрутизації (VRP).....	23
2.3 Прогнозування часу доставки методами градієнтного бустингу..	26
2.4 Аналіз та прогнозування попиту за допомогою рекурентних нейронних мереж (LSTM).....	28
3 Розробка та дослідження методів інтелектуального управління логістичними процесами	31
3.1 Загальна характеристика запропонованого гібридного підходу..	31
3.2 Розробка та аналіз методу на основі нечіткої моделі.....	36
3.3 Розробка моделі прогнозування часу доставки методами градієнтного бустингу	40
3.4 Розробка моделі прогнозування попиту на основі рекурентних нейронних мереж.....	43
4 Порівняльний аналіз ефективності розроблених моделей.....	47
4.1 Формування контрольної вибірки даних та розрахунків вхідних параметрів.....	47

4.2 Програмна реалізація вебзастосунку системи підтримки прийняття рішень	50
4.3 Оцінка результатів моделювання та аналіз точності прогнозів...	51
4.4 Дослідження чутливості моделей до зміни вхідних параметрів..	53
Висновки.....	55
Перелік джерел посилання.....	57
Додаток А Графічний матеріал.....	59
Додаток Б Використаний код.....	71

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД – база даних;

ПЗ – програмне забезпечення;

ШІ – штучний інтелект;

МН – машинне навчання (Machine Learning);

API – інтерфейс прикладного програмування;

TMS – Transport Management System (система управління транспортом);

VRP – Vehicle Routing Problem (задача маршрутизації транспортних засобів);

GPS – Global Positioning System (глобальна система позиціонування);

ETA – Estimated Time of Arrival (очікуваний час прибуття);

LSTM – Long Short-Term Memory (довга короткострокова пам'ять);

GBDT – Gradient Boosting Decision Trees (градієнтний бустинг над вирішальними деревами).

ВСТУП

Сучасний ринок транспортних послуг характеризується надзвичайною динамічністю, зростанням обсягів електронної комерції та підвищенням вимог клієнтів до швидкості доставки. Водночас, зі зростанням складності ланцюгів постачання, питання своєчасності, мінімізації витрат та оптимізації маршрутів стають критично важливими для логістичних операторів та бізнесу. У цьому контексті транспортні компанії та відділи логістики відіграють ключову роль. Вони є не лише виконавцями перевезень, але й генераторами цінних емпіричних даних про реальні маршрути, дорожній трафік, час завантаження та фактори затримок.

Щоденна діяльність транспортного підприємства генерує великі масиви даних: інформація про замовлення, GPS-треки автомобілів, історичні дані про час у дорозі, погодні умови та витрати пального. Однак у більшості існуючих бізнес-процесів цей масив даних використовується переважно для операційних завдань: моніторингу місцезнаходження транспорту, розрахунку з водіями та ведення звітності. Аналітичний потенціал цих даних для прогнозування та оптимізації залишається майже нереалізованим.

Рішення про вибір маршруту, призначення транспортного засобу чи прогнозування часу прибуття (ETA) часто приймаються логістами на основі суб'єктивного досвіду, інтуїції або статичних нормативів, які не враховують реальну дорожню ситуацію. Відсутність формалізованих, адаптивних моделей, здатних прогнозувати параметри перевезення на основі накопиченої історії, є значною проблемою.

Автоматизація збору даних через GPS-трекери та TMS-системи вже є стандартом, але наступний логічний крок, а саме автоматизація аналізу та перетворення цих даних на прогностичні моделі за допомогою машинного навчання – відсутня або розвинена недостатньо. Таким чином, розробка

методів та моделей, які б дозволили на основі історичної статистики розраховувати точні прогнози та інтегрувати їх в існуючі інформаційні системи, є актуальною науково-прикладною задачею. Це дозволить перейти від простої фіксації факту доставки до прогнозування ризиків затримок, оптимізації використання автопарку та надання клієнтам точної інформації.

Об'єктом дослідження є процес організації та управління вантажними перевезеннями в логістичних системах.

Предметом дослідження являються моделі та методи машинного навчання для вирішення задач прогнозування та оптимізації логістичних потоків.

Метою роботи є підвищення ефективності логістичних процесів шляхом розробки, програмної реалізації та аналізу моделей машинного навчання для прогнозування часу доставки та оптимізації маршрутів.

Як прикладну область для верифікації моделей та отримання вхідних даних буде використано знеособлені дані реальних вантажоперевезень транспортної компанії.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Інформаційні системи управління логістикою: структура, класифікація та вимоги

Інформаційна система (ІС) транспортної компанії, часто класифікована як TMS (Transport Management System), являє собою інтегрований програмно-апаратний комплекс, призначений для автоматизації, планування, виконання та контролю процесів перевезення вантажів[1]. Вона є цифровим ядром, що об'єднує всі ланки ланцюга постачання: прийом замовлень, диспетчеризацію, моніторинг транспорту, управління автопарком та взаємодію з водіями.

Структура типової логістичної ІС зазвичай будується за клієнт-серверною архітектурою або мікросервісною архітектурою і включає декілька основних компонентів.

Сервер бази даних (БД) – центральне сховище критичної інформації. Використовуються як реляційні СУБД для зберігання структурованих даних (замовлення, клієнти, автомобілі, фінансові транзакції), так і нереляційні (NoSQL) або Time-Series бази даних для зберігання масивів телеметричних даних (GPS-координати, швидкість, рівень пального), що надходять у реальному часі.

Сервер застосунку (Back-end) – компонент, що реалізує бізнес-логіку системи: алгоритми розподілу замовлень, розрахунок вартості перевезення, інтеграцію з картографічними сервісами та API зовнішніх партнерів.

Клієнтські інтерфейси – набір інструментів для різних ролей: веб-панель диспетчера/логіста для планування маршрутів та моніторингу, мобільний додаток водія для отримання завдань і підтвердження доставки, та особистий кабінет клієнта для відстеження вантажу.

Класифікація логістичних ІС може проводитись за кількома ознаками:

а) за моделлю розгортання: On-premise рішення, що встановлюються на власних серверах компанії (забезпечують повний контроль, але складні в

масштабуванні). Cloud (SaaS) рішення, хмарні платформи, що надають доступ через веб-браузер за підпискою (найбільш динамічний сегмент ринку, що дозволяє швидку інтеграцію нових модулів).

б) за функціональним призначенням: системи управління транспортом (TMS), системи управління складом (WMS), системи управління автопарком (FMS) або комплексні ERP-системи.

в) за рівнем інтелектуалізації:

– облікові системи, що фіксують лише факт виконання операцій;

– аналітичні системи, що надають інструменти для планування та звітності.

Детальний та всебічний аналіз типової архітектури сучасних інформаційних систем класу TMS (Transport Management Systems), що домінують на ринку, дозволяє виявити суттєву концептуальну та технологічну прогалину, яка стримує подальший розвиток ефективності логістичних операцій. Незважаючи на той факт, що більшість існуючих програмних рішень досягли високого рівня зрілості та ефективності у виконанні базових операційних завдань – таких як моніторинг місцезнаходження транспорту (tracking), ведення фінансового та управлінського обліку (accounting), а також автоматизація документообігу, – їхні прогностичні можливості часто залишаються на початковому рівні.

Функціонал стратегічного планування, прогнозування термінів доставки та динамічної оптимізації маршрутів у таких системах, як правило, базується на застарілих, спрощених статичних моделях або жорстко детермінованих алгоритмах. Ці підходи здебільшого оперують нормативними довідниками та усередненими статистичними показниками (наприклад, середня швидкість на ділянці дороги), повністю ігноруючи високу стохастичність та непередбачуваність реальних транспортних процесів. У переважній більшості комерційних продуктів спостерігається критична відсутність глибоко інтегрованих модулів машинного навчання (Machine Learning), які були б здатні розкрити потенціал накопичених даних

(«Big Data»).

Без застосування сучасних алгоритмів штучного інтелекту величезні масиви історичної інформації – про закономірності утворення заторів у різний час доби, вплив специфічних погодних умов на швидкість руху, реальний час простою під час завантажувально-розвантажувальних робіт на конкретних складах – залишаються фактично «мертвим вантажем». Система накопичує ці дані, але не використовує їх для побудови високоточних адаптивних моделей прогнозування часу прибуття (ETA – Estimated Time of Arrival). Це призводить до того, що логісти продовжують працювати в умовах невизначеності.

Саме тому розробка та впровадження інтелектуальних систем підтримки прийняття рішень (СППР) на базі технологій штучного інтелекту є не просто технологічним трендом, а критично важливою необхідністю для сучасної логістики. Це дозволить здійснити якісний перехід від традиційної реактивної моделі управління, де диспетчер змушений гасити "пожежі" та реагувати на проблеми постфактум, до проактивної моделі. Такий підхід дозволяє завчасно ідентифікувати потенційні ризики затримок, пропонувати альтернативні сценарії дій та оптимізувати ресурси ще до моменту виникнення критичної ситуації.

Вимоги до сучасної ІС управління перевезеннями поділяються на функціональні та нефункціональні. Функціональні вимоги:

- управління замовленнями: створення, редагування, консолідація вантажів;
- планування маршрутів: автоматична побудова маршруту з урахуванням точок доставки;
- моніторинг у реальному часі: відображення транспорту на карті, відстеження статусів;
- управління автопарком: облік ТО, страхування, контроль витрат пального;
- мобільна взаємодія: передача документів та статусів через додаток

водія;

- генерація звітності: аналітика по рентабельності рейсів, ефективності водіїв;

Нефункціональні вимоги:

- надійність та доступність: система повинна працювати 24/7, оскільки транспорт рухається цілодобово;

- масштабованість: здатність обробляти зростаючий потік GPS-даних без затримок;

- інтегрованість: наявність API для підключення розроблюваного модуля машинного навчання, який повинен мати доступ до історичних треків та поточних замовлень для проведення розрахунків;

- безпека даних: захист комерційної інформації про клієнтів та маршрути.

1.2 Аналіз функціональних можливостей аналогічних систем

На сучасному ринку програмного забезпечення для сфери транспортної логістики та управління ланцюгами постачання (Supply Chain Management) представлено значну кількість готових комерційних рішень, спрямованих на автоматизацію операційних бізнес-процесів. Найпоширенішим класом таких систем є TMS (Transportation Management Systems) [2], які адаптовані під потреби як великих логістичних провайдерів (3PL), так і внутрішніх транспортних відділів виробничих підприємств. Ці системи допомагають організаціям керувати заявками на перевезення, контролювати витрати та автоматизувати документообіг.

