

Харківський національний автомобільно-дорожній університет

Міністерство освіти і науки України

Харківський національний автомобільно-дорожній університет

Міністерство освіти і науки України

*Кваліфікаційна наукова
робота на правах рукопису*

ГРИЦУК ВАЛЕРІЙ ЮРІЙОВИЧ

УДК 004.65:004.85:629.33.08

ДИСЕРТАЦІЯ

**МЕТОДИ ТА МОДЕЛІ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ
МОНІТОРИНГУ ТЕХНІЧНОГО СТАНУ ТРАНСПОРТНИХ ЗАСОБІВ**

Спеціальність 122 – Комп’ютерні науки

Галузь знань 12 – Інформаційні технології

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ В. Ю. Грицук

Науковий керівник: Пронін Сергій Вікторович.

кандидат технічних наук, доцент

Харків – 2026

АНОТАЦІЯ

Грицук В.Ю. Методи та моделі інформаційних технологій для моніторингу технічного стану транспортних засобів. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 122 «Комп'ютерні науки» (12 – Інформаційні технології). – Харківський національний автомобільно-дорожній університет Міністерства освіти і науки України, Харків, 2026.

Роботу виконано на кафедрі комп'ютерних наук та інформаційних систем Харківського національного автомобільно-дорожнього університету Міністерства освіти і науки України.

Об'єктом дослідження є інформаційні процеси обробки та аналізу діагностичних даних о технічному стані транспортних засобів.

Предметом дослідження є методи, моделі та інформаційні технології прогностичного моніторингу стану транспортних засобів.

Мета дослідження – підвищення ефективності технічного обслуговування транспортних засобів шляхом розробки методів, моделей та інформаційних технологій аналізу даних для прогностичного моніторингу технічного стану транспортних засобів, що забезпечує науково обґрунтований перехід від реактивного до проактивного обслуговування.

Наукова новизна одержаних результатів:

Вперше розроблено розподілену інформаційну технологію дистанційного контролю транспортних засобів, яка, на відміну від відомих реляційних систем, інтегрує нереляційні NoSQL-бази даних (MongoDB), паралельні обчислення MapReduce та потокову обробку через брокери повідомлень (Apache Kafka). Застосування такого підходу вперше обґрунтовується необхідністю відмови від жорстко структурованих реляційних SQL-схем, що працюють за традиційною

та повільною моделлю «запит-відповідь», на користь гнучкої гетерогенної архітектури (BASE), здатної обробляти швидкі нестационарні потоки телеметрії. Це забезпечує консистентність даних і горизонтальну масштабованість за умов нерегулярного надходження діагностичної інформації (Logged Vehicle Data, LVD).

Удосконалено метод подолання розрідженості навчальних вибірок баз даних LVD шляхом застосування статистичних оцінок щільності розподілу ймовірностей (за рахунок використання статистичного фільтра на основі непараметричного критерію Колмогорова-Смірнова) та алгоритмів мінімізації втрат даних при розширенні діагностичного простору ознак (із використанням алгоритму пучкового пошуку – beam search – для поетапного додавання релевантних ознак з урахуванням граничних умов розміру датасету), що підвищує якість формування навчальних датасетів для задач прогностичної аналітики.

Отримало подальший розвиток архітектури багатошарового перцептрона (MLP) та рекурентних мереж довгої короткочасної пам'яті (LSTM) для виявлення прихованих дефектів у нестационарних потоках телеметрії OBD-II; порівняльним аналізом доведено переваги ансамблевих методів (Random Forest) над логістичною регресією за метриками F1-score та Matthews Correlation Coefficient (MCC) при класифікації технічного стану транспортних засобів.

Зміст дисертації

У вступі обґрунтовано актуальність теми в контексті стрімкого розвитку технологій підключених транспортних засобів (Connected Vehicles) та інтелектуальних транспортних систем. Сформульовано мету, завдання, об'єкт, предмет і методи дослідження, визначено наукову новизну та практичну цінність результатів.

У першому розділі виконано системний аналіз методів та інформаційних систем діагностики, оцінки і прогнозування технічного стану транспортних

засобів. Розглянуто еволюцію систем управління базами даних – від реляційних моделей до екосистем Big Data. Проаналізовано парадигми технічного обслуговування: реактивне (Run-to-Failure), планово-профілактичне та прогностичне (Predictive Maintenance), обґрунтовано переваги останнього. Досліджено можливості технологій Big Data для прогнозування транспортних потоків: методи Floating Car Data (FCD), алгоритми машинного навчання для керування трафіком, а також застосування аналітики в концепції розумного міста (Smart City, V2X). Показано потенціал OBD-II-даних як джерела Big Data для задач машинного навчання та обґрунтовано доцільність OLAP і дослідницького аналізу даних (EDA) для підтримки управлінських рішень.

У другому розділі розглянуто принципи збору телеметричної інформації через інтерфейс OBD-II, детально проаналізовано стандартизовані режими діагностики та класифікацію кодів несправностей (DTC). Вирішено проблему розрідженості даних у базах LVD засобами статистичного оцінювання щільності розподілу та мінімізації втрат вибірки. Обґрунтовано методологію проектування сховища даних на базі ETL-процесів, нереляційних документо-орієнтованих СУБД (MongoDB) та концепції семантичного сховища з онтологічними зв'язками між різномірними джерелами діагностичної інформації.

У третьому розділі побудовано аналітичний конвеєр прогностичного моніторингу та розроблено математичні моделі системи. Для класифікації технічного стану транспортних засобів реалізовано й порівняно чотири підходи: багатошаровий перцептрон (MLP), рекурентні мережі LSTM для аналізу нестационарних часових рядів показань датчиків, алгоритм Random Forest та логістичну регресію. Оцінку ефективності розроблених моделей здійснено за розширеним набором метрик, що базується на аналізі матриці плутанини, оскільки набори даних автосервісу є глибоко незбалансованими. Зокрема, використано Ассурасу (загальну точність класифікації), F1-score (гармонійне

середнє між точністю та повнотою), ROC AUC (площу під кривою робочої характеристики приймача) та MCC (коефіцієнт кореляції Метьюса для комплексної оцінки якості класифікації). Головним критерієм оцінки ефективності визначено здатність алгоритму максимізувати показники F1-score та MCC при одночасній суворій мінімізації хибно-негативних прогнозів (False Negatives), що дозволяє уникнути небезпечного пропуску реальних критичних дефектів ТЗ. За результатами обчислювального експерименту встановлено, що моделі на базі MLP та Random Forest демонструють суттєво вищу здатність до мінімізації хибно-негативних прогнозів критичних дефектів порівняно з логістичною регресією (конкретні числові значення метрик необхідно вказати після фіналізації експериментальної частини).

У четвертому розділі проведено проектування інформаційної системи, розроблено об'єктно-орієнтовану UML-модель та деталізовано архітектуру програмного комплексу: модуль збору даних з OBD-сканера, препроцесор нормалізації та перевірки цілісності, підсистема реального часу, сховище даних та блок машинного навчання. Реалізовано відмовостійку мікросервісну архітектуру з балансувальниками навантаження та резервними вузлами. Описано механізми автоматизованого планування ETL-процесів, стиснення архівних даних та реляційну схему метабази (PostgreSQL) для збереження структурованих результатів діагностики.

Практичне значення результатів. Розроблена система дозволяє СТО та сервісним центрам перейти від реактивної моделі ремонту до прогностичної аналітики, що оптимізує логістику запасних частин, мінімізує час простою комерційного транспорту та знижує ризик відмов критичних вузлів. Програмний і математичний апарат може бути інтегрований у системи розумного міста (Smart City) для комплексної оптимізації транспортних потоків та екологічного моніторингу. Результати роботи узгоджуються з пріоритетами національної транспортної стратегії.

Ключові слова: технології Big Data, великі дані, автомобільний сервіс, прогностичне обслуговування, штучні нейронні мережі, розподілені системи, багатошаровий перцептрон (MLP), машинне навчання, інфокомунікаційні мережеві системи, інформаційно-керувальна система, рекурентні мережі LSTM, бортова діагностика OBD-II, системи управління, Floating Car Data (FCD), NoSQL, MongoDB, ETL, сховища даних, семантичне сховище даних, UML-моделювання, метод аналізу ієрархій, Smart City.

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Наукові праці, в яких опубліковані основні наукові результати

1. Golovan, A., Gritsuk, I., Honcharuk, I., Volska, O., Grytsuk, Y., Hritsuk, V. Features of Optimization of Maintenance Plan for Thermoelectric Generators of Marine Propulsion Systems. In: Prentkovskis, O., Yatskiv (Jackiva), I., Skačkauskas, P., Karpenko, M., Stosiak, M. (eds) TRANSBALTICA XV: Transportation Science and Technology. TRANSBALTICA 2024. Lecture Notes in Intelligent Transportation and Infrastructure. Springer, Cham. P. 26-35 https://doi.org/10.1007/978-3-031-85390-6_3

(Особистий внесок здобувача: визначення оптимальної періодичності технічного обслуговування)

2. Грицук В.Ю. Аналіз та обґрунтування вибору структури даних для завдань керування транспортними потоками. *Вісник ХНАДУ*, вип.106, 2024. С.7-15. DOI: 10.30977/BUL.2219-5548.2024.106.0.7.

3. Gritsuk I., Pohorletskyi D., Khudiakov I., Bulgakov M., Volodarets M., Smyrnov O., Korohodskyi V., Symonenko R., Holovashchenko O., Hritsuk V. Development of an approach to the construction of an adapted model for ensuring the processes of thermal readiness of a vehicle based on fuel consumption and exhaust gas emissions. *Eastern-European Journal of Enterprise Technologies*. 2024. Vol. 6, no. 4 (132). P. 26–45. URL: <https://doi.org/10.15587/1729-4061.2024.316922>

(Особистий внесок здобувача: участь в розробці інформаційної моделі дистанційного збору телеметричних даних через інтерфейс OBD-II)

4. Грицук В.Ю., Пронін С.В. Аналіз та обґрунтування вибору моделі для моніторингу параметрів транспортного засобу та прогнозування технічного обслуговування. *Вісник ПДТУ. Серія: Технічні науки*. Том 1 №49 (2024). 219 с. С.56-73.

(Особистий внесок здобувача: брав участь у обговоренні результатів, підготував основну частину тексту, провів моделювання, отримав і обробив результати моделювання)

5. Грицук В.Ю., Пронін С.В. Перспективи впровадження прогностичного обслуговування транспортних засобів на основі машинного навчання. *Матеріали Всеукраїнської науково-практичної конференції здобувачів вищої освіти та молодих учених «Сучасний стан досліджень в сфері ІТ» (19 квітня 2024 року)*. Харків: ХНАДУ, 2024. 215 с. С.77 – 81.

(Особистий внесок здобувача: брав участь у обговоренні результатів, підготував основну частину тексту, провів моделювання, отримав і обробив результати моделювання)

6. Грицук В.Ю., Пронін С.В. Інтелектуальна система діагностики транспортних засобів на основі використання нейронних мереж. *29-й Міжнародний молодіжний форум «Радіoeлектроніка та молодь у ХХІ столітті»*. Зб. матеріалів форуму. Т. 6. Харків: ХНУРЕ. 2025. 630 с. С.117-119.

(Особистий внесок здобувача: брав участь у обговоренні результатів, підготував основну частину тексту, провів моделювання, отримав і обробив результати моделювання)

7. Грицук В.Ю., Грицук Ю.В. Прогностичне обслуговування в автомобільному сервісі: використання ІОТ та BIG DATA. *Збірник тез та доповідей міжнародної конференції «Енергетичні установки та альтернативні джерела енергії»*. 11–12 березня 2024 року. Харків: ФОП Бровін О.В., 2024. 424 с.

C.264-266.

(Особистий внесок здобувача: доповідав та брав участь у обговоренні результатів, підготував основну частину тексту, визначив основні напрямки застосування в прогностичному обслуговуванні)

8. Грицук В.Ю. Концептуальний підхід до створення сховища даних для автомобільного сервісу. *Міжнародна науково-практична конференція «Інноваційні підходи у відновленні транспортної інфраструктури в особливих умовах воєнного стану: виклики та перспективи». Збірник тез доповідей.* К.: НТУ, 2024. 662 с. С.56-59.

9. Грицук В.Ю., Грицук Ю.В. Модульна архітектура обробки та зберігання даних транспортних засобів. *Збірник тез та доповідей міжнародної конференції "Енергетичні установки та альтернативні джерела енергії". 11–12 березня 2025 року.* Харків: ФОП Бровін О.В., 2025. 192 с. С. 125-129.

(Особистий внесок здобувача: брав участь у обговоренні результатів, підготував основну частину тексту, розробка та обґрунтування модульної архітектури інтегрованої системи збору, попередньої обробки, зберігання й аналізу телеметричних даних транспортних засобів із використанням технологій Big Data, що концептуально формалізовано у вигляді UML-діаграм активності та компонентів)

10. Грицук В.Ю., Грицук Ю.В. Формування бази даних Logged Vehicle Data для машинного навчання в автомобільному сервісі. *Збірник тез доповідей міжнародної конференції «Енергетичні установки та альтернативні джерела енергії». 11–12 березня 2026 року.* Харків: ФОП Бровін О.В., 2026. 208 с. С.140-143.

(Особистий внесок здобувача: брав участь у обговоренні результатів, підготував основну частину тексту, розробка та обґрунтування методології подолання розрідженості навчальних вибірок бази даних LVD шляхом застосування статистичного фільтра на основі непараметричного критерію

Колмогорова-Смірнова та обгорткового методу підбору параметрів для задач машинного навчання)

Наукові праці, які засвідчують апробацію матеріалів дисертації:

11. Hritsuk V. Yu. The usage of databases and big data technology in the automobile transport. *Студентство. Наука. Іноземна мова: збірник наукових праць студентів, аспірантів та молодих науковців*. Вип. 14. Харків: ХНАДУ, 2022. 236 с. С.75-78.

12. Грицук В.Ю. Технології Big Data в автомобільному сервісі: принципи та проблеми. *Комп'ютерні технології і мехатроніка. Збірник наукових праць за матеріалами V Міжнародної науково-методичної конференції*. Харків, ХНАДУ, 2023. 99 с. С.3-7.

13. Грицук В.Ю. Розвиток систем автономного водіння за рахунок використання машинного навчання та BIG DATA. *Збірник наукових праць за матеріалами II Міжнародної науково-практичної конференції «Вища технічна освіта XXI століття: виклики, проблеми, перспективи», 14–15 грудня 2023 р.* Краматорськ – Івано-Франківськ: ДонНАБА. 2023. 338 с. С.325-328.

14. Mateichyk V., Gritsuk I., Ghita B., Polishchuk O., Litvinov M., Hritsuk V. Building the predictive models and a forecast schedule in the monitoring system for the temperature of the engine coolant. *Науково-прикладні аспекти автомобільної і транспортно-дорожньої галузей: матеріали VIII міжнарод. науково-практична конференція. м. Луцьк, 30-31 трав. 2024 р.* Луцьк, 2024. 107 с. С.9-12.

(Особистий внесок здобувача: розробка концепції аналітичного модуля для передбачення виходу параметрів технічного стану (зокрема температури охолоджувальної рідини) за припустимі межі)

15. Грицук В.Ю. Розробка концептуальної схеми побудови сховища даних. *Матеріали міжнародної наукової конференції «Математичне*

модельовання та інформаційні технології сучасності» (10 жовтня 2024 року). Харків: ХНАДУ. 2024. 240 с. С.25-29.

16. Грицук В.Ю. Інтеграція методів бортової самодіагностики транспортних засобів: класифікація та модельовання. *Матеріали міжнародної наукової конференції «Процеси цифровізації екосистем» (31 жовтня 2024 року). Харків, ХНАДУ, 2024. 173 с. С. 18-25.*

Наукові праці, які додатково відображають наукові результати дисертації:

17. Грицук В.Ю., Грицук Ю.В. Обмін інформацією та міжоб'єктна взаємодія технології V2I. *Комп'ютерні інтелектуальні системи та мережі. Матеріали XI Всеукраїнської науково практичної WEB конференції аспірантів, студентів та молодих вчених (21-23 березня 2018 р.). Кривий Ріг: ДВНЗ «Криворізький національний університет», 2018. С.66 – 68.*

(Особистий внесок здобувача: розробка структурної схеми обміну інформацією в рамках технології V2I)

18. Грицук А.І., Грицук В.Ю., Ченцов А.В., Матейчик П.В., Краснокутська З.І. Використання інформаційних баз даних в задачах моніторингу автомобільного транспорту. *Сучасні енергетичні установки на транспорті і технології та обладнання для їх обслуговування. 9-а Міжнародна науково-практична конференція, 13-14 вересня 2018 р. Херсон: ХДМА. 376 с. С.37-39.*

(Особистий внесок здобувача: аналіз принципів побудови реляційних баз даних та обґрунтування доцільності їх застосування для збору, збереження й обробки даних у системах дистанційного моніторингу технічного стану транспортних засобів)

19. Gritsuk I., Polishchuk O., Hritsuk V. Formation of an information exchange scheme on the example of a ship power plant. *Прикладні науково-технічні дослідження: матеріали VI міжнар. наук.-прак. конф., 14-16 трав. 2024 р.*

Академія технічних наук України. Івано-Франківськ: Видавець Кушнір Г. М., 2024. 292 с. С. 177-179.

(Особистий внесок здобувача: формування схеми інформаційного обміну телеметричними даними, яка передбачає збір показників із шини CAN та їхню передачу на сервер для подальшого прогностичного моніторингу технічного стану енергетичної установки)

20. Mateichyk V., Gritsuk I., Ghita B., Polishchuk O., Litvinov M., Hritsuk V. Formation of the data flow diagram of the information system for monitoring of a ship power plant. *Сучасні інформаційні та інноваційні технології на транспорті (MINTT – 2024). Збірник матеріалів XVI Міжнародної науково-практичної конференції (29–31 травня 2024 р., м. Одеса).* Одеса: Херсонська державна морська академія, 2024. 426 с. С.198-201.

(Особистий внесок здобувача: участь в розробці алгоритмів аналітичного модуля для прогностичного моделювання зміни показників технічного стану суднової енергетичної установки.)

ABSTRACT

Hrytsuk, V.Yu. Information technology methods and models for monitoring the technical condition of vehicles. – Qualifying scientific thesis in manuscript form.

Dissertation for the degree of Doctor of Philosophy in the specialty 122 “Computer Science” (12 – Information Technology). – Kharkiv National Automobile and Highway University of the Ministry of Education and Science of Ukraine, Kharkiv, 2026.

This work was carried out at the Department of Computer Science and Information Systems of the Kharkiv National Automobile and Highway University of the Ministry of Education and Science of Ukraine.

The object of the study is the information processes involved in processing and analyzing diagnostic data regarding the technical condition of vehicles.

The subject of the study is the methods, models, and information technologies for the predictive monitoring of vehicle condition.

The aim of the study is to improve the efficiency of vehicle maintenance by developing data analysis methods, models, and information technologies for the predictive monitoring of vehicle technical condition, thereby facilitating a scientifically grounded transition from reactive to proactive maintenance.

Scientific novelty of the results obtained

For the first time, a distributed information technology for remote vehicle monitoring has been developed that, unlike known relational systems, integrates non-relational NoSQL databases (MongoDB), parallel MapReduce computations, and stream processing via message brokers (Apache Kafka). This approach is justified by the need to move away from rigidly structured relational SQL schemas, which operate under the traditional and slow “query-response” model, in favor of a flexible, heterogeneous architecture (BASE) capable of processing fast, non-stationary telemetry streams. This ensures data consistency and horizontal scalability under conditions of irregular inflow of diagnostic information (Logged Vehicle Data, LVD).

The method for addressing the sparsity of training samples in LVD databases has been improved by applying statistical estimates of probability density (using a statistical filter based on the nonparametric Kolmogorov-Smirnov test) and algorithms that minimize data loss when expanding the diagnostic feature space (using a beam search algorithm to incrementally add relevant features while accounting for constraints on the dataset size), thereby improving the quality of training datasets for predictive analytics tasks.

The application of deep learning architectures – multilayer perceptrons (MLP) and long short-term memory (LSTM) networks – for detecting hidden defects in non-stationary OBD-II telemetry streams has been further developed. A comparative analysis

demonstrated that ensemble methods (Random Forest) outperformed logistic regression, based on the F1-score and Matthews Correlation Coefficient (MCC), when classifying the technical condition of vehicles.

Contents of the Dissertation

The introduction justifies the topic's relevance amid the rapid development of connected vehicle technologies and intelligent transportation systems. The purpose, objectives, object, subject, and research methods are defined, and the scientific novelty and practical value of the results are identified.

The first chapter presents a systematic analysis of methods and information systems for diagnosing, assessing, and predicting the technical condition of vehicles. The evolution of database management systems is examined—from relational models to Big Data ecosystems. Maintenance paradigms are analyzed: reactive (Run-to-Failure), scheduled preventive, and predictive maintenance, with the advantages of the latter substantiated. The potential of Big Data technologies for forecasting traffic flows is investigated: Floating Car Data (FCD) methods, machine learning algorithms for traffic management, and the application of analytics in the Smart City and V2X concepts. The potential of OBD-II data as a source of Big Data for machine learning tasks is demonstrated, and the feasibility of OLAP and exploratory data analysis (EDA) for supporting managerial decisions is substantiated.

The second chapter examines the principles of collecting telemetry data via the OBD-II interface and provides a detailed analysis of standardized diagnostic modes and the classification of diagnostic trouble codes (DTCs). The problem of data sparsity in LVD databases is addressed using statistical methods to estimate distribution density and minimize sampling loss. A methodology is justified for designing a data warehouse based on ETL processes, a non-relational document-oriented DBMS (MongoDB), and the concept of a semantic warehouse with ontological links among heterogeneous sources of diagnostic information.

In the third chapter, an analytical pipeline for predictive monitoring is developed, and mathematical models of the system are formulated. To classify the technical condition of vehicles, four approaches are implemented and compared: a multilayer perceptron (MLP), LSTM recurrent networks for analyzing non-stationary time series of sensor readings, the Random Forest algorithm, and logistic regression. The effectiveness of the developed models was evaluated using an extended set of metrics derived from confusion matrix analysis, given the high class imbalance in car service datasets. Specifically, the following metrics were used: Accuracy (overall classification accuracy), F1-score (harmonic mean of precision and recall), ROC AUC (area under the receiver operating characteristic curve), and MCC (Matthews' correlation coefficient) for a comprehensive assessment of classification quality. The primary criterion for evaluating effectiveness is the algorithm's ability to maximize the F1-score and MCC while strictly minimizing false negatives, thereby preventing the dangerous omission of actual critical vehicle defects. The results of the computational experiment show that models based on MLP and Random Forest significantly outperform logistic regression in minimizing false negatives for critical defects (specific numerical values of the metrics must be provided after the experimental part is finalized).

In the fourth chapter, an object-oriented UML model was developed, and the software architecture was detailed: a module for collecting data from an OBD scanner, a preprocessor for normalization and integrity checking, a real-time subsystem, a data warehouse, and a machine learning unit. A fault-tolerant microservice architecture with load balancers and redundant nodes has been implemented. The paper describes mechanisms for automated ETL scheduling, archive data compression, and the relational schema of the metadata database (PostgreSQL) for storing structured diagnostic results.

Practical significance of the results. The developed system enables auto repair shops and service centers to shift from a reactive repair model to predictive analytics,

optimizing spare parts logistics, minimizing commercial vehicle downtime, and reducing the risk of critical component failures. The software and mathematical framework can be integrated into Smart City systems to comprehensively optimize traffic flows and environmental monitoring. The results of this work align with the priorities of the national transportation strategy.

Keywords: Big Data technologies, Big Data, automotive service, predictive maintenance, artificial neural networks, distributed systems, multilayer perceptron (MLP), machine learning, information and communication network systems, information and control system, LSTM recurrent networks, OBD-II on-board diagnostics, control systems, Floating Car Data (FCD), NoSQL, MongoDB, ETL, data warehouses, semantic data warehouse, UML modeling, analytic hierarchy process, Smart City.

LIST OF APPLICANT'S PUBLICATIONS

Scientific papers, in which the main scientific results are published:

1. Golovan, A., Gritsuk, I., Honcharuk, I., Volska, O., Grytsuk, Y., Hritsuk, V. Features of Optimization of Maintenance Plan for Thermoelectric Generators of Marine Propulsion Systems. In: Prentkovskis, O., Yatskiv (Jackiva), I., Skačkauskas, P., Karpenko, M., Stosiak, M. (eds) TRANSBALTICA XV: Transportation Science and Technology. TRANSBALTICA 2024. Lecture Notes in Intelligent Transportation and Infrastructure. Springer, Cham. P. 26-35 https://doi.org/10.1007/978-3-031-85390-6_3

(Osobystyi vnesok zdobuvacha: vyznachennia optymalnoi periodychnosti tekhnichnoho obsluhovuvannia)

2. Hrytsuk V.Yu. Analiz ta obgruntuvannia vyboru struktury danykh dlia zavdan keruvannia transportnymy potokamy. *Visnyk KhNADU*, vyp.106, 2024. S.7-15. DOI: 10.30977/BUL.2219-5548.2024.106.0.7

3. Gritsuk I., Pohorletskyi D., Khudiakov I., Bulgakov M., Volodarets M., Smyrnov O., Korohodskyi V., Symonenko R., Holovashchenko O., Hritsuk V.

Development of an approach to the construction of an adapted model for ensuring the processes of thermal readiness of a vehicle based on fuel consumption and exhaust gas emissions. *Eastern-European Journal of Enterprise Technologies*. 2024. Vol. 6, no. 4 (132). P. 26–45. URL: <https://doi.org/10.15587/1729-4061.2024.316922>

(Osobystyi vnesok zdobuvacha: uchast v rozrobtsi informatsiinoi modeli dystantsiinoho zboru telemetrychnykh danykh cherez interfeis OBD-II)

4. Hrytsuk V.Yu., Pronin S.V. Analiz ta obgruntuvannia vyboru modeli dlia monitorynhu parametriv transportnoho zasobu ta prohnozuvannia tekhnichnoho obsluhovuvannia. *Visnyk PDTU. Seriia: Tekhnichni nauky*. Tom 1 №49 (2024). 219 s. S.56-73.

(Osobystyi vnesok zdobuvacha: brav uchast u obhovorenni rezultativ, pidhotuvav osnovnu chastynu tekstu, proviv modeliuvannia, otrymav i obrobiv rezultaty modeliuvannia)

5. Hrytsuk V.Yu., Pronin S.V. Perspektyvy vprovadzhennia prohnostychnoho obsluhovuvannia transportnykh zasobiv na osnovi mashynnoho navchannia. *Materialy Vseukrainskoi naukovo-praktychnoi konferentsii zdobuvachiv vyshchoi osvity ta molodykh uchenykh «Suchasnyi stan doslidzhen v sferi IT» (19 kvitnia 2024 roku)*. Kharkiv: KhNADU, 2024. 215 s. S.77 – 81.