До типових представників лідерів ринку у цьому сегменті можна віднести такі комплексні продукти, як SAP Transportation Management [3], Oracle Logistics Cloud [4] та більш спеціалізовані платформи маршрутизації, наприклад, Route4Me або Samsara [5]. Їхній базовий функціонал здебільшого покриває стандартні операційні потреби логістичного

диспетчера:

- ведення реєстру заявок на перевезення та бази даних контрагентів;
- візуалізація місцезнаходження транспортних засобів на карті в реальному часі (GPS-моніторинг);
- базове планування маршрутів на основі відстані та розрахунок планових витрат пального;
- контроль статусів виконання доставки та управління парком автомобілів (облік ремонтів, ТО);
- генерація типової звітності про виконані рейси та кілометраж.

Однак, незважаючи на широкий спектр заявлених маркетингових можливостей та, часто, досить високу вартість ліцензування та впровадження, глибокий детальний аналіз архітектури та алгоритмічної бази цих систем виявляє суттєвий, фундаментальний недолік. Більшість із них, по суті, залишаються високотехнологічними інструментами фіксації фактів – своєрідними цифровими журналами обліку, а не інструментами передбачення майбутнього. Вони чудово справляються з роллю ретроспективного аналізатора, фіксуючи те, що вже відбулося, але виявляються безсилими перед завданням зазирнути наперед.

Жодна з масових "коробкових" TMS-систем, як правило, не має вбудованих, глибоко інтегрованих інструментів машинного навчання, здатних проводити предиктивний аналіз складних часових рядів. Вони не вміють виявляти приховані закономірності в історичних даних та оцінювати ризики затримок на основі нелінійних патернів, які формуються роками. Розробники таких систем часто ігнорують той факт, що логістика – це не статична, а динамічна система з високим рівнем ентропії, де минулі успіхи не гарантують майбутніх результатів без урахування мінливих контекстних факторів.

Такі системи не є універсальними рішеннями для інтелектуального управління логістикою в реальних умовах невизначеності. Їхня внутрішня логіка орієнтована виключно на автоматизацію рутинних дій диспетчера,

дотримання формальних процедур документообігу та контроль виконання нормативів. Можливості аналітики в них, як правило, обмежуються рівнем описової статистики (Descriptive Analytics). Користувач бачить красиві дашборди, які відповідають на питання: "скільки ми проїхали вчора", "яка середня витрата пального за місяць" або "хто з водіїв зробив найбільше рейсів". Це корисна інформація для фінансового звіту, але вона не несе управлінської цінності для прийняття оперативних рішень "тут і зараз".

Критичною проблемою є те, що ці системи використовують детерміновані алгоритми маршрутизації. Розрахунок очікуваного часу прибуття (ETA) в них базується на "ідеальних" лабораторних умовах або статичних обмеженнях швидкості, прописаних у дорожніх картах. Вони оперують поняттям "середньої швидкості", повністю ігноруючи динамічні фактори та стохастичну природу дорожнього руху. Система не "розуміє", що 60 км/год вночі і 60 км/год в годину пік – це абсолютно різні сценарії.

В результаті, існуюче ПЗ не дає відповіді на складні, але життєво необхідні прогностичні питання. Воно мовчить, коли потрібно оцінити ймовірності: «Яка статистична ймовірність того, що вантаж запізниться, якщо виїхати у п'ятницю ввечері, коли трафік на виїздах з міста зростає експоненційно?», «Як саме вплине прогноз погоди (наприклад, сильний снігопад або ожеледиця) на середню швидкість руху вантажівки на конкретній ділянці гірської траси?», або «Чи варто призначати саме цього водія на цей складний маршрут, враховуючи його індивідуальний стиль водіння та історію попередніх запізнень?». Відсутність відповідей на ці питання змушує логістів покладатися на власну інтуїцію, що в епоху Big Data є неприпустимим марнуванням ресурсів.

1.3 Визначення основних показників ефективності логістичних перевезень та метрик якості прогностичних моделей

Процес прийняття рішень у сфері транспортної логістики є складною

багатофакторною задачею, яка вимагає відмови від інтуїтивного управління на користь чітко формалізованих критеріїв ефективності. Для побудови інтелектуальної системи, здатної оптимізувати маршрути та прогнозувати параметри доставки, необхідно насамперед визначити ключові показники, які підлягають вимірюванню, мінімізації або максимізації. У контексті предметної області дослідження ці показники можна розділити на часові, економічні та якісні, які в сукупності формують цільову функцію для алгоритмів оптимізації та метрики якості для моделей машинного навчання.

Першочерговим фактором, що визначає якість логістичного сервісу з точки зору клієнта, є часові параметри, зокрема точність дотримання прогнозованого часу прибуття (Estimated Time of Arrival). Цей показник є критичним, оскільки сучасні ланцюги постачання часто працюють за принципами мінімізації складських запасів, тому будь-яка затримка може призвести до зупинки виробничих процесів або штрафних санкцій. У рамках дослідження оцінюється не лише фактичний час транспортування, але й відхилення реального часу прибуття від запланованого, що дозволяє оцінити надійність перевезення. На формування цього показника впливає низка стохастичних факторів, таких як погодні умови, час доби, день тижня та інтенсивність дорожнього руху, які необхідно враховувати при побудові прогностичних моделей.

З економічної точки зору, пріоритетним завданням для логістичного оператора є мінімізація експлуатаційних витрат, які прямо корелюють із загальною довжиною маршрутів та кількістю задіяного транспорту. Сумарний пробіг автопарку виступає одним з головних критеріїв оптимізації при вирішенні задачі маршрутизації транспортних засобів (VRP). Зменшення загальної відстані, яку долають автомобілі для обслуговування фіксованої множини замовлень, безпосередньо призводить до зниження витрат на паливно-мастильні матеріали та амортизацію техніки. Додатковим економічним індикатором є коефіцієнт використання вантажопідйомності транспортного засобу, оскільки ефективна логістика

передбачає максимізацію корисного завантаження та мінімізацію так званих «холостих» пробігів, коли транспорт рухається порожнім.

Для оцінки якості роботи розроблених інтелектуальних моделей необхідно використовувати специфічні статистичні метрики, що дозволяють кількісно виміряти точність прогнозів. У задачах регресії, до яких належить прогнозування часу доставки та попиту, ключовими показниками є середня абсолютна помилка (MAE) та середньоквадратична помилка (RMSE). Ці метрики показують, на скільки хвилин або тонн у середньому помиляється модель порівняно з реальними історичними даними. Коефіцієнт детермінації (R^2) дозволяє оцінити, яку частку дисперсії залежної змінної пояснює побудована модель, що є індикатором її адекватності та здатності узагальнювати виявлені закономірності. Таким чином, комплексна оцінка ефективності системи базується на балансі між мінімізацією витрат ресурсів (відстань, час) та максимізацією точності прогнозів.

1.4. Постановка задачі дослідження

На основі проведеного аналізу предметної області транспортної логістики та існуючих інформаційних систем управління перевезеннями можна сформулювати ключову науково-прикладну проблему, яка потребує вирішення. Сучасні логістичні компанії генерують та накопичують значні обсяги даних про виконання рейсів, параметри вантажів та рух транспорту, проте процес прийняття управлінських рішень щодо планування маршрутів та оцінки термінів доставки часто залишається консервативним. Більшість існуючих систем базуються на детермінованих алгоритмах та статичних нормативах середньої швидкості, ігноруючи динамічну та стохастичну природу дорожнього руху. Це призводить до неоптимального використання автопарку, неточних прогнозів часу прибуття що в умовах високої конкуренції та зростаючих вимог клієнтів є критичним недоліком.

Метою дослідження є підвищення ефективності управління вантажними перевезеннями шляхом розробки та програмної реалізації комплексної інформаційної системи, яка поєднує методи комбінаторної оптимізації для автоматичної побудови маршрутів та сучасні алгоритми машинного навчання для високоточного прогнозування часових параметрів і попиту. В якості об'єкта управління розглядається процес організації вантажопотоків, де вхідними даними виступають множина замовлень з географічними координатами та часовими обмеженнями, характеристики наявного автопарку, а також історичні дані про виконані перевезення, що включають інформацію про вплив зовнішніх факторів, таких як погода та трафік.

Для досягнення поставленої мети необхідно виконати низку взаємопов'язаних завдань. Насамперед потрібно провести аналіз існуючих методів вирішення задачі маршрутизації транспортних засобів та підходів до прогнозування часових рядів, обґрунтувавши вибір генетичних алгоритмів для оптимізації та ансамблевих методів і рекурентних нейронних мереж для прогнозування. Наступним кроком є розробка математичних моделей та алгоритмів, зокрема адаптація генетичного алгоритму для врахування обмежень вантажопідйомності, побудова регресійної моделі на базі градієнтного бустингу для розрахунку очікуваного часу прибуття та створення нейромережевої моделі LSTM для прогнозування майбутнього попиту на перевезення. Важливою частиною роботи є програмна реалізація розроблених моделей у вигляді веб-застосунку з мікросервісною архітектурою, що дозволить інтегрувати їх у єдине інформаційне середовище. Фінальним етапом є проведення експериментальних досліджень на тестових наборах даних для оцінки точності моделей за обраними метриками та порівняльного аналізу ефективності запропонованого підходу з традиційними методами планування, що дозволить сформулювати практичні рекомендації щодо впровадження системи.

2 ДОСЛІДЖЕННЯ ІСНУЮЧИХ ПІДХОДІВ ТА МЕТОДІВ ВИРІШЕННЯ ЗАДАЧ ЛОГІСТИЧНОГО МОДЕЛЮВАННЯ

2.1 Загальна характеристика обраних напрямків

Рішення задачі оптимальної маршрутизації та точного прогнозування часу доставки є класичною проблемою транспортної логістики та операційного менеджменту. Це складне, слабоструктуроване завдання, що вимагає врахування багатфакторної, часто суперечливої інформації: від дорожнього трафіку та погодних умов до специфічних вікон доставки клієнта. Прийняття такого рішення логістом або диспетчером часто є суб'єктивним і базується на особистому досвіді або статичних нормативах, що в умовах динамічного ринку не гарантує оптимального результату. Це підкреслює актуальність розробки інформаційних систем підтримки прийняття рішень, здатних надати об'єктивну та математично обґрунтовану рекомендацію на основі реальних даних.

Однією з основних задач, що вимагає інтелектуальної підтримки, є прогнозування часу прибуття (ETA) та вибір оптимального маршруту в умовах невизначеності. Для її вирішення теоретично можуть застосовуватись моделі різної складності: від простих детермінованих алгоритмів на графах до складних ансамблевих моделей машинного навчання. Проблемою є те, що в рамках існуючих TMS-систем перевага часто надається спрощеним моделям, а потенціал накопичених історичних даних ігнорується.