(Особистий внесок здобувача: брав участь у обговоренні результатів, підготував основну частину тексту, провів моделювання, отримав і обробив результати моделювання)

6. Hrytsuk V.Yu., Pronin S.V. Intelktualna systema diahnostryky transportnykh zasobiv na osnovi vykorystannia neironnykh merezh. *29-y Mizhnarodnyi molodizhnyi forum «Radioelektronika ta molod u KhKhI stolitti»*. Zb. materialiv forumu. T. 6. Kharkiv: KhNURE. 2025. 630 s. S.117-119.

(Osobystyi vnesok zdobuvacha: brav uchast u obhovorenni rezultativ, pidhotuvav osnovnu chastynu tekstu, proviv modeliuvannia, otrymav i obrobiv rezultaty modeliuvannia)

7. Hrytsuk V.Yu., Grytsuk Yu.V. Prohnostychnie obsluhovuvannia v avtomobilnomu servisi: vykorystannia IOT ta BIG DATA. *Zbirnyk tez ta dopovidei mizhnarodnoi konferentsii «Enerhetychni ustanovky ta alternatyvni dzherela enerhii». 11–12 bereznia 2024 roku.* Kharkiv: FOP Brovin O.V., 2024. 424 s. S.264-266.

(Osobystyi vnesok zdobuvacha: dopovidav ta brav uchast u obhovorenni rezultativ, pidhotuvav osnovnu chastynu tekstu, vyznachyv osnovni napriamky zastosuvannia v prohnostychnomu obsluhovuvanni)

8. Hrytsuk V.Yu. Kontseptualnyi pidkhid do stvorennia skhovyshcha danykh dlia avtomobilnoho servisu. *Mizhnarodna nauково-praktychna konferentsiia «Innovatsiini pidkhody u vidnovlenni transportnoi infrastruktury v osoblyvykh umovakh voiennoho stanu: vyklyky ta perspektyvy».* Zbirnyk tez dopovidei. K.: NTU, 2024. 662 s. S.56-59.

9. Hrytsuk V.Yu., Grytsuk Yu.V. Modulna arkhitektura obrobky ta zberihannia danykh transportnykh zasobiv. *Zbirnyk tez ta dopovidei mizhnarodnoi konferentsii "Enerhetychni ustanovky ta alternatyvni dzherela enerhii". 11–12 bereznia 2025 roku.* Kharkiv: FOP Brovin O.V., 2025. 192 s. S. 125-129.

(Osobystyi vnesok zdobuvacha: brav uchast u obhovorenni rezultativ, pidhotuvav osnovnu chastynu tekstu, rozrobka ta obgruntuvannia modulnoi arkhitektury intehrovanoi systemy zboru, poperednoi obrobky, zberihannia y analizu telemetrychnykh danykh transportnykh zasobiv iz vykorystanniam tekhnolohii Big Data, shcho kontseptualno formalizovano u vyhliadi UML-diahram aktyvnosti ta komponentiv)

10. Hrytsuk V.Yu., Grytsuk Yu.V. Formuvannia bazy danykh Logged Vehicle Data dlia mashynnoho navchannia v avtomobilnomu servisi. *Zbirnyk tez dopovidei mizhnarodnoi konferentsii «Enerhetychni ustanovky ta alternatyvni dzherela enerhii». 11–12 bereznia 2026 roku.* Kharkiv: FOP Brovin O.V., 2026. 208 s. S.140-143.

(Osobystyi vnesok zdobuvacha: brav uchast u obhovorenni rezultativ, pidhotuvav osnovnu chastynu tekstu, rozrobka ta obgruntuvannia metodolohii

podolannya rozridzhenosti navchalnykh vybirok bazy danykh LVD shliakhom zastosuvannya statystychnoho filtra na osnovi neparametrychnoho kryteriiu Kolmohorova-Smirnova ta obhortkovoho metodu pidboru parametriv dlia zadach mashynnoho navchannia)

Published works for approbation purpose:

11. Hritsuk V. Yu. The usage of databases and big data technology in the automobile transport. *Studentstvo. Nauka. Inozemna mova: zbirnyk naukovykh prats studentiv, aspirantiv ta molodykh naukovtsiv. Vyp. 14. Kharkiv: KhNADU, 2022. 236 s. S.75-78.*

12. Hrytsuk V.Yu. Tekhnolohii Big Data v avtomobilnomu servisi: pryntsypy ta problemy. *Kompiuterni tekhnolohii i mekhatronika. Zbirnyk naukovykh prats za materialamy V Mizhnarodnoi naukovo-metodychnoi konferentsii. Kharkiv, KhNADU, 2023. 99 s. S.3-7.*

13. Hrytsuk V.Yu. Rozvytok system avtonomnoho vodinnia za dopomohoiu mashynnoho navchannia ta BIG DATA. *Zbirnyk naukovykh prats za materialamy II Mizhnarodnoi naukovo-praktychnoi konferentsii «Vyscha tekhnichna osvita KhKhI stolittia: vyklyky, problemy, perspektyvy», 14–15 hrudnia 2023. Kramatorsk – Ivano-Frankivsk: DonNABA. 2023. 338 s. S.325-328.*

14. Mateichyk V., Gritsuk I., Ghita B., Polishchuk O., Litvinov M., Hritsuk V. Building the predictive models and a forecast schedule in the monitoring system for the temperature of the engine coolant. *Naukovo-prykladni aspekty avtomobilnoi i transportno-dorozhnoi haluzei: materialy VIII mizhnarod. naukovo-praktychna konferentsiia. m. Lutsk, 30-31 trav. 2024 r. Lutsk, 2024. 107 s. S.9-12.*

(Osobystyi vnesok zdobuvacha: rozrobka kontseptsii analitychnoho modulia dlia peredbachennia vykhodu parametriv tekhnichnoho stanu (zokrema temperatury okholodzhuvanoi ridyny) za prypustymi mezhi))

15. Hrytsuk V.Yu. Rozrobka kontseptualnoi skhemy pobudovy skhovyshcha danykh. *Materialy mizhnarodnoi naukovoï konferentsii «Matematychni modeliuvannia ta informatsiini tekhnolohii suchasnosti» (10 zhovtnia 2024 roku)*. Kharkiv: KhNADU. 2024. 240 s. S.25-29.

16. Hrytsuk V.Yu. Intehratsiia metodiv bortovoi samodiahnostyky transportnykh zasobiv: klasyfikatsiia ta modeliuvannia. *Materialy mizhnarodnoi naukovoï konferentsii «Protsey tsyfrovizatsii ekosystem» (31 zhovtnia 2024 roku)*. Kharkiv, KhNADU, 2024. 173 s. S. 18-25.

Scientific works that additionally represent the scientific results of the dissertation:

17. Hrytsuk V.Yu., Grytsuk Yu.V. Obmin informatsiieiu ta mizhobiektna vzaiemodiia tekhnolohii V2I. *Kompiuterni intelektualni systemy ta merezhi. Materialy XI Vseukrainskoi naukovo praktychnoi WEB konferentsii aspirantiv, studentiv ta molodykh vchenykh (21-23 bereznia 2018 r.)*. Kryvyi Rih: DVNZ «Kryvorizkyi natsionalnyi universytet», 2018. S.66 – 68.

(Osobystyi vnesok zdobuvacha: rozrobka strukturnoi skhemy obminu informatsiieiu v ramkakh tekhnolohii V2I)

18. Hrytsuk A.I., Hrytsuk V.Yu., Chentsov A.V., Mateichyk P.V., Krasnokutska Z.I. Vykorystannia informatsiinykh baz danykh v zadachakh monitorynhu avtomobilnoho transportu. *Suchasni enerhetychni ustanovky na transporti i tekhnolohii ta obladnannia dlia yikh obsluhovuvannia. 9-a Mizhnarodna naukovo-praktychna konferentsiia, 13-14 veresnia 2018 r.* Kherson: KhDMA. 376 s. S.37-39.

(Osobystyi vnesok zdobuvacha: analiz pryntsypiv pobudovy reliatsiinykh baz danykh ta obgruntuvannia dotsilnosti yikh zastosuvannia dlia zboru, zberezhennia y obrobky danykh u systemakh dystantsiinoho monitorynhu tekhnichnoho stanu transportnykh zasobiv)

19. Gritsuk I., Polishchuk O., Hritsuk V. Formation of an information exchange

scheme on the example of a ship power plant. *Prykladni naukovo-tekhnichni doslidzhennia: materialy VI mizhnar. nauk.-prak. konf., 14-16 trav. 2024 r.* Akademiia tekhnichnykh nauk Ukrainy. Ivano-Frankivsk: Vydavets Kushnir H. M., 2024. 292 s. S. 177-179.

(Osobystyi vnesok zdobuvacha: formuvannia skhemy informatsiinoho obminu telemetrychnymy danymy, yaka peredbachaie zbir pokaznykiv iz shyny CAN ta yikhniu peredachu na server dlia podalshoho prohnostychnoho monitorynhu tekhnichnoho stanu enerhetychnoi ustanovky)

20. Mateichyk V., Gritsuk I., Ghita B., Polishchuk O., Litvinov M., Hritsuk V. Formation of the data flow diagram of the information system for monitoring of a ship power plant. *Suchasni informatsiini ta innovatsiini tekhnolohii na transporti (MINTT – 2024). Zbirnyk materialiv XVI Mizhnarodnoi naukovo-praktychnoi konferentsii (29–31 travnia 2024 r., m. Odesa).* Odesa: Khersonska derzhavna morska akademiia, 2024. 426 s. S.198-201.

(Osobystyi vnesok zdobuvacha: uchast v rozrobtsi alhorytmiv analitychnoho modulia dlia prohnostychnoho modeliuvannia zminy pokaznykiv tekhnichnoho stanu sudnovoii enerhetychnoi ustanovky)

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ.....	23
ВСТУП	26
РОЗДІЛ 1 АНАЛІЗ МЕТОДІВ ТА ІНФОРМАЦІЙНИХ СИСТЕМ	
ДІАГНОСТИКИ, ОЦІНКИ ТА ПРОГНОЗУВАННЯ ТЕХНІЧНОГО СТАНУ	33
1.1. Огляд технологій Big Data	33
1.2. Прогностичне обслуговування в автомобільному сервісі	37
1.3. Застосування Big Data для прогнозування	43
1.4. Постановка задачі дослідження.....	50
1.5. Висновки до першого розділу	52
РОЗДІЛ 2 ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ СТВОРЕННЯ СХОВИЩА ДАНИХ	
ДІАГНОСТИКИ ТРАНСПОРТНИХ ЗАСОБІВ	54
2.1. Процес збору даних	54
2.2. Режими роботи діагностичного обладнання	61
2.3. Методи пошуку несправностей	65
2.4. Logged Vehicle Data (LVD) та зв'язки з функціональними блоками	75
2.5. Побудова сховища даних	91
2.6. Висновки до другого розділу	112
РОЗДІЛ 3 МОДЕЛІ МАШИННОГО НАВЧАННЯ ДЛЯ СИСТЕМИ	
ПРОГНОСТИЧНОГО МОНІТОРИНГУ	115
3.1. Підготовка датасету	115
3.2. Вибір архітектури нейронної мережі для системи прогностичного	
моніторингу	121
3.2.1 Багатошаровий перцептрон.	121
3.2.2 Рекурентна мережа.	124
3.2.3. Підбір оптимізатора для нейронної мережі.	130
3.2.4. Визначення оптимальної структури автокодера (Autoencoder).	138

	22
3.2.5. Визначення оптимальної структури рекурентної мережі LSTM.....	139
3.3. Порівняний аналіз різних моделей машинного навчання	140
3.4. Двоступеневий конвеєр предиктивної діагностики	159
3.5. Багатокласова та мультилейбл діагностика	163
3.6. Архітектура інформаційної системи.....	175
3.7. Висновки до третього розділу	179
РОЗДІЛ 4 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ МОНІТОРИНГУ І ПРОГНОЗУВАННЯ ТЕХНІЧНОГО СТАНУ АТЗ	182
4.1. Загальна концепція системи.....	182
4.2. Компоненти системи	189
4.3. Структура зв'язків блоків даних	195
4.4. Оцінка ефективності системи	210
4.4.1. Оцінка якості обробки даних та моделей класифікації.	210
4.4.2. Розрахунок експлуатаційно-економічних коефіцієнтів.....	211
4.4.3. Багатокритеріальна оцінка методом аналізу ієрархій (MAI).	214
4.5. Висновки до четвертого розділу	216
ЗАГАЛЬНІ ВИСНОВКИ	218
ЛІТЕРАТУРА	219
ДОДАТОК А. АКТИ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОЇ РОБОТИ	247

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

АСІК – Автоматизована система ітеративного контролю.

БД – База даних (або бази даних).

БП – Багатошаровий перцептрон (також використовується англomовний аналог MLP).

ГП – Горизонт прогнозування (також використовується англomовний аналог PH).

ГРМ – Газорозподільний механізм.

ДВЗ – Двигун внутрішнього згоряння.

ЕБК (або ЕБУ) – Електронний блок керування (електронний блок управління).

ЕКМ – Електронний контрольний модуль.

КШМ – Кривошипно-шатунний механізм.

СД – Сховище даних (також використовується англomовний аналог DW).