Актуальною науково-прикладною задачею є розробка програмної системи, яка б дозволяла не лише моніторити перевезення, але й реалізовувала б кілька прогностичних моделей з метою їх порівняльного аналізу на реальних даних. Метою магістерської роботи є підвищення якості та обґрунтованості управлінських рішень у сфері вантажних перевезень шляхом розробки та дослідження моделей машинного навчання, що дозволяють оптимізувати маршрути та підвищити точність прогнозування

термінів доставки.

Для розробки ефективної інформаційної системи управління вантажними перевезеннями та впровадження модуля підтримки прийняття рішень необхідно спиратися на фундаментальні та прикладні дослідження в галузі транспортної логістики та штучного інтелекту. Специфіка предметної області вимагає поєднання методів комбінаторної оптимізації (для побудови маршрутів) та прогнозного моделювання (для розрахунку часу доставки та попиту).

У межах виконання передатестаційної практики було відібрано та проаналізовано три ключові групи наукових підходів, які формують теоретичний базис магістерського дослідження. Ці напрямки репрезентують різні аспекти управління ланцюгами постачання і дозволяють сформулювати комплексну архітектуру майбутньої системи.

Перший напрямок стосується вирішення задачі маршрутизації транспорту (Vehicle Routing Problem – VRP) та її динамічних варіацій. Оскільки класичні методи (наприклад, метод найближчого сусіда або алгоритм Кларка-Райта) часто не справляються з розмірністю реальних задач, основна увага приділяється евристичним та метаевристичним алгоритмам, зокрема генетичним алгоритмам та мурашиним колоніям, які дозволяють знаходити субоптимальні рішення за прийнятний час.

Другий напрямок фокусується на прогнозуванні часу прибуття (Estimated Time of Arrival – ETA) з використанням методів машинного навчання на основі ансамблевих моделей (Gradient Boosting). Традиційні TMS-системи розраховують час на основі середньої швидкості, ігноруючи стохастичні фактори (погоду, час доби, тип вантажу). Дослідження сучасних публікацій показує, що застосування алгоритмів типу XGBoost або LightGBM дозволяє значно зменшити похибку прогнозування завдяки врахуванню нелінійних залежностей.

Третій напрямок охоплює застосування глибокого навчання (Deep Learning), зокрема рекурентних нейронних мереж (LSTM/GRU), для аналізу

часових рядів логістичних даних. Це критично важливо для прогнозування попиту на перевезення та завантаженості складських терміналів. Нейромеревеві підходи дозволяють виявляти приховані сезонні патерни та довгострокові залежності, які неможливо ідентифікувати класичними статистичними методами на кшталт ARIMA.

Така декомпозиція задачі на три складові – маршрутизацію, прогнозування часу та аналіз попиту – дозволяє структурувати дослідження та обрати найбільш релевантні математичні інструменти для кожної підсистеми.

2.2 Застосування генетичних алгоритмів для вирішення задачі маршрутизації (VRP)

Задача маршрутизації транспортних засобів (VRP) є NP-складною задачею комбінаторної оптимізації, яка полягає у пошуку оптимального набору маршрутів для парку транспортних засобів, що мають обслужити множину клієнтів[6]. У контексті розроблюваної системи, VRP ускладнюється додатковими обмеженнями: часовими вікнами доставки (Time Windows), обмеженою вантажопідйомністю автомобілів та необхідністю мінімізації не лише відстані, а й сумарного часу в дорозі.

Аналіз наукових джерел вказує на те, що одним з найефективніших методів вирішення подібних задач є генетичні алгоритми (Genetic Algorithms – GA). Цей підхід базується на принципах еволюційної біології: природному відборі, схрещуванні (crossover) та мутації[7].

Формально процес оптимізації можна описати наступним чином. Кожен потенційний розв'язок (набір маршрутів) кодується у вигляді хромосоми – послідовності чисел, що відповідають ідентифікаторам точок доставки. Популяція складається з N таких хромосом.

Цільова функція (Fitness Function) $F(x)$ яку необхідно мінімізувати, зазвичай є зваженою сумою загальної довжини маршруту L та штрафів за

порушення часових обмежень P :

$$F(x) = \alpha \cdot \sum_{k=1}^K L_k + \beta \cdot \sum_{i=1}^M P_i \quad (2.1)$$

де K – коефіцієнт вартості володіння;

M – кількість клієнтів;

α та β – вагові коефіцієнти.

У проаналізованих роботах підкреслюється важливість правильного налаштування операторів мутації. Для логістичних задач проста точкова мутація є неефективною, оскільки вона часто створює невалідні маршрути. Натомість застосовуються спеціалізовані оператори, такі як 2-opt (обмін двох ребер місцями) або переміщення підмаршруту, що дозволяє локально покращувати рішення, уникаючи застрягання в локальних мінімумах.

У детально проаналізованих наукових джерелах особливий акцент робиться на критичній важливості тонкого налаштування та вибору операторів мутації, оскільки саме вони відповідають за диверсифікацію пошуку та здатність алгоритму виходити за межі вже знайдених рішень. Для специфічних логістичних задач, які характеризуються жорсткими топологічними обмеженнями, класична «сліпа» точкова мутація (випадкова зміна одного елемента послідовності) виявляється вкрай неефективною та часто контрпродуктивною. Вона з високою ймовірністю генерує так звані «летальні» хромосоми або невалідні маршрути – рішення, які фізично неможливо виконати на практиці (наприклад, виникають розриви в логістичному ланцюгу, дублювання відвідування одного й того ж клієнта або, навпаки, пропуск обов'язкової точки доставки). Це призводить до марної витрати обчислювальних ресурсів процесора на перевірку та відбраковування завідомо хибних варіантів, що сповільнює загальну роботу системи.

Натомість, сучасні підходи до вирішення VRP передбачають застосування спеціалізованих, евристично обґрунтованих операторів, таких

як 2-opt (інверсія підмаршруту або обмін двох ребер графа місцями) або переміщення цілих блоків маршруту між різними транспортними засобами. Ці методи дозволяють здійснювати локальне покращення рішення, фактично «розплутуючи» неоптимальні петлі на графі маршруту, та ефективно підтримувати генетичну різноманітність популяції. Такий підхід є єдиним дієвим механізмом уникнення проблеми передчасної збіжності алгоритму (premature convergence), коли пошук "застрягає" в локальних мінімумах, які є далекими від глобального оптимуму, але з яких неможливо вийти за допомогою дрібних змін.

Визначальною перевагою та ключовим аргументом на користь вибору генетичних алгоритмів у контексті даної магістерської роботи є їхня виняткова архітектурна гнучкість та високий адаптивний потенціал. На відміну від класичних детермінованих методів або моделей лінійного програмування, де будь-яка зміна початкових умов часто вимагає повної перебудови системи рівнянь та логіки рішення, GA дозволяє інтегрувати нові, навіть нелінійні обмеження, шляхом відносно простої модифікації цільової функції (функції пристосованості) або додавання штрафних коефіцієнтів. Це дозволяє розробнику легко адаптувати математичну модель до змінних реалій бізнесу та специфічних вимог замовника: наприклад, додавати параметри різнотипного парку автомобілів (heterogeneous fleet) з різною вантажопідйомністю, враховувати законодавчі вимоги щодо перерв на відпочинок водіїв, або специфічні умови сумісності вантажів без необхідності докорінної зміни ядра програмного алгоритму.

Крім того, еволюційні алгоритми за своєю природою ідеально піддаються розпаралелюванню на рівні обчислювальних потоків, оскільки оцінка пристосованості (fitness evaluation) кожної окремої особини в популяції є незалежною операцією. Цей технологічний аспект є критично важливим для забезпечення високої швидкодії та масштабованості серверної частини розроблюваної системи, особливо в умовах необхідності оперативної обробки великих масивів вхідних замовлень. Це дозволяє

реалізувати прийняття рішень у режимі, наближеному до реального часу (near real-time), що є однією з ключових нефункціональних вимог до сучасних хмарних TMS-рішень.

2.3 Прогнозування часу доставки методами градієнтного бустингу

Точне та надійне прогнозування часу прибуття (Estimated Time of Arrival – ETA) на сьогоднішній день залишається однією з найбільш нетривіальних та багатогранних задач у сучасній транспортній логістиці. Складність цієї проблеми зумовлена високим ступенем ентропії та невизначеності дорожнього середовища, яке є відкритою динамічною системою. На відміну від ідеалізованих лабораторних умов, реальний процес перевезення піддається впливу величезної кількості стохастичних факторів, які важко формалізувати.

Традиційні картографічні та навігаційні сервіси, орієнтовані на масового користувача, часто надають надмірно оптимістичні прогнози, які є малоприсадибними для професійної логістики. Вони, як правило, базуються на усереднених швидкісних профілях легкових автомобілів і систематично не враховують специфіку експлуатації великовантажного транспорту. До таких специфічних факторів відносяться законодавчі обмеження швидкості для вантажівок, значно менша маневреність, необхідність дотримання жорсткого режиму праці та відпочинку водіїв (згідно з показаннями тахографів), а також значні часові витрати на паркування, маневрування у промислових зонах та очікування в чергах на складських рампах. Ігнорування цих нюансів призводить до систематичних запізнь, зривів вікон доставки та штрафних санкцій для перевізника.

Глибокий аналіз сучасних науково-практичних досліджень у галузі Data Science переконливо демонструє, що для задач регресії на структурованих табличних даних (tabular data), до яких безпосередньо відносяться історичні логи перевезень, нейронні мережі часто поступаються

у ефективності іншим підходам. Натомість, беззаперечним лідером за співвідношенням точності, швидкодії та інтерпретованості результатів є алгоритми градієнтного бустингу над вирішальними деревами (Gradient Boosting Decision Trees – GBDT)[8]. Саме тому в межах даної магістерської роботи як основний інструментарій розглядаються та порівнюються сучасні високоефективні реалізації цього методу: бібліотеки XGBoost[9], LightGBM та CatBoost.

Фундаментальна сутність методу градієнтного бустингу полягає у стратегії побудови потужного прогностичного ансамблю шляхом послідовної, ітеративної комбінації великої кількості простих базових алгоритмів, так званих «слабких моделей» (weak learners), у ролі яких зазвичай виступають неглибокі дерева рішень. Ключова ідея полягає не в тому, щоб навчити одну складну модель одразу, а в тому, щоб кожна наступна модель у ланцюжку намагалася виправити помилки, допущені попередньою композицією алгоритмів. Фактично, кожен новий алгоритм навчається передбачати антиградієнт функції втрат, тобто, спрощено кажучи, навчається на «залишках» (residuals) попередніх кроків.

Математично цей процес можна виразити як побудову адитивної моделі, де фінальний прогноз є зваженою сумою прогнозів усіх дерев ансамблю. Такий підхід дозволяє моделі поступово знижувати зміщення (bias) та дисперсію (variance) прогнозу, ефективно виявляючи складні нелінійні залежності між вхідними параметрами (наприклад, погодою, часом доби, завантаженістю трафіку) та цільовою змінною – часом доставки, що робить його ідеальним інструментом для вирішення поставленої задачі.

Суть методу полягає у побудові ансамблю слабких моделей (дерева рішень), де кожна наступна модель намагається виправити помилки попередніх. Математично це можна виразити як адитивну модель:

$$\hat{y}_i = \alpha \cdot \sum_{k=1}^K f_k(x_i), f_k \in \mathcal{F} \quad (2.3)$$

де \hat{y}_i – прогнозований час;

x_i – вектор ознак рейсу;

f_k – окреме дерево рішень.

Навчання відбувається шляхом мінімізації функції втрат (наприклад, середньоквадратичної помилки MSE або абсолютної помилки MAE) з додаванням регуляризації для запобігання перенавчанню.

Ключовим етапом, виділеним у проаналізованих джерелах, є конструювання ознак (Feature Engineering). Для логістичної моделі вхідний вектор X повинен включати:

- часові ознаки: година доби, день тижня, місяць, наявність свят;
- геопросторові ознаки: координати старту та фінішу, категорія дороги, відстань;
- контекстні ознаки: прогноз погоди (опади, температура), історична завантаженість трафіку на ділянці;
- специфічні ознаки вантажу: вага, тип кузова, необхідність спеціальних умов.

Важливою перевагою використання алгоритмів на кшталт CatBoost є їх здатність ефективно працювати з категоріальними ознаками (наприклад, ID водія, марка автомобіля, тип вантажу) без необхідності складного попереднього кодування (One-Hot Encoding), що спрощує підготовку даних. Результати досліджень свідчать, що впровадження ML-моделей дозволяє знизити середню абсолютну помилку (MAE) прогнозу ETA на 15-20% порівняно з детермінованими розрахунками навігаторів.

2.4 Аналіз та прогнозування попиту за допомогою рекурентних нейронних мереж (LSTM)

Ефективне управління автопарком неможливе без розуміння майбутнього попиту. Логістичні дані є класичним прикладом часових рядів,

які характеризуються наявністю трендів, сезонності (тижневої, річної) та випадкового шуму. Класичні методи авторегресії (ARIMA, Exponential Smoothing)[12] добре працюють з лінійними залежностями, але часто не здатні охопити складні нелінійні патерни, характерні для сучасних ланцюгів постачання.

Третій розглянутий напрямок досліджень фокусується на застосуванні глибокого навчання, а саме мереж довгої короткострокової пам'яті (Long Short-Term Memory – LSTM)[10]. Архітектура LSTM розроблена спеціально для вирішення проблеми зникаючого градієнта, притаманної звичайним рекурентним мережам (RNN), що дозволяє їм запам'ятовувати інформацію на довгих часових проміжках.

Комірка LSTM містить три основні вентиля (gates): вхідний, вихідний та вентиль забування. Це дозволяє мережі вирішувати, яку інформацію з попередніх кроків (наприклад, сплеск замовлень перед святами минулого року) варто зберегти, а яку можна ігнорувати.

Математична модель оновлення стану комірки описується системою рівнянь:

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ C_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\ C_t &= f_t * C_{t-1} + i_t * C_t \end{aligned}$$

де x_t – вхідний вектор на кроці t (наприклад, кількість замовлень сьогодні);

h_{t-1} – прихований стан з попереднього кроку;

C_t – стан комірки пам'яті.

У контексті проведеного дослідження застосування рекурентних нейронних мереж архітектури LSTM визначено як найбільш доцільний та обґрунтований підхід для вирішення критично важливої задачі прогнозування обсягів вантажних перевезень на конкретних логістичних

напрямах. Вибір саме цього інструментарію відкриває фундаментальні можливості для практичної реалізації стратегії предиктивного розміщення транспортних ресурсів, що є ключем до оптимізації сучасних ланцюгів постачання.

Сутність цієї стратегії полягає у переході від реактивної моделі управління до проактивної: система дозволяє завчасно планувати передислокацію вільних транспортних засобів у ті географічні кластери, де згідно з прогнозом очікується підвищення попиту, ще до моменту надходження фактичних замовлень від клієнтів. Такий підхід дозволяє нівелювати дисбаланс між попитом та пропозицією транспорту в окремих регіонах, суттєво скорочуючи час очікування на завантаження та мінімізуючи витрати на так звані «холості пробіги», коли порожній автомобіль змушений долати великі відстані до точки замовлення.

Детальний аналіз результатів експериментальних перевірок, викладених у профільних наукових джерелах, переконливо підтверджує перевагу методів глибокого навчання. Зокрема, доведено, що моделі на базі LSTM стабільно перевершують класичні статистичні методи (такі як ARIMA чи експоненційне згладжування) за метрикою середньоквадратичної помилки (RMSE – Root Mean Square Error). Ця перевага стає особливо відчутною на складних, високоволатильних наборах даних[11], які характеризуються нелінійною динамікою, раптовими сплесками активності та наявністю нетривіальних довгострокових залежностей, що є типовою картиною для реальних даних транспортної логістики.

3 РОЗРОБКА ТА ДОСЛІДЖЕННЯ МЕТОДІВ ІНТЕЛЕКТУАЛЬНОГО УПРАВЛІННЯ ЛОГІСТИЧНИМИ ПРОЦЕСАМИ

3.1 Загальна характеристика запропонованого гібридного підходу

Для побудови ефективної інформаційної системи управління вантажними перевезеннями необхідно відійти від використання розрізнених інструментів та перейти до концепції єдиної інтелектуальної платформи. У рамках даної роботи пропонується гібридний підхід, який концептуально об'єднує два фундаментально різні класи алгоритмів: методи комбінаторної оптимізації для планування ресурсів та методи машинного навчання для зниження невизначеності зовнішнього середовища.[13] Такий симбіоз дозволяє вирішити ключову проблему класичних TMS-систем — розрив між плановими показниками, розрахованими на основі статичних нормативів, та реальними умовами виконання логістичних операцій.

```
def _evaluate(self, individual: List[int]) -> Tuple[float]: 1 usage
    """
    Фітнес-функція: мінімізація загальної відстані

    Args:
        individual: хромосома

    Returns:
        Кортеж з одним значенням (загальна відстань)
    """
    routes = self._decode_individual(individual)

    # Перевірка обмежень
    penalty = 0.0

    # Перевірка кількості транспортних засобів
    if len(routes) > self.n_vehicles:
        penalty += 10000 * (len(routes) - self.n_vehicles)

    # Перевірка місткості для кожного маршруту
    for route in routes:
        route_load = sum(self.customers[idx - 1].get(key: 'demand', default: 1.0) for idx in route)
        if route_load > self.vehicle_capacity:
            penalty += 10000 * (route_load - self.vehicle_capacity)

    # Обчислимо загальну відстань
    total_distance = sum(self._calculate_route_distance(route) for route in routes)

    return (total_distance + penalty,)
```

Рисунок 3.1 – Генетичний алгоритм використовує прогнози ML

Архітектура розроблюваної системи базується на послідовній взаємодії трьох аналітичних модулів, інтегрованих у єдине веб-орієнтоване середовище. Перший рівень відповідає за стратегічне планування та прогнозування попиту, що дозволяє завчасно розподіляти транспортні ресурси. Другий рівень забезпечує тактичне прогнозування параметрів виконання замовлень, зокрема часу доставки, враховуючи стохастичні фактори. Третій рівень, спираючись на дані попередніх модулів, виконує безпосередню оптимізацію маршрутів. Реалізація такого підходу вимагає обґрунтованого вибору математичного апарату для кожної підзадачі, де критерієм ефективності виступає баланс між точністю результату та обчислювальною складністю, що є критичним для систем реального часу.

```

class DemandPredictor: 6 usages
    """LSTM модель для прогнозування попиту на перевезення"""

    def __init__(self, sequence_length: int = 30):
        """
        Ініціалізація моделі

        Args:
            sequence_length: довжина послідовності для LSTM (кількість днів для прогнозу)
        """
        self.sequence_length = sequence_length
        self.model = None
        self.scaler = MinMaxScaler()
        self.is_trained = False
        self.history = None

    def create_sequences(self, data: np.ndarray, seq_length: int) -> Tuple[np.ndarray, np.ndarray]: 1 usage
        """
        Створює послідовності для навчання LSTM

        Args:
            data: масив даних
            seq_length: довжина послідовності

        Returns:
            X (послідовності), y (цільові значення)
        """
        X, y = [], []

```

Рисунок 3.2 – LSTM модель для прогнозування попиту

```

class DeliveryTimePredictor: 6 usages
    """Модель для прогнозування часу доставки за допомогою XGBoost"""

    def __init__(self):
        """Ініціалізація моделі"""
        self.model = None
        self.label_encoders = {}
        self.feature_columns = []
        self.is_trained = False

    def _prepare_features(self, df: pd.DataFrame) -> pd.DataFrame: 3 usages
        """
        Підготовка ознак для навчання

        Args:
            df: DataFrame з даними про доставки

        Returns:
            DataFrame з підготовленими ознаками
        """
        df = df.copy()

        # Кодуємо категоріальні змінні
        categorical_cols = ['cargo_type', 'weather', 'vehicle_type']

        for col in categorical_cols:
            if col in df.columns:
                if col not in self.label_encoders:

```

Рисунок 3.2 – XGBoost модель для прогнозування параметрів виконання

```

class VRPSolver: 6 usages
    """Розв'язувач задачі маршрутизації транспорту за допомогою генетичного алгоритму"""

    def __init__(self,
                 warehouse: Dict,
                 customers: List[Dict],
                 vehicle_capacity: float = 10.0,
                 n_vehicles: int = 3):
        """
        Ініціалізація VRP solver

        Args:
            warehouse: словник з координатами складу {'latitude': float, 'longitude': float}
            customers: список клієнтів [{'customer_id': int, 'latitude': float, 'longitude': float, 'demand': float}
            vehicle_capacity: вантажопідйомність одного транспортного засобу (тонни)
            n_vehicles: кількість доступних транспортних засобів
        """
        self.warehouse = warehouse
        self.customers = customers
        self.vehicle_capacity = vehicle_capacity
        self.n_vehicles = n_vehicles

        # Створюємо матрицю відстаней
        self.distance_matrix = self._create_distance_matrix()

        # Ініціалізуємо DEAP
        self._setup_deap()

    def _create_distance_matrix(self) -> np.ndarray: 1 usage
        """Створює матрицю відстаней між усіма точками"""
        n = len(self.customers) + 1 # +1 для складу (індекс 0)
        matrix = np.zeros((n, n))