СТО – Станція технічного обслуговування.

СУБД – Система управління базами даних.

ТЗ – Транспортний засіб.

УДК – Універсальна десяткова класифікація.

ЦПГ – Циліндро-поршнева група.

ІНМ – Штучна нейронна мережа.

Accuracy (CA) – Точність класифікації (Classification Accuracy).

BASE – Базова доступність, гнучкий стан, узгодженість у кінцевому підсумку (Basically Available, Soft State, Eventual Consistency – концепція нереляційних систем).

CAN – Мережа контролерів (Controller Area Network – автомобільний протокол внутрішньобортового зв'язку).

CAN FD – CAN з гнучкою швидкістю передачі даних (CAN with Flexible Data-Rate).

CEC – Карусель постійних помилок (Constant Error Carousel – елемент архітектури LSTM-мереж).

DTC – Діагностичні коди несправностей (Diagnostic Trouble Codes).

EDA – Дослідницький аналіз даних (Exploratory Data Analysis).

ETL – Процеси вилучення, перетворення та завантаження даних (Extract, Transform, Load).

F1-score – F-міра (гармонійне середнє між точністю та повнотою класифікації).

FCD – Дані рухомих автомобілів (Floating Car Data).

FN – Хибно-негативні результати (False Negatives – пропуски критичних дефектів).

FP – Хибно-позитивні результати (False Positives – помилкові тривоги).

HDFS – Розподілена файлова система Hadoop (Hadoop Distributed File System).

LIN – Локальна мережа зв'язку (Local Interconnect Network – низькошвидкісний автомобільний протокол).

LSTM – Рекурентні нейронні мережі довгої короткочасної пам'яті (Long Short-Term Memory).

LVD – Зареєстровані/записані діагностичні дані транспортного засобу (Logged Vehicle Data).

MCC – Коефіцієнт кореляції Меттьюса (Matthews Correlation Coefficient – метрика оцінки якості класифікації).

NoSQL – нереляційні бази даних (зокрема, документ-орієнтована СУБД MongoDB).

OLAP – Online Analytical Processing (технологія оперативної інтерактивної аналітичної обробки даних).

Predictive Maintenance (PdM) – Проактивне прогностичне технічне обслуговування ТЗ.

Precision (Prec) – Точність (прецизійність) класифікації.

Recall – Повнота класифікації.

ROC AUC – Площа під робочою характеристикою приймача (Receiver Operating Characteristic – Area Under Curve).

Run-to-Failure (RtF) – Реактивне технічне обслуговування (робота до відмови).

SDW – Семантичне сховище даних (Semantic Data Warehouse).

Smart City – Концепція «розумного міста».

TN – Істинно негативні результати (True Negatives).

TP – Істинно позитивні результати (True Positives).

UML – Уніфікована мова моделювання (Unified Modeling Language).

V2X – Технології зв'язку транспортного засобу з іншими об'єктами (Vehicle-to-Everything).

VSR – База даних технічного обслуговування транспортних засобів (Vehicle Maintenance database).

ВСТУП

Обґрунтування вибору теми дослідження. Сучасний етап розвитку автомобільного транспорту характеризується глибинною трансформацією, яку супроводжує перехід до концепції інтелектуальних транспортних засобів (Connected Vehicles), що функціонують як повноцінні вузли в екосистемі Інтернету речей (IoT) та які генерують значні обсяги телеметричної інформації у режимі реального часу. Умови інтенсивної експлуатації вимагають безперервного моніторингу фактичного технічного стану вузлів і агрегатів. Традиційні парадигми обслуговування – реактивна (Run-to-Failure) та планово-профілактична (Preventive Maintenance) – значною мірою вичерпали свій потенціал через високі економічні втрати від непередбачуваних простоїв і неповне використання ресурсів деталей. Законною еволюційною альтернативою є проактивне прогнозне обслуговування (Predictive Maintenance), яке базується на аналізі поточкових даних від бортових систем для точного прогнозування залишкового ресурсу механізмів.

Транспортні засоби оснащені розгалуженими мережами електронних блоків керування (ЕБК), об'єднаних гетерогенними протоколами (CAN, LIN, FlexRay). Збір телеметрії традиційно здійснюється через стандартизований інтерфейс OBD-II. Проте існуючі діагностичні архітектури, побудовані на моделі поллінгу («запит-відповідь»), не забезпечують необхідної швидкодії (обмежуючись у середньому 9 запитами на секунду), що ускладнює достовірний моніторинг швидкоплинних газо- і термодинамічних процесів. Отже, виникає актуальне науково-прикладне завдання розробки розподілених інформаційно-керувальних систем, здатних обробляти високошвидкісні потоки даних у режимі реального часу.

Обсяги, швидкість надходження та структура автомобільної телеметрії повністю відповідають парадигмі Big Data. Використання класичних реляційних

СУБД (SQL) для збереження таких даних є неефективним через розрідженість матриць ознак, що зумовлено варіативністю специфікацій різних моделей ТЗ. Це обґрунтовує доцільність застосування нереляційних документо-орієнтованих баз даних (зокрема MongoDB), які працюють за гнучкою концепцією BASE, забезпечуючи адаптивність схем та високу відмовостійкість завдяки горизонтальному масштабуванню. Для забезпечення безперервної маршрутизації та інтеграції потоків необхідне залучення екосистеми Hadoop, брокерів повідомлень (Apache Kafka) та моделі паралельних обчислень MapReduce, що дозволяє реалізувати архітектуру прогностичного моніторингу «на льоту».

Фундаментальною проблемою при розробці систем прогностичної аналітики для автосервісу є критична розрідженість даних (Data Sparsity, понад 25% пропусків) у базах логованої телеметрії (Logged Vehicle Data, LVD), зумовлена нерегулярністю заїздів на СТО. Застосування стандартних методів неповної або надмірної вибірки призводить до втрати статистично значущої інформації або перенавчання моделей. Ефективним шляхом подолання цієї проблеми є впровадження статистичних фільтрів на основі непараметричного критерію Колмогорова-Смірнова в поєднанні з обгортковими алгоритмами пучкового пошуку (beam search), що дозволяє формувати високоякісні навчальні датасети в умовах жорстких обмежень на їх розмір.

Після підготовки репрезентативних вибірок критичним завданням стає класифікація технічного стану з використанням методів машинного навчання. Враховуючи нестаціонарний характер часових рядів OBD-II телеметрії, найвищу апроксимаційну здатність демонструють архітектури глибокого навчання, зокрема багатошаровий перцептрон (MLP) та рекурентні нейронні мережі з довгою короткочасною пам'яттю (LSTM). Водночас, ансамблеві алгоритми (Random Forest) забезпечують підвищену робастність до зашумлених сигналів. З огляду на екстремальний дисбаланс класів у автомобільному сервісі

(де відмови становлять меншість), оптимізація алгоритмів повинна здійснюватися за комплексними метриками (F1-score, коефіцієнт кореляції Метьюса) з фокусом на сувору мінімізацію хибно-негативних прогнозів (False Negatives), що є критичним фактором безпеки.

Впровадження таких розподілених інформаційних технологій має не лише локальний економічний ефект для СТО, але й виступає інфраструктурним базисом для розвитку концепції розумного міста (Smart City). Використання масивів телеметрії (Floating Car Data) в інтеграції з системами V2X забезпечує створенню інформаційно-керувальних систем для динамічного керування транспортними потоками та запобігання заторам. Таким чином, розробка комплексної розподіленої технології моніторингу технічного стану ТЗ на базі Big Data є своєчасним, науково обґрунтованим і практично необхідним кроком для автомобільної галузі.

Зв'язок роботи з науковими програмами, планами, темами.

Дисертаційна робота виконана відповідно до тематичних планів наукових досліджень кафедри комп'ютерних наук та інформаційних систем Харківського національного автомобільно-дорожнього університету Міністерства освіти і науки України у межах науково-дослідної тематики відповідно до завдань держбюджетних тем МОН України: «Застосування інформаційних технологій в сфері цифровізації транспортних систем» (ДР № 0124U004241) і безпосередньо узгоджуються з пріоритетами національної транспортної стратегії та концептуальними засадами Національної стратегії Міністерства інфраструктури України «Drive Ukraine 2030» щодо впровадження інноваційних технологій обробки великих масивів інформації для розбудови інтелектуальних транспортних систем.

Мета та завдання дослідження. Мета дослідження полягає у підвищенні загальної ефективності технічного обслуговування транспортних засобів шляхом розробки методів, моделей та інформаційних технологій аналізу даних

для прогностичного моніторингу технічного стану транспортних засобів, що забезпечує науково обґрунтований перехід від реактивного до проактивного прогностичного обслуговування (Predictive Maintenance).

Для досягнення поставленої мети необхідно розв'язати наступні **завдання**:

1. Здійснити системний аналіз наявних методів діагностики технічного стану транспортних засобів і дослідити еволюцію систем управління базами даних – від реляційних моделей до екосистем Big Data – у контексті розвитку концепції Smart City.
2. Дослідити принципи збору телеметричної інформації через інтерфейс OBD-II та розробити методологію подолання розрідженості даних у базах Logged Vehicle Data (LVD) за рахунок використання статистичних фільтрів для підвищення якості навчальних вибірок.
3. Побудувати, налаштувати та порівняти ефективність математичних моделей машинного навчання (багатошаровий перцептрон, рекурентні мережі LSTM, Random Forest, логістична регресія) для класифікації технічного стану ТЗ із фокусом на мінімізацію хибно-негативних прогнозів.
4. Розробити об'єктно-орієнтовану UML-модель і спроектувати відмовостійку архітектуру програмного комплексу, що інтегрує нереляційні документо-орієнтовані СУБД (MongoDB) і процеси паралельної обробки (MapReduce) для роботи в режимі реального часу.

Об'єктом дослідження є інформаційні процеси обробки та аналізу діагностичних даних о технічному стані транспортних засобів.

Предметом дослідження є методи, моделі та інформаційні технології прогностичного моніторингу стану транспортних засобів.

Мета дослідження – підвищення ефективності технічного обслуговування транспортних засобів шляхом розробки методів, моделей та

інформаційних технологій аналізу даних для прогностичного моніторингу технічного стану транспортних засобів, що забезпечує науково обґрунтований перехід від реактивного до проактивного обслуговування.

Методи дослідження. Методологічна база дослідження ґрунтується на застосуванні методів системного аналізу, об'єктно-орієнтованого проєктування (UML), теорії нереляційних баз даних (NoSQL) і технологій паралельних обчислень MapReduce для побудови масштабованих архітектур збору й обробки телеметричних даних. Для аналітичного моделювання та класифікації технічного стану використано методи математичної статистики (зокрема, непараметричний критерій Колмогорова-Смірнова), а також сучасний апарат штучних нейронних мереж (багатошаровий перцептрон, LSTM) і ансамблеві алгоритми машинного навчання.

Наукова новизна одержаних результатів полягає в тому, що :

Вперше розроблено розподілену інформаційну технологію дистанційного контролю транспортних засобів, яка, на відміну від відомих реляційних систем, інтегрує нереляційні NoSQL-бази даних (MongoDB), паралельні обчислення MapReduce та потокову обробку через брокери повідомлень (Apache Kafka). Застосування такого підходу вперше обґрунтовується необхідністю відмови від жорстко структурованих реляційних SQL-схем, що працюють за традиційною та повільною моделлю «запит-відповідь», на користь гнучкої гетерогенної архітектури (BASE), здатної обробляти швидкі нестаціонарні потоки телеметрії. Це забезпечує консистентність даних і горизонтальну масштабованість за умов нерегулярного надходження діагностичної інформації (Logged Vehicle Data, LVD).

Удосконалено метод подолання розрідженості навчальних вибірок баз даних LVD шляхом застосування статистичних оцінок щільності розподілу ймовірностей (за рахунок використання статистичного фільтра на основі непараметричного критерію Колмогорова-Смірнова) та алгоритмів мінімізації

вtrat даних при розширенні діагностичного простору ознак (із використанням алгоритму пучкового пошуку – beam search – для поетапного додавання релевантних ознак з урахуванням граничних умов розміру датасету), що підвищує якість формування навчальних датасетів для задач прогностичної аналітики.

Отримало подальший розвиток застосування архітектур глибокого навчання – багатошарового перцептрона (MLP) та рекурентних мереж довгої короткочасної пам'яті (LSTM) для виявлення прихованих дефектів у нестационарних потоках телеметрії OBD-II; порівняльним аналізом доведено переваги ансамблевих методів (Random Forest) над логістичною регресією за метриками F1-score та Matthews Correlation Coefficient (MCC) при класифікації технічного стану транспортних засобів.

Практичне значення одержаних результатів полягає в тому, що розроблена система дозволяє СТО та сервісним центрам перейти від реактивної моделі ремонту до прогностичної аналітики, що оптимізує логістику запасних частин, мінімізує час простою комерційного транспорту та знижує ризик відмов критичних вузлів. Програмний і математичний апарат може бути інтегрований у системи розумного міста (Smart City) для комплексної оптимізації транспортних потоків та екологічного моніторингу. Результати роботи узгоджуються з пріоритетами національної транспортної стратегії.

Особистий внесок здобувача. Основні положення і результати дисертаційної роботи, що представлено до захисту, отримані здобувачем особисто. Серед них: аналіз науково-технічної літератури, проведення практичних досліджень, аналіз отриманих результатів. Постановку мети і завдань досліджень, аналіз та обговорення одержаних результатів виконано спільно з науковим керівником професором, к.т.н. Проніним С.В.