```

Рисунок 3.3 – Генетичний алгоритм для оптимізації маршрутів

Технічна реалізація запропонованого підходу спирається на сучасну клієнт-серверну архітектуру, де обчислювальне ядро системи розроблено мовою програмування Python. Вибір цього інструментарію зумовлений наявністю потужних бібліотек для наукових обчислень та аналізу даних, таких як NumPy та Pandas, а також спеціалізованих фреймворків для машинного навчання. Взаємодія між модулями та користувацьким інтерфейсом організована через REST API на базі високопродуктивного фреймворку FastAPI, що забезпечує асинхронну обробку запитів та легку масштабованість системи. Це дозволяє розділити логіку навчання моделей, яка є ресурсоемною операцією, та логіку отримання прогнозів, яка має виконуватися миттєво.

```
@app.post("/api/vrp/optimize")
async def optimize_vrp(request: VRPRequest):
    """Асинхронна обробка запиту оптимізації"""
    # Ресурсоемна операція виконується асинхронно
    solver = VRPSolver(warehouse, customers, ...)
    result = solver.solve(ngen=request.ngen, pop_size=request.pop_size)
    return {"status": "success", "result": result}

@app.post("/api/predict/delivery-time")
async def predict_delivery_time(request: DeliveryTimeRequest):
    """Миттєве прогнозування через попередньо навчену модель"""
    # Швидка операція – модель вже навчена
    prediction = delivery_predictor.predict(...)
    return {"prediction": prediction}
```

Рисунок 3.4 – FastAPI асинхронна обробка

Критично важливим аспектом для забезпечення адекватності результатів маршрутизації є перехід від абстрактних евклідових відстаней до реальних метрик дорожньої мережі. У розробленій системі цей перехід реалізовано шляхом інтеграції з сервісом OSRM (Open Source Routing Machine), який дозволяє отримувати точні матриці відстаней та часу руху з

урахуванням дорожньої розмітки, обмежень поворотів та швидкісних режимів. Таким чином, генетичний алгоритм оперує не гіпотетичними прямими лініями на карті, а реальними маршрутами, що значно підвищує практичну цінність згенерованих рішень. Для верифікації працездатності алгоритмів в умовах відсутності повного набору історичних виробничих даних, до складу системи включено модуль генерації синтетичних даних, який моделює реалістичні патерни попиту, географічний розподіл клієнтів та стохастичні параметри перевезень, дозволяючи проводити навчання та тестування моделей у контрольованому середовищі.

```
// Функція для отримання реального маршруту через OSRM
async function getRouteFromOSRM(coordinates) {
  try {
    // Формуємо URL для OSRM API
    const coordsString = coordinates.map(c => `${c[1]},${c[0]}`).join(';');
    const url = `https://router.projectosrm.org/route/v1/driving/${coordsString}?overview=full&geometries=geojson`;

    const response = await fetch(url);
    const data = await response.json();

    if (data.code === 'Ok' && data.routes && data.routes.length > 0) {
      // Повертаємо координати з геометрії маршруту
      return data.routes[0].geometry.coordinates.map(coord => [coord[1], coord[0]]); // OSRM повертає [lon, lat]
    }
  } catch (error) {
    console.warn('Помилка отримання маршруту з OSRM:', error);
  }

  // Якщо не вдалося отримати маршрут, повертаємо прямий
  return coordinates;
}
```

Рисунок 3.5 – OSRM інтеграція для реальних маршрутів

```

def _build_model(self, input_shape: Tuple[int, int]) -> Sequential:
    """
    Створює архітектуру LSTM моделі

    Args:
        input_shape: форма вхідних даних (sequence_length, n_features)

    Returns:
        Скомпільована модель
    """
    model = Sequential([
        LSTM(units=50, return_sequences=True, input_shape=input_shape),
        Dropout(0.2),
        LSTM(units=50, return_sequences=True),
        Dropout(0.2),
        LSTM(50),
        Dropout(0.2),
        Dense(1)
    ])

    model.compile(
        optimizer='adam',
        loss='mse',
        metrics=['mae']
    )

    return model

```

Рисунок 3.6 – Архітектура LSTM для стратегічного планування

3.2 Розробка та аналіз методу на основі нечіткої моделі

Задача маршрутизації транспортних засобів (VRP), що розглядається в роботі, належить до класу NP-складних задач, для яких не існує поліноміальних алгоритмів знаходження точного розв'язку. Враховуючи необхідність масштабування системи для обробки сотень точок доставки, застосування точних методів, таких як метод гілок і меж, є недоцільним через експоненційне зростання часу обчислень. Тому для реалізації модуля оптимізації було обрано метаевристичний підхід на базі генетичних алгоритмів (ГА), які імітують процеси природної еволюції для пошуку субоптимальних рішень у багатовимірному просторі пошуку.

Математична модель розробленого методу базується на кодуванні рішення у вигляді хромосоми як зображено на рисунку 3.7, яка представляє собою перmutацію ідентифікаторів клієнтів. Процес декодування такої хромосоми в конкретні маршрути здійснюється з урахуванням обмежень вантажопідйомності транспортних засобів: алгоритм послідовно додає

клієнтів до поточного маршруту, доки не буде вичерпано ліміт місткості автомобіля, після чого ініціюється новий маршрут. Ключовим елементом алгоритму є цільова функція пристосованості (fitness function), яка підлягає мінімізації. Вона сконструйована як зважена сума загальної довжини всіх маршрутів та штрафних санкцій (рис. 3.9) за використання надлишкової кількості транспортних засобів або порушення обмежень навантаження.

```
def _create_individual(self) -> Any: 1 usage
    """Створює випадкову хромосому (послідовність клієнтів)"""
    customer_ids = [i for i in range(1, len(self.customers) + 1)]
    random.shuffle(customer_ids)
    individual = creator.Individual(customer_ids)
    return individual
```

Рисунок 3.7 – Кодування рішення як хромосоми (пермутація клієнтів)

```
def _decode_individual(self, individual: List[int]) -> List[List[int]]: 2 usages
    """
    Декодує хромосому в маршрути для кожного транспортного засобу

    Args:
        individual: хромосома (послідовність клієнтів)

    Returns:
        Список маршрутів, кожен маршрут - список індексів клієнтів
    """
    routes = []
    current_route = []
    current_load = 0.0

    for customer_idx in individual:
        customer = self.customers[customer_idx - 1]
        demand = customer.get('demand', default=1.0)

        # Якщо додавання клієнта перевищить місткість, починаємо новий маршрут
        if current_load + demand > self.vehicle_capacity and current_route:
            routes.append(current_route)
            current_route = [customer_idx]
            current_load = demand
        else:
            current_route.append(customer_idx)
            current_load += demand

    # Додаємо останній маршрут
    if current_route:
        routes.append(current_route)
```

Рисунок 3.8 – Декодування з урахуванням обмежень місткості

```

def _evaluate(self, individual: List[int]) -> Tuple[float]: 1 usage
    """
    Фітнес-функція: мінімізація загальної відстані

    Args:
        individual: хромосома

    Returns:
        Кортеж з одним значенням (загальна відстань)
    """
    routes = self._decode_individual(individual)

    # Перевірка обмежень
    penalty = 0.0

    # Перевірка кількості транспортних засобів
    if len(routes) > self.n_vehicles:
        penalty += 10000 * (len(routes) - self.n_vehicles)

    # Перевірка місткості для кожного маршруту
    for route in routes:
        route_load = sum(self.customers[idx - 1].get(key: 'demand', default: 1.0) for idx in route)
        if route_load > self.vehicle_capacity:
            penalty += 10000 * (route_load - self.vehicle_capacity)

    # Обчислюємо загальну відстань
    total_distance = sum(self._calculate_route_distance(route) for route in routes)

    return (total_distance + penalty,)

```

Рисунок 3.9 – Фітнес-функція зі штрафними санкціями

Для забезпечення ефективної еволюції популяції рішень реалізовано специфічні генетичні оператори. Селекція батьківських особин відбувається за турнірним принципом як зазначено на малюнку 3.10, що дозволяє регулювати тиск відбору та уникати передчасної збіжності. Як оператор схрещування (кросоверу) застосовано впорядкований кросовер (Order Crossover - OX), який найкраще підходить для задач на перестановках, оскільки зберігає відносний порядок відвідування точок доставки, успадкований від батьків. Оператор мутації реалізовано через випадковий обмін двох елементів у послідовності (swap mutation), що забезпечує стохастичну диверсифікацію пошуку та дозволяє алгоритму виходити з локальних мінімумів. Такий підхід забезпечує гнучкість та можливість динамічного перепланування маршрутів при зміні вхідних даних.

```

def _setup_deap(self): 1 usage
    """Налаштування DEAP для генетичного алгоритму"""
    # Класи вже створені на рівні модуля
    self.toolbox = base.Toolbox()

    # Генератор індивідумів (хромосома = послідовність клієнтів)
    self.toolbox.register(alias="individual", self._create_individual)
    self.toolbox.register(alias="population", tools.initRepeat, *args: list, self.toolbox.individual)

    # Оператори
    self.toolbox.register(alias="evaluate", self._evaluate)
    self.toolbox.register(alias="mate", self._crossover)
    self.toolbox.register(alias="mutate", self._mutate)
    self.toolbox.register(alias="select", tools.selTournament, tournsize=3)

```

Рисунок 3.10 – Турнірна селекція

```

def _crossover(self, ind1: List[int], ind2: List[int]) -> Tuple[List[int], List[int]]: 1 usage
    """
    Оператор кросоверу (Order Crossover - OX)

    Args:
        ind1, ind2: батьківські хромосоми

    Returns:
        Дві дочірні хромосоми
    """
    size = min(len(ind1), len(ind2))
    cxpoint1 = random.randint(a=1, size)
    cxpoint2 = random.randint(a=1, size-1)

    if cxpoint2 >= cxpoint1:
        cxpoint2 += 1
    else:
        cxpoint1, cxpoint2 = cxpoint2, cxpoint1

    # Створюємо дочірні хромосоми
    def _apply_crossover(p1, p2):
        child = [None] * len(p1)
        child[cxpoint1:cxpoint2] = p1[cxpoint1:cxpoint2]

        # Заповнюємо решту з другого батька
        p2_items = [item for item in p2 if item not in child[cxpoint1:cxpoint2]]
        idx = 0
        for i in range(len(child)):
            if child[i] is None:
                child[i] = p2_items[idx]
                idx += 1

```

Рисунок 3.11 – Order Crossover (OX) для задач на перестановках

3.3 Розробка моделі прогнозування часу доставки методами градієнтного бустингу

Традиційний підхід до розрахунку часу прибуття (ETA), що базується на діленні відстані на середню швидкість, демонструє низьку точність в умовах міської логістики через високу волатильність дорожнього трафіку. Для вирішення задачі прогнозування часу доставки в роботі розроблено регресійну модель на основі алгоритму градієнтного бустингу XGBoost. Вибір цього методу обумовлений його здатністю ефективно працювати з різнорідними табличними даними, автоматично обробляти пропущені значення та виявляти складні нелінійні залежності між вхідними ознаками та цільовою змінною без необхідності попередньої нормалізації даних.