Апробація результатів дисертації. Результати наукового пошуку Грицука В.Ю. обговорювалися на V міжнародній науково-методичній

конференції «Комп'ютерні технології і мехатроніка» у 2023 році, Міжнародній науково-практичній конференції «Вища технічна освіта XXI століття: виклики, проблеми, перспективи» у 2022, 2023, 2024 роках, міжнародних конференціях «Енергетичні установки та альтернативні джерела енергії» у 2024, 2025, 2026 роках, XVI Міжнародній науково-практичній конференції «Сучасні інформаційні та інноваційні технології на транспорті (MINTT – 2024)», VIII міжнародній науково-практичній конференції «Науково-прикладні аспекти автомобільної і транспортно-дорожньої галузей», Міжнародній науково-практичній конференції «Інноваційні підходи у відновленні транспортної інфраструктури в особливих умовах воєнного стану: виклики і перспективи», міжнародних наукових конференціях «Математичне моделювання та інформаційні технології сучасності», «Процеси цифровізації екосистем».

Публікації. Результати дисертаційної роботи опубліковано у 20 наукових працях, з яких: 2 статті у наукових фахових виданнях України; 2 статті, що входять до наукометричних баз Scopus та Web of Science; 16 публікацій у матеріалах конференцій.

Структура та обсяг дисертації. Дисертаційна робота складається з анотації двома мовами, змісту, вступу, основної частини, з чотирьох розділів, висновків, списку використаних джерел, додатку. У роботі представлено 64 рисунки і 20 таблиць. Загальний обсяг дисертації складає 248 сторінок, з них: 218 основного тексту, список використаних джерел охоплює 27 сторінок. Список використаних джерел складає 219 найменувань.

РОЗДІЛ 1 АНАЛІЗ МЕТОДІВ ТА ІНФОРМАЦІЙНИХ СИСТЕМ ДІАГНОСТИКИ, ОЦІНКИ ТА ПРОГНОЗУВАННЯ ТЕХНІЧНОГО СТАНУ

1.1. Огляд технологій Big Data

Сучасні умови експлуатації транспортних засобів (ТЗ) вимагають безперервного контролю їхнього просторового положення, параметрів фактичного технічного стану та своєчасного проведення сервісних дій. Відповідно, використання новітніх інформаційних технологій для дистанційного моніторингу та аналізу працездатності ТЗ є актуальним науково-практичним завданням. Оскільки ефективність функціонування автомобіля як складної технічної системи безпосередньо залежить від її стану [1-3], виникає потреба в управлінні процесами експлуатації на основі агрегованих масивів даних.

Як зазначається у [1-3], фундаментальну роль у розробці автоматизованих систем управління транспортними підприємствами відіграють бази даних. Вони дозволяють збирати інформацію про експлуатаційні параметри автомобілів у логістичних ланцюгах, оцінювати надійність роботи на маршрутах і прогнозувати попит. Традиційно для цих завдань використовувалися реляційні бази даних, які, за висновками авторів [4, 5], забезпечують необхідний рівень абстракції для структурування інформації. Згідно зі статистикою [6], реляційні моделі досі залишаються найпоширенішим інструментом для вирішення багатьох галузевих завдань (рис. 1.1).

Однак традиційні бази даних орієнтовані на роботу з чітко організованою та структурованою інформацією. Експоненційне зростання обсягів даних, значна частина яких генерується бортовими системами у режимі реального часу, зумовило перехід до технологій Big Data. Сам термін, запроваджений у

2008 році К. Лінчем [7], описує сукупність архітектурних підходів та інструментів для обробки великих обсягів структурованих, напівструктурованих і неструктурованих даних.

DB-Engines Ranking

The DB-Engines Ranking ranks database management systems according to their popularity. The ranking is updated monthly.

Read more about the [method](#) of calculating the scores.



381 systems in ranking, December 2021

Rank			DBMS	Database Model	Score		
Dec 2021	Nov 2021	Dec 2020			Dec 2021	Nov 2021	Dec 2020
1.	1.	1.	Oracle +	Relational, Multi-model	1281.74	+9.01	-43.86
2.	2.	2.	MySQL +	Relational, Multi-model	1206.04	-5.48	-49.41
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model	954.02	-0.27	-84.07
4.	4.	4.	PostgreSQL +	Relational, Multi-model	608.21	+10.94	+60.64
5.	5.	5.	MongoDB +	Document, Multi-model	484.67	-2.67	+26.95
6.	6.	↑ 7.	Redis +	Key-value, Multi-model	173.54	+2.04	+19.91
7.	7.	↓ 6.	IBM Db2	Relational, Multi-model	167.18	-0.34	+6.74
8.	8.	8.	Elasticsearch	Search engine, Multi-model	157.72	-1.36	+5.23
9.	9.	9.	SQLite +	Relational	128.68	-1.12	+7.00
10.	↑ 11.	↑ 11.	Microsoft Access	Relational	125.99	+6.75	+9.25

Рисунок 1.1 – Статистика використання баз даних

Фундаментальна парадигма роботи з великими даними базується на концепції 3V: Volume (обсяг), Velocity (швидкість генерації та обробки) та Variety (різноманітність типів даних). Сучасні дослідники розширюють цю модель до 5V, додаючи Veracity (достовірність) та Value (цінність) [8]. З точки зору аналітики, чим більший і якісніший масив інформації зібрано про об'єкт дослідження, тим вищою є точність прогностичних моделей.

Технологічний стек для роботи з Big Data у автосервісній галузі вимагає горизонтальної масштабованості, відмовостійкості та забезпечення локальності даних. Для цього використовуються нереляційні бази даних (NoSQL), паралельні обчислення на базі моделі MapReduce, компоненти кластерної

екосистеми Hadoop [9], а також спеціалізовані платформи екосистеми Apache (Spark, AirFlow, Kafka, HBase) [10, 11] та мови програмування R і Python.

Інтеграція технологій Big Data в автомобільну галузь набуває системного характеру. Так, Національна транспортна стратегія Drive Ukraine 2030 [12] передбачає використання цих інструментів для розбудови інтелектуальних систем управління трафіком. В контексті автомобільного сервісу ключовими масивами для аналізу виступають:

- телеметричні дані: параметри роботи двигуна, гальмівної системи, систем пасивної та активної безпеки, що збираються електронними блоками управління (есу) та бортовими датчиками;
- просторово-часові дані: інформація від gps-трекерів щодо швидкості, пройденої відстані, маршрутів та геолокації автомобіля;
- поведінкові дані (дані водіїв): вік, стаж, стиль водіння, історія правопорушень або аварійних ситуацій;
- історія сервісного обслуговування: відомості про виконані регламентні роботи, дати ремонтів, типи замінених запчастин і специфічні технічні характеристики автомобіля.

Залучення методів аналізу цих масивів (рис. 1.2) дозволяє автосервісним компаніям переходити від реактивного до прогностичного обслуговування, виявляти приховані патерни у відмовах агрегатів і оптимізувати клієнтський досвід.

Окремим перспективним вектором є взаємодія і поєднання Big Data з концепціями V2I (Vehicle-to-Infrastructure) та V2X (Vehicle-to-Everything) [2, 3, 13-15]. Підключені транспортні засоби (Connected Vehicles) утворюють єдину екосистему з міською інфраструктурою, генеруючи інформацію для геоінформаційних систем (ГІС) і моделей прогнозування трафіку.

Особливе значення технології великих даних мають для розробки та вдосконалення систем автономного водіння [16-21]. Штучні нейромережі та

алгоритми глибокого навчання (Deep Learning) обробляють терабайти даних із сенсорів (лідарів, радарів, відеокамер) для розпізнавання об'єктів, прогнозування дорожньої ситуації та забезпечення безпеки пасажирів у непередбачуваних умовах.



Рисунок 1.2 – Методи аналізу та обробки Big Data

Незважаючи на широкі можливості, впровадження Big Data в автомобільному сервісі стикається з низкою принципових технічних та організаційних викликів [15, 22], які можна класифікувати за такими напрямками:

1. Апаратно-обчислювальні обмеження: невідповідність наявних потужностей швидкості експоненційного зростання обсягів даних, що

призводить до зниження швидкодії систем і потребує значних інфраструктурних витрат (зокрема на хмарні сховища).

2. Проблеми якості та релевантності даних: наявність зашумлених, неповних або недостовірних даних, що вимагає впровадження складних алгоритмів попередньої обробки для уникнення хибних аналітичних висновків.

3. Кібербезпека та конфіденційність: високі ризики несанкціонованого доступу до телеметрії та персональних даних клієнтів, що вимагає застосування надійних протоколів аутентифікації та наскрізного шифрування.

4. Етичні та правові аспекти: необхідність регулювання процесів збору поведінкової інформації водіїв для запобігання дискримінації.

Таким чином, використання технологій Big Data в автомобільному сервісі є складним, ресурсномістким процесом. Проте саме ефективна побудова архітектури збору, зберігання та інтелектуальної обробки даних виступає головним драйвером підвищення надійності транспортних засобів, оптимізації ремонтних процесів та розвитку екосистеми розумного транспорту.

1.2. Прогностичне обслуговування в автомобільному сервісі

Еволюція підходів до підтримки працездатності транспортних засобів (ТЗ) демонструє стійку тенденцію до мінімізації незапланованих простоїв та оптимізації витрат. На сьогоднішній день на автотранспортних та сервісних підприємствах застосовуються три базові стратегії технічного обслуговування [23, 24]:

- реактивне обслуговування (run-to-failure, rtf): стратегія, за якої відновлювальні роботи проводяться виключно після фактичної відмови вузла чи агрегату. цей метод економічно доцільний лише для некритичного обладнання, відмова якого не впливає на загальну безпеку та продуктивність. у контексті

складних систем тз такий підхід призводить до високих витрат на ремонт та значних простоїв;

- планово-попереджувальне обслуговування (preventive maintenance, pvm): стратегія, що базується на регламентованих часових інтервалах або пробігу. вона передбачає виконання запланованих дій для запобігання відмовам до їх виникнення. хоча цей підхід знижує ймовірність раптових поломок, він часто призводить до надлишкового обслуговування та передчасної заміни компонентів, які ще не вичерпали свій ресурс;
- прогностичне обслуговування (predictive maintenance, pdm): інноваційна стратегія, що використовує технології інтернету речей (iot) для безперервного моніторингу фактичного технічного стану автомобіля в режимі реального часу. на основі зібраних даних застосовується прогностична аналітика для точного прогнозування моменту, коли знадобиться втручання.

Аналіз робіт [23-25] дозволяє згрупувати переваги та недоліки (обмеження) прогностичного обслуговування (табл. 1.1).

Впровадження систем прогностичного обслуговування має високу актуальність для України, оскільки дозволяє оптимізувати експлуатаційні витрати комерційних автопарків, подовжити життєвий цикл компонентів ТЗ та суттєво підвищити рівень активної безпеки на дорогах за рахунок раннього виявлення прихованих дефектів.

Архітектура системної взаємодії в рамках PdM (рис. 1.3) складається з трьох фундаментальних рівнів: генерації та збору даних (на борту ТЗ), їх передачі та зберігання (хмарна інфраструктура), а також аналізу та візуалізації (аналітичне програмне забезпечення). Передача телеметричної інформації від автомобіля до серверної частини забезпечується сучасними протоколами бездротового зв'язку (4G/5G). Прикладом реалізації такого підходу на макрорівні є концепції стандартизації дистанційної діагностики, такі як OBD-III у США або RTM у Китаї [25, 26].

Таблиця 1.1 – Переваги та недоліки (обмеження) прогностичного обслуговування

Переваги (можливості) прогностичного обслуговування	Недоліки (обмеження) прогностичного обслуговування
Превентивне виявлення механічних деградацій до їх перетворення на критичні відмови.	Висока залежність від наявності репрезентативних, очищених та структурованих масивів даних.
Максимізація корисного ресурсу (терміну служби) автомобільних компонентів та вузлів.	Потреба у значних обчислювальних потужностях для розробки та навчання надійних прогностичних моделей.
Безперервний моніторинг технічного стану агрегатів у режимі реального часу.	Ризик генерування хибних прогнозів через зашумленість або неповноту вхідних даних (що веде до невиправданих витрат).
Підтримання ТЗ в оптимальному стані, що підвищує їх залишкову ринкову вартість та безпеку експлуатації.	Вразливість каналів передачі телеметрії; необхідність забезпечення кібербезпеки та конфіденційності.
Можливість проактивної комунікації дилерів та СТО з автовласниками, що підвищує рівень клієнтської лояльності.	Високі вимоги до якості апаратної частини (сенсорів) та математичної складності алгоритмів обробки.

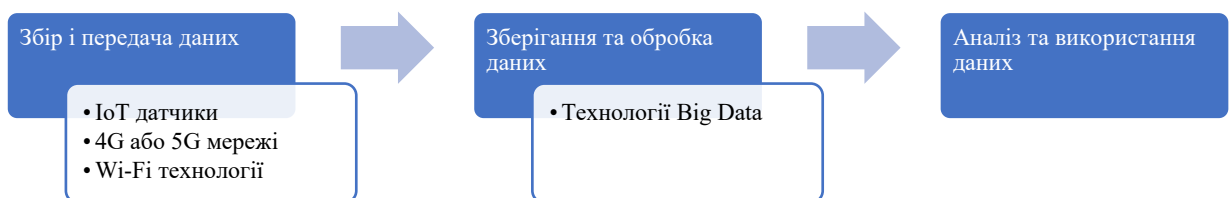


Рисунок 1.3 – Технології залучені в прогностичне обслуговування

Системну взаємодію в прогностичному обслуговуванні можна представити у вигляді схеми, що містить основні блоки (збір і передача даних, зберігання та обробка даних, аналіз і використання) (рис. 1.4).

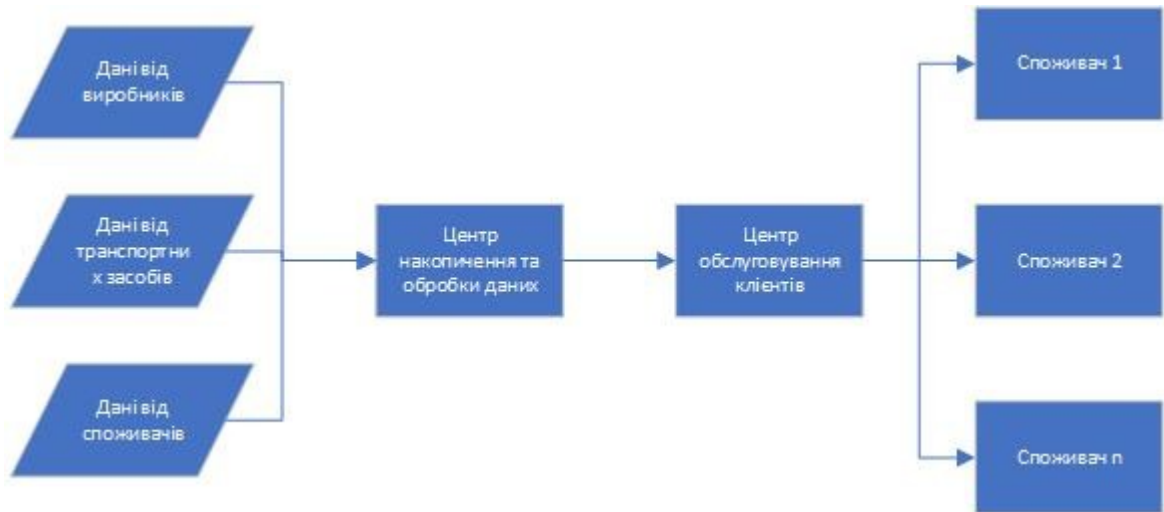


Рисунок 1.4 – Системна взаємодія прогностичного обслуговування

В системній взаємодії в прогностичному обслуговуванні транспортних засобів (рис. 1.4) можна виділити основні блоки (збір і передавання даних, зберігання та обробка даних, аналіз і використання). Блок центру накопичення та обробки даних може становити собою реляційну базу даних (або бути елементом більш складної структури).

Для побудови ефективних аналітичних моделей широко використовуються структуровані набори даних (датасети) – як синтезовані, так і зібрані в реальних умовах експлуатації. Показовим прикладом є масиви телеметрії, підготовлені Національною адміністрацією безпеки дорожнього руху США (NHTSA) [27], які застосовуються для тестування алгоритмів машинного навчання (рис. 1.5).

Як зазначається в дослідженнях [2, 3, 28], одним із перспективних векторів розвитку інтелектуальних транспортних систем є інтеграція сервісів дистанційного контролю параметрів ТЗ (рис. 1.6). Фундаментальним елементом такого сервісу є бортова діагностична система (On-Board Diagnostics, OBD), яка

об'єднує електронні блоки керування (ECU), датчики та виконавчі механізми через цифрову шину даних (зокрема, CAN-шину).

Record ID	NHTSA Campaign	Manufacturer Ca...	Vehicle Make	Vehicle Model	# Model
 1124k	20684 unique values	[null] 66% 215-D 0% Other (42023) 34%	FORD 5% CHEVROLET 4% Other (112820) 91%	15164 unique values	 1949
1	02V288000	02S41	FORD	FOCUS	2001
2	02V288000	02S41	FORD	FOCUS	2000
3	02V236000		JAYCO	FT EAGLE 10 SG	2003
4	02V237000		HOLIDAY RAMBLER	ADMIRAL SE	2003
5	02V237000		HOLIDAY RAMBLER	ADMIRAL SE	2002
6	02V237000		HOLIDAY RAMBLER	ADMIRAL	1999
7	02V237000		MONACO COACH	MONARCH SE	2003
8	02V237000		MONACO COACH	MONARCH SE	2002
9	02V237000		MONACO COACH	LAPALMA	1999
10	02V237000		MONACO COACH	LAPALMA	2003

Рисунок 1.5 – Фрагмент датасету, підготовленого Національною адміністрацією безпеки дорожнього руху США (NHTSA)

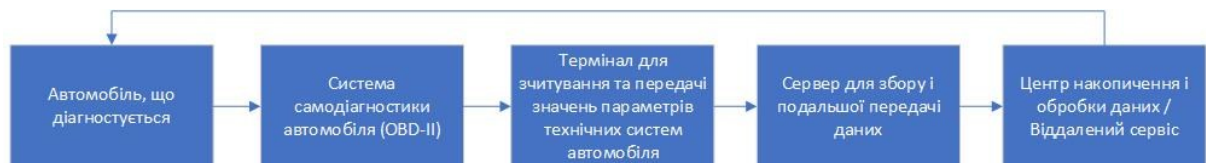


Рисунок 1.6 – Схема взаємодії елементів сервісу віддаленого контролю параметрів технічних систем транспортних засобів (за [2], трансформована)

Сучасні OBD-сканери генерують безперервний потік даних, забезпечуючи доступ до широкого спектра параметрів [29]:

- динамічні показники в реальному часі: частота обертання колінчастого вала (rpm), швидкість руху, абсолютне положення дросельної заслінки, кут

випередження запалювання, масова витрата повітря, температура охолоджувальної рідини тощо;

- діагностичні коди несправностей (dtc): стандартизовані помилки, що фіксуються електронними блоками;
- «заморожений кадр» (freeze frame): знімок поточних параметрів роботи систем автомобіля в момент фіксації критичної помилки;
- статуси емісійних систем: результати моніторингу кисневих датчиків та каталітичних нейтралізаторів;
- ідентифікаційні дані: vin-код, кількість циклів запалювання, версії калібрування пз.

Завдяки високій частоті дискретизації та значному обсягу, дані стандарту OBD повноцінно класифікуються як компоненти Big Data. Їх застосування в синергії з алгоритмами машинного навчання (Machine Learning) реалізується на кількох рівнях:

1. Прогнозування відмов агрегатів: використання історичних кодів DTC та параметрів заморожених кадрів для побудови класифікаційних моделей, здатних передбачати необхідність сервісного втручання ще до активації індикатора «Check Engine» [30].

2. Профілювання поведінки водія: агрегація понад 50 динамічних параметрів (швидкість, інтенсивність прискорення/гальмування, розрахункове навантаження на двигун) для розробки алгоритмів скорингу стилю водіння. Це дозволяє надавати персоналізовані рекомендації для зниження витрати палива та мінімізації зносу деталей [31, 32].

3. Прогнозування кінематики ТЗ: застосування рекурентних нейронних мереж, зокрема архітектури довгої короткострокової пам'яті (LSTM), для надточного прогнозування швидкості, курсу та траєкторії автомобіля в мікроінтервалах часу (наприклад, на 1 секунду вперед), що є критично важливим для систем допомоги водієві (ADAS) [33].

Таким чином [34], телеметрична інформація, отримана через інтерфейс OBD, формує надійний емпіричний базис для технологій Big Data. Її обробка за рахунок використання алгоритмів машинного навчання стає ядром систем прогностичного обслуговування, перетворюючи автомобільний сервіс із реактивної галузі на проактивну інтелектуальну екосистему.

1.3. Застосування Big Data для прогнозування

Стрімка урбанізація та зростання інтенсивності транспортних потоків призводять до перманентних заторів, що безпосередньо впливає на умови експлуатації транспортних засобів (ТЗ). Складні режими руху (часті прискорення та гальмування, тривала робота двигуна на холостому ходу) прискорюють знос вузлів автомобіля [35, 36]. Відповідно, використання комплексних наборів даних про дорожній рух стає не лише інструментом управління трафіком, але й критично важливим джерелом інформації для прогностичного автомобільного сервісу. Розуміння динаміки руху [37-39] дозволяє сервісним підприємствам точніше прогнозувати потребу в обслуговуванні, а також оптимізувати власну логістику (наприклад, маршрутизацію автомобілів технічної допомоги).

Ефективним інструментом збору агрегованих даних є метод Floating Car Data (FCD) (рис. 1.7), який базується на отриманні інформації в режимі реального часу безпосередньо від рухомих транспортних засобів, що виступають у ролі мобільних датчиків. На відміну від стаціонарних систем моніторингу, FCD генерує масиви геопросторових даних із часовими мітками (швидкість, координати, напрямок руху) [40].

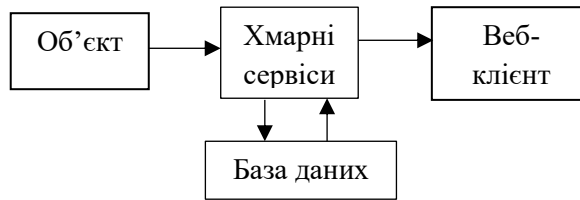


Рисунок 1.7 – Модель FCD

Ці дані дозволяють не лише оптимізувати алгоритми керування трафіком, але й оцінити фактичне експлуатаційне навантаження на автомобілі в певних міських зонах. Інтеграція потоків FCD у режимі реального часу з автоматизованими системами автосервісних мереж (рис. 1.8) дає змогу формувати динамічний профіль умов експлуатації, що є основою для коригування регламентів технічного обслуговування [41].

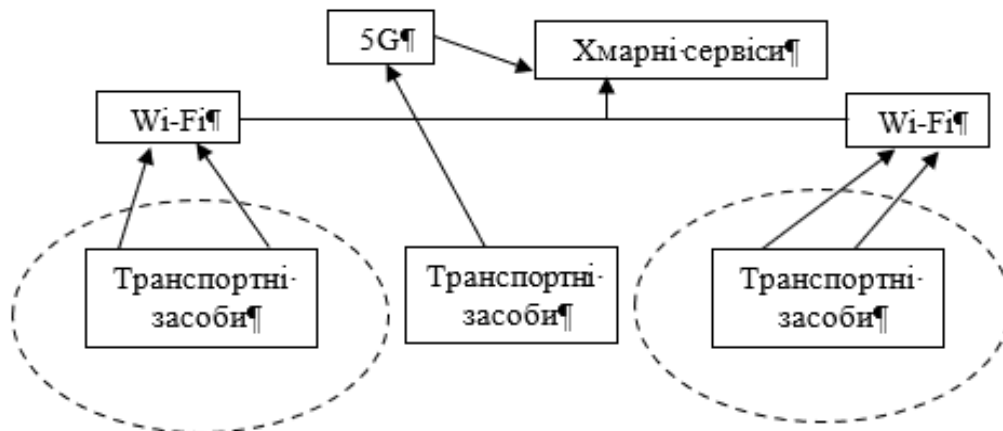


Рисунок 1.8 – Підтримка розширених даних про FCD

Використання технологій Інтернету речей (IoT) та машинного навчання (ML) дозволяє перетворити сирі дані про рух на прогностичні моделі [42]. Безперервні потоки даних від дорожніх сенсорів, камер та підключених автомобілів є основою для навчання ML-алгоритмів.

У контексті автомобільного сервісу та логістики машинне навчання виконує подвійну функцію. З одного боку, алгоритми ідентифікують тригери заторів та прогнозують дорожні умови, з іншого – навігаційні системи,

інтегровані з цими моделями, здатні завчасно генерувати альтернативні маршрути. Це мінімізує час перебування ТЗ у заторах, що безпосередньо знижує теплове та механічне навантаження на агрегати автомобіля й подовжує їхній ресурс [40].

Одна з особливостей машинного навчання – це акцент на потенціалі стратегій зміни маршруту, що сприяють отриманню даних у реальному часі. Маючи в основі набори даних із підтримкою машинного навчання, навігаційні системи стають здатними не просто реагувати на затори, а й завчасно змінювати маршрути транспортних засобів для оптимізації потоку транспорту. На сьогоднішній день навігаційні системи, озброєні статистичними даними в реальному часі, можуть динамічно коригувати маршрути, відводячи водіїв від заторів і зменшуючи тиск у міській дорожній мережі. Цей підхід не лише мінімізує затримки для окремих пасажирів, а й сприяє більш ефективному використанню всієї транспортної інфраструктури [40].

По суті, поєднання аналітики даних, машинного навчання та Інтернету речей змінює стратегії управління трафіком. Створення точних наборів даних за рахунок використання алгоритмів машинного навчання стає основою для активного прогнозування трафіку та швидкої зміни маршруту, завдяки оптимізуванню нових технологій.

Концепція розумного міста (Smart City) генерує колосальні обсяги даних, які можуть бути інтегровані в бізнес-процеси автосервісних компаній. Основні вектори застосування аналітики Big Data включають:

- оптимізація маршрутизації: динамічний вибір маршрутів з огляду на історичні та поточні дані про завантаженість;
- адаптивне керування інфраструктурою: регулювання фаз світлофорів для максимізації пропускної здатності;
- розумне паркування (smart parking): моніторинг доступності паркомісць, що зменшує час роботи двигуна під час пошуку вільного місця;

- логістика екстрених та сервісних служб: використання прогностичних моделей для швидкого доступу евакуаторів або мобільних ремонтних бригад до місця призначення.