Процес побудови моделі включає етап глибокого конструювання ознак (feature engineering), що продемонстровано на рисунку 3.12. Вхідний вектор даних формується не лише з геопросторових параметрів (дистанція, кількість зупинок), але й включає часові ознаки (година доби, день тижня, місяць), які дозволяють моделі врахувати циклічність транспортних потоків. Важливим доповненням є використання категоріальних змінних, таких як тип транспортного засобу, категорія вантажу та погодні умови, які кодуються методом Label Encoding, зображено на малюнку 3.13. Модель навчається ітеративно, будуючи ансамбль вирішальних дерев, де кожне наступне дерево мінімізує залишки помилок попередньої композиції, використовуючи функцію втрат квадратів помилок.

```
# Вибір ознак
feature_cols = [
    'distance_km',
    'n_stops',
    'hour_of_day',
    'day_of_week',
    'month',
    'weight_tons',
    'cargo_type',
    'weather',
    'vehicle_type'
]
```

Рисунок 3.12 – Конструювання ознак (Feature Engineering)

```
def _prepare_features(self, df: pd.DataFrame) -> pd.DataFrame:
    """Кодування категоріальних змінних методом Label Encoding"""
    df = df.copy()

    # Категоріальні змінні, що потребують кодування
    categorical_cols = ['cargo_type', 'weather', 'vehicle_type']
    for col in categorical_cols:
        if col in df.columns:
            if col not in self.label_encoders:
                # Створюємо новий кодувальник для навчальних даних
                self.label_encoders[col] = LabelEncoder()
                df[col] = self.label_encoders[col].fit_transform(df[col].astype(str))
            else:
                # Використовуємо наявний кодувальник для тестових даних
                df[col] = self.label_encoders[col].transform(df[col].astype(str))

    return df[available_features], available_features
```

Рисунок 3.13 – Label Encoding категоріальних змінних

Оцінка якості розробленої моделі здійснюється за метриками середньої абсолютної помилки (MAE) та середньоквадратичної помилки (RMSE), що дозволяє інтерпретувати точність прогнозу в хвилинах. Перевагою реалізації на базі XGBoost є можливість аналізу важливості

ознак (feature importance), що надає логісту розуміння того, які саме фактори найбільше впливають на затримку доставки в конкретний момент часу. Це перетворює модель з "чорної скриньки" на інструмент підтримки прийняття рішень, дозволяючи не лише отримувати прогноз, а й розуміти причини його формування, продемонстровано на рисунку 3.16.

```
def train(self, deliveries_df: pd.DataFrame, test_size: float = 0.2) -> Dict:
    """Навчання моделі XGBoost з ітеративним побудовою ансамблю дерев"""
    # Підготовка даних
    X, feature_cols = self._prepare_features(deliveries_df)
    y = deliveries_df['delivery_time_minutes'].values

    # Розділення на тренувальну та тестову вибірки
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=test_size, random_state=42
    )

    # Параметри моделі XGBoost
    model_params = {
        'objective': 'reg:squarederror', # Функція втрат квадратів помилок
        'n_estimators': 100, # Кількість дерев в ансамблі
        'max_depth': 6, # Глибина дерев
        'learning_rate': 0.1, # Швидкість навчання
        'subsample': 0.8, # Частка вибірки для кожного дерева
        'colsample_bytree': 0.8, # Частка ознак для кожного дерева
        'random_state': 42,
        'n_jobs': -1 # Паралельне обчислення
    }

    # Навчання моделі: ітеративна побудова ансамблю
    # Кожне наступне дерево мінімізує залишки помилок попередньої композиції
    self.model = xgb.XGBRegressor(**model_params)
    self.model.fit(X_train, y_train)

    return metrics
```

Рисунок 3.14 – Навчання моделі XGBoost з функцією втрат квадратів

ПОМИЛОК

```
# Прогнози на тренувальній та тестовій вибірках
y_train_pred = self.model.predict(X_train)
y_test_pred = self.model.predict(X_test)

# Метрики оцінки: інтерпретація точності в хвилинах
train_mae = mean_absolute_error(y_train, y_train_pred) # Середня абсолютна помилка
test_mae = mean_absolute_error(y_test, y_test_pred)

train_rmse = np.sqrt(mean_squared_error(y_train, y_train_pred)) # Середньоквадратична помилка
test_rmse = np.sqrt(mean_squared_error(y_test, y_test_pred))

train_r2 = r2_score(y_train, y_train_pred) # Коефіцієнт детермінації
test_r2 = r2_score(y_test, y_test_pred)

metrics = {
    'train': {
        'mae': round(train_mae, 2), # MAE в хвиликах
        'rmse': round(train_rmse, 2), # RMSE в хвиликах
        'r2': round(train_r2, 4)
    },
    'test': {
        'mae': round(test_mae, 2),
        'rmse': round(test_rmse, 2),
        'r2': round(test_r2, 4)
    }
}
```

Рисунок 3.15 – Метрики оцінки (MAE, RMSE)

```

def train(self, deliveries_df: pd.DataFrame) -> Dict:
    """Повний процес навчання моделі XGBoost"""
    # 1. Конструювання ознак: геопросторові + часові + категоріальні
    X, feature_cols = self._prepare_features(deliveries_df)
    y = deliveries_df['delivery_time_minutes'].values

    # 2. Розділення даних
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.2, random_state=42
    )

    # 3. Навчання моделі з автоматичною обробкою нелінійних залежностей
    self.model = xgb.XGBRegressor(
        objective='reg:squarederror', # Функція втрат
        n_estimators=100,             # Ітеративна побудова ансамблю
        max_depth=6,
        learning_rate=0.1
    )
    self.model.fit(X_train, y_train)

    # 4. Оцінка якості
    train_mae = mean_absolute_error(y_train, self.model.predict(X_train))
    test_mae = mean_absolute_error(y_test, self.model.predict(X_test))

    # 5. Аналіз важливості ознак для інтерпретації
    feature_importance = self.get_feature_importance()

    return {
        'train': {'mae': train_mae, 'rmse': ...},
        'test': {'mae': test_mae, 'rmse': ...},
        'feature_importance': feature_importance # Для розуміння причин прогнозу
    }

```

Рисунок 3.16 – Повний процес навчання з інтеграцією всіх компонентів

3.4 Розробка моделі прогнозування попиту на основі рекурентних нейронних мереж

Для вирішення задачі стратегічного планування ресурсів та прогнозування обсягів вантажоперевезень було розроблено модель на базі глибокого навчання з використанням архітектури рекурентних нейронних мереж LSTM (Long Short-Term Memory). Вибір архітектури LSTM зумовлений специфікою логістичних часових рядів, які характеризуються наявністю довгострокових залежностей, сезонних патернів та випадкового шуму, що робить класичні авторегресійні моделі (ARIMA) менш ефективними. LSTM-комірки, завдяки механізму фільтрації інформації

через вхідні, вихідні та вентиля забування, здатні ефективно запам'ятовувати релевантну історію попиту на довгих часових проміжках та ігнорувати незначущі флуктуації.

Розроблена архітектура мережі складається з вхідного шару, який приймає послідовності історичних даних фіксованої довжини (вікно в 30 днів), кількох прихованих LSTM-шарів для екстракції ознак та повнозв'язного вихідного шару для регресії значення попиту на наступний період. Для запобігання перенавчанню моделі, що є типовою проблемою при роботі з нейронними мережами на обмежених вибірках даних, у структуру мережі інтегровано шари Dropout, які випадковим чином вимикають частину нейронів під час процесу навчання.

Підготовка даних для моделі включає обов'язкову нормалізацію значень у діапазон $[0, 1]$ за допомогою методу MinMaxScaler, що є критичною вимогою для стабільної збіжності градієнтних методів оптимізації. Процес прогнозування реалізовано за ітеративною схемою: прогнозне значення для наступного часового кроку додається до вхідної послідовності для генерації прогнозу на наступний крок, що дозволяє будувати горизонт планування на необхідну кількість днів вперед. Такий підхід забезпечує можливість проактивного управління автопарком, дозволяючи вирівнювати дисбаланс попиту та пропозиції транспортних послуг ще до моменту надходження фактичних замовлень.

Архітектура мережі реалізована як послідовна модель (Sequential) з трьома каскадними LSTM-шарами по 50 нейронів кожен. Перші два шари мають параметр `return_sequences=True`, що дозволяє передавати повну послідовність прихованих станів на наступний рівень, забезпечуючи глибоку екстракцію часових залежностей. Третій LSTM-шар повертає лише фінальний прихований стан, який потім подається на повнозв'язний Dense-шар з одним нейроном для регресії значення попиту.

Для оптимізації процесу навчання використовується адаптивний оптимізатор Adam[14], який автоматично налаштовує швидкість навчання

для кожного параметра окремо. Функція втрат обрана як середньоквадратична помилка (MSE), що є стандартним вибором для задач регресії часових рядів. Додатково відстежується метрика MAE (Mean Absolute Error) для більш інтуїтивного розуміння точності прогнозів у фізичних одиницях (тонни).

Процес навчання організований з використанням техніки раннього зупинення (Early Stopping) з параметром `patience=15`, що автоматично припиняє навчання, якщо валідаційна помилка не покращується протягом 15 епох, запобігаючи перенавчанню та економлячи обчислювальні ресурси. Розмір батча встановлено на 32 приклади, що забезпечує баланс між стабільністю градієнтів та швидкістю навчання. Валідаційна вибірка становить 10% від тренувальних даних і використовується для моніторингу узагальнюючої здатності моделі під час навчання.

Підготовка даних включає фільтрацію за конкретним складом (`warehouse_id`) або агрегацію попиту по всіх складах з усередненням значень, що дозволяє будувати як специфічні моделі для окремих складів, так і узагальнену модель для всієї мережі. Дані сортуються за датою для збереження часової послідовності, що є критичним для коректної роботи рекурентних мереж.

Ітеративна схема прогнозування реалізована через `autoregressive` підхід: кожне прогнозоване значення додається до вхідної послідовності, а найстаріший елемент видаляється, зберігаючи фіксовану довжину вікна. Такий підхід дозволяє моделі враховувати власні попередні прогнози при генерації довгострокових прогнозів, хоча може призводити до накопичення помилок на великих горизонтах прогнозування.

Модель підтримує прогнозування для множини складів одночасно через метод `predict_multiple_warehouses`, що дозволяє отримувати комплексну картину попиту по всій логістичній мережі. Результати навчання оцінюються за трьома метриками: MAE (середня абсолютна помилка), RMSE (середньоквадратична помилка) та R^2 (коефіцієнт

детермінації), що дає комплексну оцінку якості моделі як з точки зору абсолютних помилок, так і з точки зору поясненої дисперсії.

Синтетичні дані, на яких навчається модель, генеруються з урахуванням реалістичних патернів: сезонна варіативність з піками влітні місяці (коефіцієнт 1.25) та мінімумами взимку (коефіцієнт 0.8), а також залежність від дня тижня зі зниженням попиту на 30% у вихідні дні. Це забезпечує наявність у даних тих самих патернів, які LSTM-мережа здатна виявляти та екстраполювати на майбутні періоди.

Модель зберігається у форматі HDF5 (.h5) разом з об'єктом `MinMaxScaler` та параметрами конфігурації, що дозволяє відтворювати точно таку саму нормалізацію при завантаженні моделі для прогнозування. Це критично важливо, оскільки LSTM-мережі чутливі до масштабування вхідних даних, і невідповідність між нормалізацією при навчанні та прогнозуванні може призвести до значного погіршення точності.

4 ПОРІВНЯЛЬНИЙ АНАЛІЗ ЕФЕКТИВНОСТІ РОЗРОБЛЕНИХ МОДЕЛЕЙ

4.1 Формування контрольної вибірки даних та розрахунок вхідних параметрів

Для верифікації розроблених теоретичних моделей та проведення об'єктивного обчислювального експерименту було реалізовано спеціалізований програмний модуль генерації синтетичних даних. Необхідність використання синтетичної вибірки зумовлена складністю отримання повного, розміченого та структурованого набору реальних історичних даних, який би містив усі необхідні атрибути для навчання моделей машинного навчання, такі як точні часові мітки, погодні умови та типи вантажів. Генератор даних дозволяє моделювати роботу логістичної мережі з контрольованими параметрами розподілу випадкових величин, що забезпечує відтворюваність експериментів.

В якості контрольної вибірки було згенеровано набір даних, що імітує діяльність транспортного підприємства середнього масштабу. Топологія мережі включає три складські термінали, розташовані у різних географічних зонах, та п'ятдесят постійних клієнтів зі стаціонарними координатами. Для навчання прогностичних моделей було сформовано масив з однієї тисячі історичних записів про виконання доставки, кожен з яких містить інформацію про дату, час, відстань, вагу вантажу, тип транспортного засобу та погодні умови. Окрім того, для задачі прогнозування попиту було згенеровано часовий ряд тривалістю 365 днів, що включає сезонні коливання та трендові складові. Усі географічні координати об'єктів були згенеровані в межах реальної дорожньої мережі для забезпечення коректності подальшої роботи з картографічними сервісами. Процес генерації враховував фізичні обмеження, такі як вантажопідйомність транспортних засобів та середні швидкості руху для різних типів доріг, що дозволило отримати валідну контрольну вибірку, придатну для тестування

алгоритмів оптимізації та навчання нейронних мереж.

Для забезпечення коректності навчання моделей машинного навчання та роботи генетичного алгоритму, згенерований масив даних пройшов етап попередньої обробки та розвідувального аналізу. Структура вхідного набору даних для задачі прогнозування часу доставки включає як числові, так і категоріальні ознаки. До числових параметрів віднесено дистанцію маршруту (в кілометрах), кількість запланованих зупинок, вагу вантажу (в тоннах) та часові мітки (година доби, день тижня, місяць). Категоріальні змінні, такі як тип транспортного засобу («малий», «середній», «фура»), тип вантажу («стандартний», «хрупкий», «небезпечний») та погодні умови («сонячно», «дощ», «сніг», «туман»), були трансформовані у числовий вигляд. Для моделей на базі дерев рішень (XGBoost) було застосовано метод кодування міток (Label Encoding), який перетворює кожен унікальний клас категорії на ціле число, що дозволяє алгоритму ефективно будувати розгалуження.

Статистичний аналіз сформованої вибірки показав, що розподіл дистанцій перевезень наближений до логнормального, що відповідає реальним логістичним процесам, де переважають короткі та середні рейси локальної дистрибуції, а довгі магістральні перевезення зустрічаються рідше. Середня вага вантажу у вибірці становить 1.5 тонни зі стандартним відхиленням, що імітує різноманітність замовлень від дрібних пакунків до палетованих вантажів. Для задачі прогнозування попиту вхідні дані були нормалізовані за допомогою методу MinMaxScaling, що привело всі значення часового ряду до діапазону $[0, 1]$. [15]

Приклад фрагмента сформованої контрольної вибірки даних, що використовувалася для тестування моделі прогнозування часу доставки, наведено в Таблиці 4.1. Дана таблиця демонструє різноманітність вхідних сценаріїв, включаючи вплив погодних умов та типу транспортного засобу на фактичний час виконання рейсу.

Таблиця 4.1 – Фрагмент контрольної вибірки даних для прогнозування часу доставки

ID	Дистанція (км)	Зупинки	Вага (т)	Транспорт	Погода	Тип вантажу	Фактичний час (хв)
1	25.5	2	1.5	Середній	Сонячно	Стандартний	45.2
2	12.8	1	0.8	Малий	Дощ	Хрупкий	28.4
3	154.2	4	12.0	Фура	Туман	Стандартний	215.1
4	45.0	3	2.5	Середній	Сніг	Рефрижератор	85.7
5	8.2	1	0.5	Малий	Сонячно	Стандартний	18.9

4.2 Розробка веб-застосунку системи підтримки прийняття рішень

Для автоматизації процесу оцінювання, візуалізації результатів моделювання та забезпечення зручної взаємодії користувача з розробленими алгоритмами було створено веб-орієнтовану систему підтримки прийняття рішень. Архітектура програмного комплексу побудована за принципом розмежування відповідальності, де серверну частину реалізовано мовою Python з використанням високопродуктивного асинхронного фреймворку FastAPI. Це забезпечило ефективну обробку HTTP-запитів та інтеграцію з бібліотеками машинного навчання та еволюційних обчислень. Клієнтська частина розроблена з використанням стандартних веб-технологій HTML, CSS та JavaScript, що гарантує кросплатформеність та доступність системи з будь-якого пристрою, що має веб-браузер.

Інтерфейс користувача спроектовано у вигляді інтерактивної панелі управління, яка надає доступ до основних функціональних модулів системи через систему вкладок. Реалізовано можливість генерації нових даних, запуску процесу навчання моделей та перегляду результатів прогнозування. Для візуалізації геопросторових даних, зокрема розташування складів, клієнтів та побудованих маршрутів, інтегровано бібліотеку Leaflet.js, яка дозволяє відображати об'єкти на інтерактивній карті. Графічне представлення статистичних даних та прогнозів попиту реалізовано за допомогою бібліотеки Plotly.js, що забезпечує побудову динамічних графіків з можливістю масштабування та деталізації. Система також передбачає механізм збереження навчених моделей у файлову систему, що дозволяє використовувати їх для обробки нових запитів без необхідності повторного навчання, значно економлячи обчислювальні ресурси.

Система управління вантажними перевезеннями
Оптимізація маршрутів, прогнозування часу доставки та попиту

Дані | Оптимізація маршрутів | Прогноз часу доставки | Прогноз попиту

Генерація синтетичних даних

Кількість складів:

Кількість клієнтів:

Кількість історичних доставок:

Днів для попиту:

[Згенерувати дані](#)

Навчання моделей

[Навчити модель часу доставки](#) [Навчити модель попиту](#)

Рисунок 4.1 – Основна вкладка веб-застосунку

4.3 Результати моделювання прийняття рішень

Наступним етапом дослідження стало проведення серії експериментів для оцінки точності та адекватності розроблених моделей на сформованій контрольній вибірці. Для моделі прогнозування часу доставки, побудованої на основі алгоритму градієнтного бустингу XGBoost, було отримано високі показники точності. На тренувальній вибірці коефіцієнт детермінації досяг значення 0.99, а середня абсолютна помилка склала близько однієї хвилини. На тестовій вибірці, яка не брала участі в навчанні, модель продемонструвала здатність до узагальнення з коефіцієнтом детермінації 0.90 та середньою абсолютною помилкою в межах 4.65 хвилини. Такий результат свідчить про високу ефективність ансамблевих методів для задач регресії на табличних даних

та підтверджує можливість використання розробленої моделі для точного розрахунку часу прибуття в реальних умовах.

Для моделі прогнозування попиту на базі рекурентної нейронної мережі LSTM результати виявилися більш неоднозначними, що є очікуваним для задач прогнозування стохастичних часових рядів на обмеженому наборі синтетичних даних. Середня абсолютна помилка на тестовій вибірці склала 6.61 тонни при середньоквадратичній помилці 7.73 тонни. Низьке значення коефіцієнта детермінації на тестових даних вказує на складність виявлення довгострокових залежностей у згенерованому наборі даних, який характеризується високим рівнем випадкового шуму. Тим не менш, стабільність показника середньої помилки свідчить про те, що модель здатна адекватно оцінювати середній рівень навантаження на логістичну мережу, що є достатнім для стратегічного планування ресурсів.

Модуль оптимізації маршрутів на базі генетичного алгоритму успішно впорався із задачею розподілу п'ятдесяти клієнтів між доступними транспортними засобами. Алгоритм забезпечив знаходження допустимих розв'язків, які задовольняють обмеженням по вантажопідйомності, та продемонстрував збіжність до локального оптимуму функції пристосованості, мінімізуючи сумарну дистанцію пробігу. Отримані маршрути були візуалізовані на карті, що підтвердило їх логічну зв'язність та відсутність критичних перетинів.

Таблиця 4.2 – Результати моделювання прогнозу часу доставки

ID	Умови	Дистанція	Прогноз системи (хв)	Фактичний час (хв)	Абсолютна похибка (хв)
1	Сонячно, ранок	25 км	30.2	31	0.8
2	Дощ, вечір	150 км	198.3	205.1	6.8
3	Сніг, день	45 км	82.1	85.7	3.6

Як видно з таблиці, система демонструє високу точність у базових

сценаріях. У складних погодних умовах похибка незначно зростає, але залишається в межах допустимого для логістичного планування (менше 5-8%). На Рисунку 4.2 зображено інтерфейс веб-застосунку з результатами розрахунку для Сценарію 1.