Поєднання даних із різних джерел (V2X, дорожня інфраструктура, смартфони) формує цілісне уявлення про ландшафт міської мобільності, що дозволяє сервісним компаніям ухвалювати data-driven рішення.

Основною проблемою під час створення масивів даних (датасетів) є забезпечення їхньої релевантності, точності та конфіденційності. Надійна методологія вимагає впровадження протоколів анонімізації телеметрії та інтеграції гетерогенних джерел даних у єдину узгоджену структуру.

Прикладом ефективного використання масивів даних є впровадження систем автоматизованого управління сигналами (ATSPM) у Лос-Анджелесі (рис. 1.9) [43]. Агрегація даних світлофорів, IoT-сенсорів, погодних умов та підключених ТЗ дозволила динамічно оптимізувати трафік.

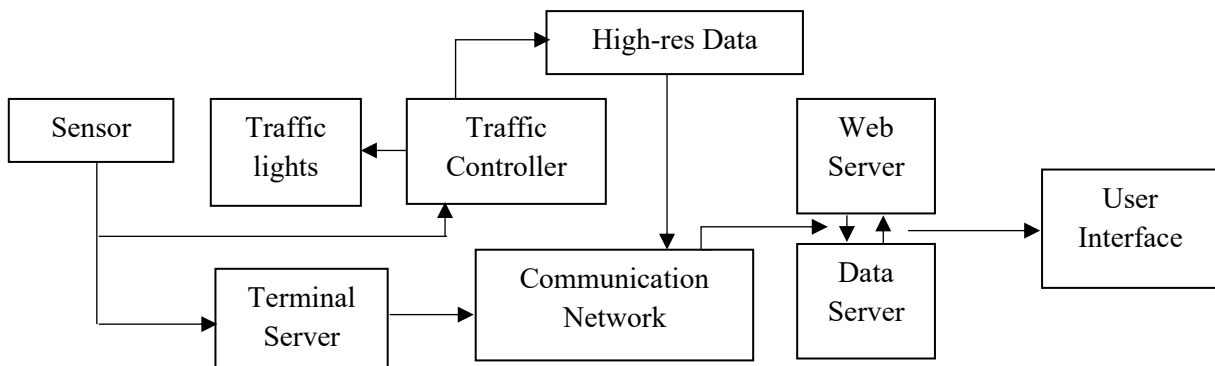


Рисунок 1.9 – Принцип роботи ATSPM

Враховуючи високу розмірність, швидкість надходження та різноманітність форматів (3V) таких даних, їх обробка потребує використання сучасної хмарної інфраструктури та гібридних баз даних. Структуровані метрики (наприклад, логування сигналів, історичні табличні дані) доцільно зберігати у реляційних СУБД (SQL) [1], тоді як неструктуровані масиви

(відеопотоки, сирі логи сенсорів) потребують нереляційних рішень (NoSQL) або об'єктних сховищ рівня Data Lake (наприклад, Amazon S3).

Для забезпечення узгодженості інформації налаштовуються конвеєри даних – ETL-процеси (Extract, Transform, Load) (рис. 1.10). Потік даних проходить етапи екстракції з різноманітних API, валідації (порівняння із заданими патернами), очищення від аномалій та завантаження у Сховище даних (Data Warehouse). Для обробки масивів у реальному часі застосовуються інструменти розподілених обчислень (Apache Spark, Apache Flink), що дозволяє паралельно обчислювати агреговані показники (середня швидкість, час затримки) без критичного навантаження на систему.

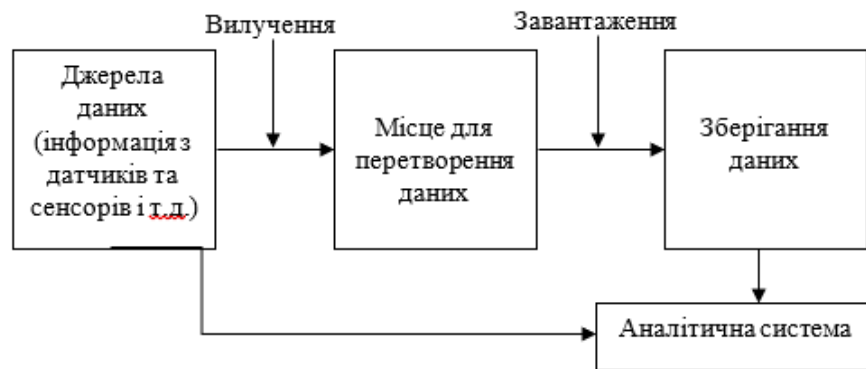


Рисунок 1.10 – Модель ETL

З метою оптимізації доступу до даних та зменшення надлишкового фізичного копіювання ефективним рішенням є віртуалізація даних (Data Virtualization) [44, 45]. Це створює єдиний логічний шар доступу до розрізнених джерел, що прискорює аналітичні процеси та спрощує інтеграцію додаткового програмного забезпечення для потреб автосервісних підприємств.

На сьогоднішній день, можна констатувати, що розвиток систем збору та аналізу транспортних даних базується на кількох ключових технологічних драйверах:

- мережі 5g, які забезпечують надвисоку швидкість та мінімальну затримку (low latency) передачі даних, що є критичним для функціонування v2x-комунікацій і потокової аналітики в реальному часі;
- удосконалені сенсорні мережі: інтеграція lidar, радарів та систем комп'ютерного зору значно підвищує гранулярність і достовірність просторових даних;
- екосистема iot: глибинна інтеграція пристроїв інтернету речей у конструкцію тз та дорожню інфраструктуру створює безперервний інформаційний контур;
- глибоке навчання (deep learning): перехід від базової аналітики до створення складних нейромережових моделей, здатних до самоадаптації в умовах мінливої динаміки міського середовища.

При використанні Big Data постійно існує потреба в удосконаленні процесів ухвалення рішень. Оскільки кількість інформації зростає з кожним днем, то вирішення цих потреб є однією з ключових потреб. Одним з варіантів моделі прийняття рішень є модель OLAP (OnLine Analytical Processing). Вона характеризується швидким аналізом великих обсягів даних у багатовимірних просторах, що у свою чергу може забезпечити швидку агрегацію даних, а саме можливість об'єднувати дані за різними категоріями: час, тип транспортного засобу, регіон і т.д. Також ця модель дозволяє аналізувати й виявляти тенденції та зразки у великих обсягах даних, що стосуються обслуговування та ремонту автомобілів. Одна з ключових проблематик в якій можна це застосувати - сервісні центри постійно стикаються з проблемами неефективного управління запасами, незадовільної якості обслуговування клієнтів та високою тривалістю ремонтних робіт, що призводить до зниження задоволеності клієнтів та втрати доходів. Щоб вирішити це, потрібно знайти певні закономірності, основні тенденції та ключові зони розвитку в такій моделі:

1. Збір даних. Усі дані з кожного сервісного центру збираються в окрему основну базу даних. Ці дані включатимуть інформацію про всіх клієнтів, їхні автомобілі, проведені ремонтні роботи, використані запчастини, тривалість ремонтів, відгуки клієнтів.

2. OLAP аналіз. За рахунок використання OLAP кубів (певних багатовимірних структурах) які дозволяють легко виконувати аналіз за різними вимірами, такими як тип автомобіля (легковий, вантажний, SUV), частота сервісних відвідувань, і сума витрат на послуги. Виходячи з цього маємо такі параметри цього аналізу:

- сегментація клієнтів: клієнти класифікуються за типом автомобіля, частотою відвідувань, та витратами на ремонт, що дозволяє розробити цільові маркетингові стратегії;
- аналіз запасів: визначення, які запчастини мають високий попит за рахунок історичних даних, щоб оптимізувати закупівлі та зберігання запасів;
- тривалість ремонтів: аналізується час, витрачений на різні типи ремонтних робіт, щоб виявити процедури, які вимагають непропорційно багато часу;
- якість послуг: використовуючи olap для аналізу відгуків клієнтів і звернень щодо сервісу, щоб виявити схожість повторюваних проблем із певними видами робіт або запчастинами.

3. Впровадження змін. На основі отриманих даних можна впровадити певні зміни для оптимізації роботи:

- персоналізоване обслуговування: програми лояльності та персоналізовані пропозиції для постійних клієнтів;
- оптимізація запасів: керування залишками стає більш гнучким і відповідальним, зменшуються витрати на зберігання надлишкових товарів, покращується швидкість реакції на різкі зміни на ринку;

- покращення процесів ремонту: навчання персоналу та внесення змін у робочі процеси для скорочення тривалості ремонтів;
- стратегічне планування: результати такого аналізу допомагають робити обґрунтовані рішення про розширення сервісних послуг, відкриття нових локацій або впровадження інноваційних технологій.

Комплексна система підтримки прийняття рішень (DSS) агрегує інформацію з цих і інших джерел, безпосередньо впливаючи на ефективність управління. Однак побудова прогностичних моделей у DSS вимагає детального попереднього дослідження структури даних.

Оскільки така система не може бути ідеально спланованою, потрібні додаткові параметри роботи та моделі, за якими здійснюватиметься аналіз і розв'язання проблеми. Одним з ключових методів аналізу є експлораторний аналіз даних (Exploratory data analysis – EDA). Він включає аналіз закономірностей зв'язків між змінними, коли відсутні варіанти їх взаємодії.

Відповідно, обробляючи взаємозв'язки між змінними та можливі аномалії у даних за рахунок використання візуалізацій та статистичних методів можна визначити найважливіші змінні для використання у даній моделі за рахунок використання кореляційного аналізу або більш складних методів вибору змінних, таких як методи машинного навчання.

1.4. Постановка задачі дослідження

Проведений в пп.1.1–1.3 аналіз дозволяє констатувати наукове протиріччя між зростаючою потребою автомобільного сервісу в переході до проактивного прогностичного обслуговування на основі парадигми великих даних та обмеженою здатністю існуючих інформаційних систем забезпечити такий перехід: традиційна модель опитування OBD-II «запит-відповідь» не має достатньої пропускну здатності для потокового збору даних, реляційні СУБД

не пристосовані до розрідженості та гетерогенності баз Logged Vehicle Data (LVD), а базові алгоритми класифікації втрачають надійність в умовах екстремального дисбалансу класів, характерного для вибірок технічного стану ТЗ. Виходячи з цього, науково-прикладна задача дослідження полягає у розробці розподіленої інформаційної технології дистанційного контролю технічного стану транспортних засобів на основі методів великих даних та машинного навчання, яка забезпечує обґрунтований перехід від реактивного до проактивного обслуговування.

Розв'язання цього протиріччя вимагає послідовного розв'язання таких задач:

1. Систематизувати функціональні режими OBD-II (Live Data, Freeze Frame, DTC) з погляду придатності для машинного навчання та обґрунтувати перехід від моделі циклічного опитування «запит-відповідь» до потокового збору даних (Data Streaming), що є необхідною передумовою для формування якісних навчальних вибірок на наступних етапах.

2. Розробити методологію подолання розрідженості навчальних вибірок на основі статистичного фільтра за непараметричним критерієм Колмогорова-Смірнова та обгорткового алгоритму пучкового пошуку (beam search) для формування релевантного простору ознак, а також спроектувати сховище даних на основі документо-орієнтованої СУБД з підтримкою потокової обробки і паралельних обчислень.

3. Побудувати класифікатор, що відображає простір ознак у множину класів технічного стану, порівнявши моделі на основі необхідних метрик, виходячи з того, що саме пропуск дефекту, а не хибна тривога, становить критичний ризик безпеки експлуатації.

На основі розв'язків задач 1–3 спроектувати об'єктно-орієнтовану модель взаємодії ключових сутностей системи (генерація даних, препроцесор, аналітичне ядро, результат прогнозу) та відмовостійку мікросервісну

архітектуру (контейнеризація, оркестрація, балансування навантаження), що забезпечує безперервну роботу в режимі реального часу та подальшу інтеграцію в екосистему Smart City.

1.5. Висновки до першого розділу

1. Здійснено системний аналіз методів діагностики технічного стану транспортних засобів. Встановлено, що телеметричні дані, отримані через стандартизований інтерфейс OBD-II (зокрема, діагностичні коди несправностей DTC та показники датчиків у реальному часі), є базовим джерелом масивів інформації. Їх використання формує інформаційне підґрунтя для переходу автомобільного сервісу від застарілих реактивних (ремонт після відмови) та планових стратегій до проактивного прогностичного обслуговування.

2. Досліджено еволюцію систем управління базами даних у контексті потреб автомобільного сервісу. Виявлено, що традиційні реляційні бази даних та класичні багатовимірні моделі (OLAP), орієнтовані переважно на обліково-фінансову аналітику, вичерпують свою ефективність при обробці високошвидкісних та об'ємних потоків телеметрії. Обґрунтовано необхідність переходу до розподілених екосистем Big Data, які здатні консолідувати гетерогенні дані з різних сервісних центрів для швидкого знаходження закономірностей та розробки складних прогностичних моделей технічного стану ТЗ.

3. Визначено роль прогностичного моніторингу в контексті розвитку концепції Smart City. Доведено, що умови експлуатації в інтенсивному міському середовищі прямо корелюють зі зносом агрегатів автомобіля. Застосування методу агрегації даних рухомих автомобілів (Floating Car Data) та архітектури V2X дозволяє інтегрувати телеметрію в екосистему «розумного міста»,

враховувати вплив трафіку як додатковий предиктор зносу та оптимізувати загальне теплове і механічне навантаження на деталі ТЗ.