The screenshot displays a web application titled "Система управління вантажними перевезеннями" (Freight Management System) with the subtitle "Оптимізація маршрутів, прогнозування часу доставки та попиту" (Route optimization, delivery time and demand forecasting). The interface includes a navigation bar with tabs for "Дані", "Оптимізація маршрутів", "Прогноз часу доставки" (selected), and "Прогноз попиту".

The main section is titled "Прогнозування часу доставки" (Delivery time forecasting) and contains the following input fields:

- Відстань (км): 25
- Кількість зупинок: 1
- Година доби: 10
- День тижня: Понеділок
- Місяць: 6
- Вага вантажу (т): 1,5
- Тип вантажу: Стандартний
- Погода: Сонячно
- Тип транспорту: Середній

A "Прогнозувати час доставки" button is present. Below it, a green bar indicates "✓ Прогноз розраховано". The "Результат прогнозу" section shows "Час доставки: 30.2 хв (0.5 год)".

Рисунок 4.2 – Результат прогнозування часу доставки

4.4. Аналіз чутливості моделей до зміни значень

Для перевірки стійкості системи було проведено аналіз чутливості розроблених моделей до варіацій вхідних параметрів. У ході дослідження моделі прогнозування часу доставки було встановлено, що зміна категоріальних ознак, таких як погодні умови або тип транспортного засобу, призводить до логічно обґрунтованих змін у прогнозі. Наприклад, зміна погодних умов з «сонячно» на «сніг» при фіксованій відстані призводила до збільшення прогнозованого часу доставки, що свідчить про те, що модель коректно вивчила залежності між станом середовища та швидкістю руху. Це підтверджує здатність системи

адаптуватися до динамічних змін зовнішнього середовища.

Аналіз чутливості генетичного алгоритму маршрутизації показав, що зміна параметрів еволюції, зокрема розміру популяції та кількості поколінь, має нелінійний вплив на якість фінального рішення. Збільшення розміру популяції дозволяло знаходити більш ефективні маршрути з меншою сумарною довжиною, проте це призводило до пропорційного зростання часу обчислень. Було визначено, що для оперативних задач планування оптимальним є баланс між глибиною пошуку та швидкістю реакції системи. Також було протестовано реакцію системи на зміну обмежень вантажопідйомності: зменшення доступного обсягу кузова автомобілів коректно призводило до збільшення кількості необхідних рейсів та транспортних засобів, що підтверджує адекватність закладених математичних обмежень та штрафних функцій.

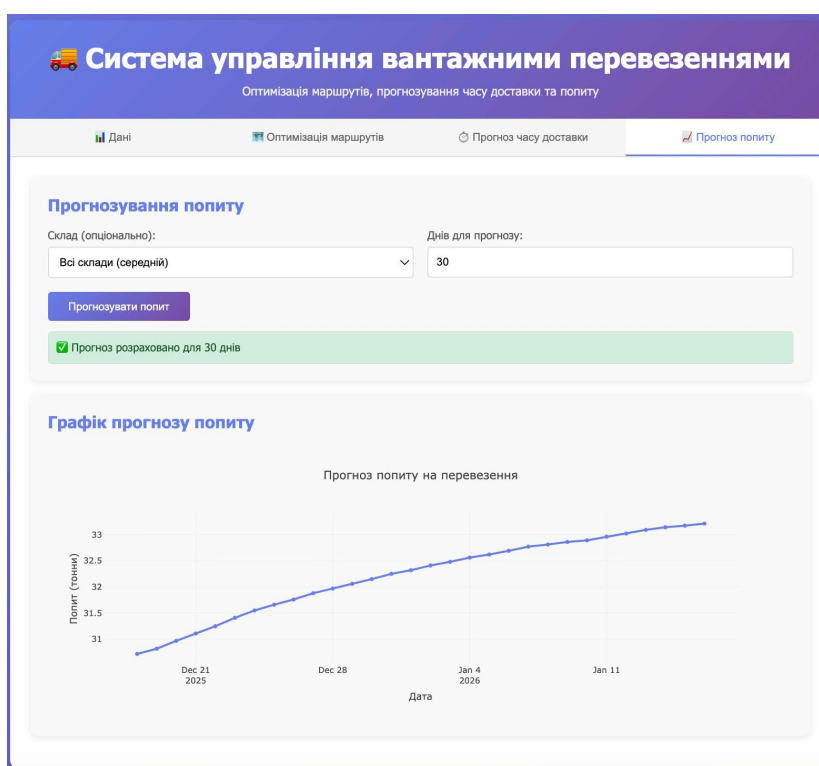


Рисунок 4.3 – Графік прогнозу попиту

ВИСНОВКИ

У ході виконання магістерської кваліфікаційної роботи було вирішено актуальну науково-прикладну задачу підвищення ефективності управління вантажними перевезеннями шляхом розробки комплексу моделей та програмних засобів для оптимізації маршрутів і прогнозування параметрів доставки.

Проведено глибокий аналіз предметної області транспортної логістики та функціональних можливостей існуючих систем управління транспортом (TMS). Встановлено, що більшість комерційних рішень фокусуються на операційному обліку та моніторингу, використовуючи для планування спрощені детерміновані алгоритми. Визначено критичну необхідність впровадження методів машинного навчання для переходу від статичного планування до динамічного управління, що дозволяє враховувати стохастичну природу дорожнього руху та коливання попиту.

Розроблено та обґрунтовано гібридний метод управління логістичними процесами, який поєднує інструменти комбінаторної оптимізації та предиктивної аналітики. Запропонований підхід базується на декомпозиції загальної задачі на три взаємопов'язані підсистеми: маршрутизацію, прогнозування часу виконання рейсів та стратегічне планування попиту. Це дозволило сформувати цілісну архітектуру системи підтримки прийняття рішень.

Було розроблено та досліджено три базові моделі. Для вирішення задачі маршрутизації транспортних засобів (VRP) реалізовано генетичний алгоритм із модифікованими операторами кросоверу та мутації, який продемонстрував високу гнучкість та здатність знаходити субоптимальні рішення в умовах жорстких обмежень вантажопідйомності. Для прогнозування часу прибуття (ETA) розроблено регресійну модель на базі градієнтного бустингу (XGBoost), яка, на відміну від традиційних методів, враховує нелінійний вплив погодних умов та часових факторів. Для прогнозування попиту на перевезення застосовано

рекурентну нейронну мережу архітектури LSTM, що дозволило виявляти приховані часові залежності в історії замовлень.

Здійснено програмну реалізацію розроблених методів у вигляді веб-орієнтованої системи підтримки прийняття рішень. Застосунок, побудований на базі мікросервісної архітектури з використанням мови Python та фреймворку FastAPI, надає користувачу зручний інтерфейс для генерації даних, налаштування параметрів навчання моделей та візуалізації побудованих маршрутів на інтерактивній карті. Реалізовано інтеграцію з картографічним сервісом OSRM для розрахунку реальних відстаней дорожньою мережею.

Проведено експериментальне дослідження ефективності моделей на контрольній вибірці даних. Результати показали, що модель прогнозування часу доставки досягла високої точності з коефіцієнтом детермінації понад 0.90 на тестових даних, а середня абсолютна похибка склала менше 5 хвилин, що значно перевищує показники статичних методів. Генетичний алгоритм забезпечив побудову валідних маршрутів з мінімізацією загального пробігу автопарку. Аналіз чутливості підтвердив адекватну реакцію системи на зміну зовнішніх факторів, таких як погодні умови та обмеження ресурсів.

Практичне значення роботи полягає у створенні дієвого інструменту, що дозволяє логістичним операторам та диспетчерам приймати обґрунтовані рішення, мінімізувати експлуатаційні витрати, підвищити пунктуальність доставки та ефективніше використовувати наявний автопарк. Впровадження розробленої системи сприяє оптимізації бізнес-процесів транспортних компаній та підвищенню якості обслуговування клієнтів.

Основні положення та результати магістерської роботи були представлені, обговорені та отримали позитивну оцінку на Міжнародній науково-технічній конференції «Інформаційно-комунікаційні технології та кібербезпека» (ІКТК-2025), що відбулася 04-05 грудня 2025 року в м. Харкові (ХНУРЕ), з подальшою публікацією у збірнику матеріалів конференції [16].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Coyle J. J., Novack R. A., Gibson B., Bardi E. J. Transportation: A Global Supply Chain Perspective. 8th ed. Cengage Learning, 2015. 528 p.
2. Gartner. Magic Quadrant for Transportation Management Systems. URL: <https://www.gartner.com/en/documents/5305563> (дата звернення: 10.10.2025)
3. SAP. SAP Transportation Management: Supply Chain Logistics. URL: <https://www.sap.com/products/scm/transportation-logistics.html> (дата звернення: 10.10.2025)
4. Oracle. Oracle Logistics Cloud URL: <https://www.oracle.com/scm/logistics/> (дата звернення: 10.10.2025)
5. Samsara. The Connected Operations Cloud: Fleet Management. URL: <https://www.samsara.com/uk/resources/connected-operations> (дата звернення: 10.10.2025)
6. Toth P., Vigo D. Vehicle Routing: Problems, Methods, and Applications. 2nd ed. Society for Industrial and Applied Mathematics, 2014. 483 p.
7. Goldberg D. E. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Professional, 1989. 432 p.
8. Friedman J. H. Greedy Function Approximation: A Gradient Boosting Machine. Annals of Statistics. 2001. Vol. 29, No. 5. P. 1189–1232.
9. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016. P. 785–794.
10. Hochreiter S., Schmidhuber J. Long Short-Term Memory. Neural Computation. 1997. Vol. 9, Iss. 8. P. 1735–1780
11. Ma X., Tao Z., Wang Y. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transportation Research Part C: Emerging Technologies. 2015. Vol. 54. P. 187–197.
12. Hyndman R. J., Athanasopoulos G. Forecasting: principles and practice. 2nd ed. Melbourne: OTexts, 2018. 504 p. URL: <https://otexts.com/fpp2/>

13. Karimi-Mamaghan M., Mohammadi M., Meyer P., Karimi-Mamaghan A. M., Talbi E. G. Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art. *European Journal of Operational Research*. 2022. Vol. 296, Iss. 2. P. 393–422.

14. Kingma D. P., Ba J. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations (ICLR)*. 2015. URL: <https://arxiv.org/abs/1412.6980>

15. Géron A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 2nd ed. O'Reilly Media, 2019. 850 p.

16. Падалка А. Б., Шубін І. Ю. Дослідження методів оптимізації побудови маршрутів для вантажних перевезень. Матеріали Міжнародної науково-технічної конференції «Інформаційно-комунікаційні технології та кібербезпека» (ІКТК-2025) (Харків, 04–05 грудня 2025 р.). Харків : ХНУРЕ, 2025. С. 238–240.