4. Сформовано вимоги до попередньої обробки великих даних автосервісу. Оскільки масиви телеметрії часто є неоднорідними, розрідженими та зашумленими, доведено, що обов'язковим перехідним етапом до застосування алгоритмів машинного навчання є експлораторний аналіз даних (EDA). Цей етап дозволяє виявити приховані закономірності, локалізувати аномалії та виокремити найбільш релевантні предиктори для забезпечення високої точності прогностичного моніторингу.

5. Моніторинг технічного стану автомобіля є ключовим елементом сучасних інформаційно-керуючих систем транспортних засобів. Систематичний збір та аналіз даних про роботу вузлів та агрегатів (двигун, трансмісія, ходова частина, системи безпеки та ін) дозволяє не тільки своєчасно виявляти відхилення від норми, а й формувати управлінські рішення на основі отриманої інформації. Інтеграція результатів моніторингу в ІКС забезпечує прогнозування несправностей та планування профілактичного обслуговування, оптимізацію експлуатаційних витрат за рахунок зниження аварійності та простою, підвищення безпеки руху рахунок оперативного реагування на критичні параметри, автоматизацію управління технічним станом.

РОЗДІЛ 2 ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ СТВОРЕННЯ СХОВИЩА ДАНИХ ДІАГНОСТИКИ ТРАНСПОРТНИХ ЗАСОБІВ

2.1. Процес збору даних

Доступ до більшості інформації, яка описує особливості стану ТЗ, можна отримати через CAN-шину транспортного засобу. CAN (Controller Area Network) – це внутрішня мережа, яка з'єднує всі датчики автомобіля і через яку вони взаємодіють між собою. Безпосередньо під'єднатися до неї дуже проблематично: тому зазвичай використовується інтерфейс OBD, що дає змогу отримувати дані з CAN-шини. Процес збору даних полягає в підключенні до мережі CAN портативного комп'ютера (або датчика OBD II) з метою отримання даних під час експериментів. [46].

Наразі існує п'ять комунікаційних протоколів, що використовуються в системах OBD II (рис. 2.1, 2.2), два з яких визначені Товариством автомобільних інженерів (SAE), а три - Міжнародною організацією зі стандартизації (ISO) – ISO-9141, ISO-14230 (також називається KWP2000), PWM, VPW і CAN (також у кожного з протоколів є кілька різновидів - наприклад, різновиди відрізняються за швидкістю передачі даних). Проте, треба враховувати, що одна й та сама модель з одним і тим самим двигуном, одного року виробництва може бути виготовлена для кількох країн з підтримкою різних протоколів діагностування (точно так само протоколи можуть різнитися і за зразками двигунів, роками виробництва) [47]:

- протокол ISO-9141 – використовується виробниками з Азії та Європи і використовує однопровідну 12-вольтову шину, визначається при наявності контакту 7 у діагностичному роз'ємі (K-line) і відсутністю 2 та/або 10 контактів у діагностичному роз'ємі. Використовувані виводи - 4, 5, 7, 15 (може не бути), 16;

- SAE J1850 зі змінною шириною імпульсу (VPW) - використовується GM і позначається як лінія передачі інформації класу 2; він використовує однопровідну 7-вольтову шину, яка з'єднується зі сканером через контакт 2. Використовувані виводи - 2, 4, 5, 16 (без 10).

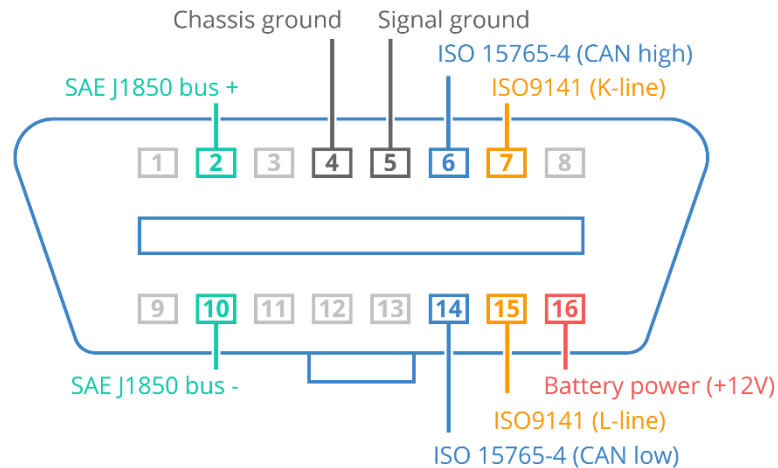


Рисунок 2.1 – Схема OBD II порта

PIN	Description	PIN	Description
1	Manufacturer Specific	9	Manufacturer Specific
2	J1850 Bus+	10	J1850 Bus
3	Manufacturer Specific	11	Manufacturer Specific
4	Chassis GND	12	Manufacturer Specific
5	Signal GND	13	Manufacturer Specific
6	CAN High	14	CAN Low
7	ISO-9141-2 K-LINE	15	ISO-9141-2 L-LINE
8	Manufacturer Specific	16	+12 V (Battery Power)

Рисунок 2.2 – Опис роз'єму OBD-II

- SAE J1850 PWM (Pulse Width Modulation - Широтно-імпульсна модуляція (ШІМ)) - Застосовується компанією Ford і має двопровідну 5-ти вольтову шину, яка з'єднується зі сканером через контакти 2 і 10. Використовувані виводи - 2, 4, 5, 10, 16.

- ISO 14230 Keyword (ISO 9141-2) - новіша версія стандарту ISO 9141, яка виглядає так само Це європейський протокол, який зв'язується зі сканером через вивід 7;

- ISO 15765 Controller Area Network (CAN) - був поступово запроваджений впродовж 2005-2007 модельних років. Він став уніфікованим протоколом зв'язку силової установки для всіх автомобілів 2008 року випуску та пізніших моделей. Автомобілі, обладнані CAN, з'єднуються зі сканером через контакти 6 і 14;

- Контакт 4 - заземлення шасі для блоку керування силовим агрегатом (PCM - Powertrain Control Module) і приладу для сканування.

- Контакт 5 - це заземлення модуля контролю силового агрегату (PCM - Powertrain Control Module);

- Контакт 16 - це запобіжник напруги акумуляторної батареї, який забезпечує електроживлення приладу для сканування.

Удосконалення технічного обслуговування та управління технічним станом автомобілів призводить до підвищення ефективності використання ТЗ, шляхом більш повного використання його індивідуальних можливостей у процесі експлуатації.

Сутність проблеми полягає в тому, що через високу варіацію ресурсів вузлів і механізмів автомобілів їхні індивідуальні властивості за планового контролю реалізуються не повністю. У результаті цього мають місце значні витрати матеріальних ресурсів через пропуск поломок, передчасну профілактику та недостатню індивідуальну інформацію про стан кожного автомобіля. Відповідно, при роботі з ТЗ більша частина ресурсів йде на поточний ремонт, що полягає, як правило, в усуненні відмов через несвоєчасне виявлення несправностей, так як основним джерелом інформації про стан автомобіля є технічне обслуговування (ТО). [48].

Одна з найбільш ефективних стратегій для підтримки ТЗ у справному стані – технічне обслуговування. Але її неможливо застосовувати без діагностичних робіт та засобів для проведення такої роботи, щоб дізнатись повну індивідуальну інформацію про ТЗ (цілком, окремо по вузлах чи системах). Ключовим в цій роботі є рішення задачі зменшення витрат та оптимізація діагностування теперішніх та майбутніх поломок. Для вирішення цього можна розробити певні напрями для роботи:

- підвищення ефективності зовнішнього стаціонарного діагностування шляхом вдосконалення його методів і засобів, у поєднанні з впровадженням автоматизованих систем управління ТО та ремонту;
- розроблення засобів вбудованого діагностування, що дають змогу здійснювати безперервний контроль за технічним станом автомобіля за мінімальних витрат.

Розвиток обох цих напрямів має бути на єдиній технологічній основі діагностування, що забезпечить максимальну ефективність цього. Відповідно, це дозволить краще контролювати технічний стан автомобіля та покращить його ефективні показники (наприклад, потужність, розхід палива тощо).

При аналізі стану ТЗ, більшість несправностей виявляється у вигляді зовнішніх факторів, більшість з яких має однаковий характер. Для того щоб правильно і швидко поставити діагноз під час перевірки комплексного об'єкта за рахунок використання окремих засобів діагностування, необхідно мати у розпорядженні велику кількість даних про функціональні зв'язки між можливими поломками та їхніми симптомами [49]. Якщо за якоюсь складовою частиною відомі лише комбінації симптомів і їхні зв'язки з відповідними несправностями, але невідомі ймовірності найчастішого виникнення хоча б деяких із них, характерних для цього симптому, то в цьому разі пошук конкретної несправності здійснюється, виходячи з умови, що при такому симптомі всі пов'язані з ним несправності є рівно ймовірними.

Пошук несправностей розподіляють на складові частини, для яких існує певне співвідношення між часом виявлення поломки та ймовірністю її виникнення. За умови однакової ймовірності виникнення двох або більше несправностей, характерних для будь-якого симптому, пошук здійснюють, виходячи з мінімального часу, витраченого на перевірку. Якщо відношення однакове для пошуку несправностей із однаковими зовнішніми ознаками, то в цьому разі пошук за методом «час-ймовірність» неефективний, оскільки він спричиняє невизначеність, тобто випадковий вибір послідовності пошуку несправності, що виникла [50].

Однією з ключових несправностей, яка може статися в сучасному ТЗ, – відмова у вузлах електронних блоків керування (ЕБК). Це відбувається через те, що наразі новітні ТЗ мають дедалі складніші комп'ютеризовані системи керування, які потребують дедалі складніших методів діагностування, контролю та тестування. Для роботи з електронною складовою ТЗ наразі є один із універсальних засобів діагностики – OBD II сканер із підключеним діагностичним програмним забезпеченням (рис.2.3) [51].

Технічно можливо поєднати бортовий комп'ютер та діагностичний сканер у один пристрій, яке займе місце в ТЗ в місці розміщення бортового комп'ютера. Схема роботи діагностичної системи (рис. 2.4) при такому вигляді цільної системи буде наступним:

- завантажуючись в систему потрібно буде обрати режим роботи з ебк;
- відбувається перевірка зв'язку з модулем ебк;
- відбувається запит параметрів системи двигуна (перевірка потужності двигуна, якщо менше 85% - сигналізування про необхідність то);
- сканування двигуна ебк на наявність помилок (відповідно, якщо вони є – розшифровка та показ рекомендацій для усунення);
- якщо помилку не виявлено, то система переходить в режим опитування для виявлення несправностей по характерним ознакам.

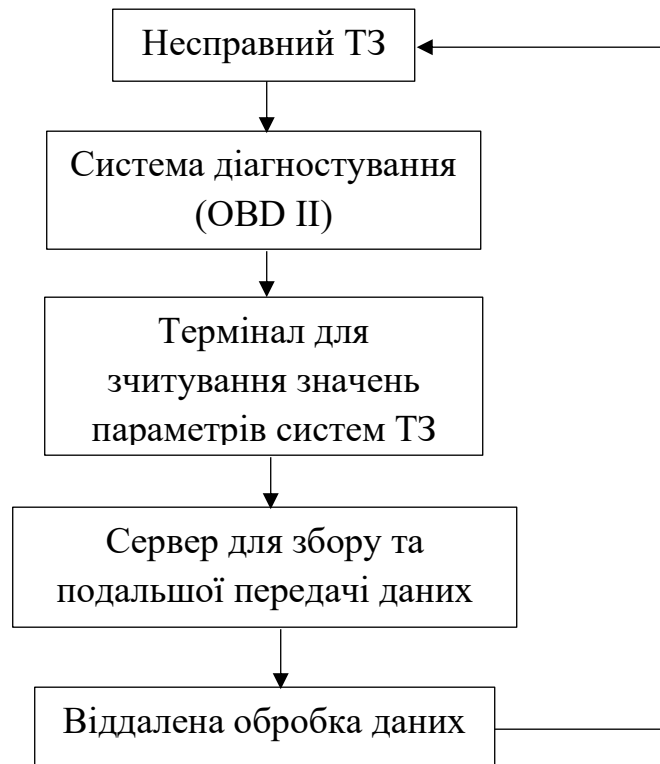


Рисунок 2.3 – Взаємодія сервісу віддаленого контролю параметрів технічного стану [53]

- при завершенні діагностичної перевірки можна знов обрати режим роботи:
 1. якщо помилки не виявлені, але є підозри про несправність, то відбувається опитування системи, в ході якого виявлені коди помилок автоматично розшифровуються
 2. якщо підозра на несправність не вирішена і не виявлена іншими режимами, або помилка не пов'язана з електронним контрольним модулем (екм), то рекомендується автономний режим опитування, який в ручному режимі обирає якісну ознаку несправності елементу тз.
 3. штатний режим контролює параметри тз, відображає їх теперішній стан та записує в систему необхідні дані для кожного циклу запуску двигуна.
- за необхідності інформація на індикаторі оновлюється з конвертацією даних, отриманих з ЕБУ. Інформація для користувача виводиться в зручній формі, тобто у вигляді розгорнутих буквено-цифрових повідомлень і підказок. Обсяг інформації для відображення в розгорнутому вигляді є дуже великим, що

тягне за собою збільшення потреби в пам'яті для її зберігання. Отримані з ЕБУ дані в деяких випадках мають бути перераховані (точність обчислень при цьому може бути невеликою) і перетворені з двійкової форми на символний формат.

- закінчення роботи та повторення даного циклу заново [52].

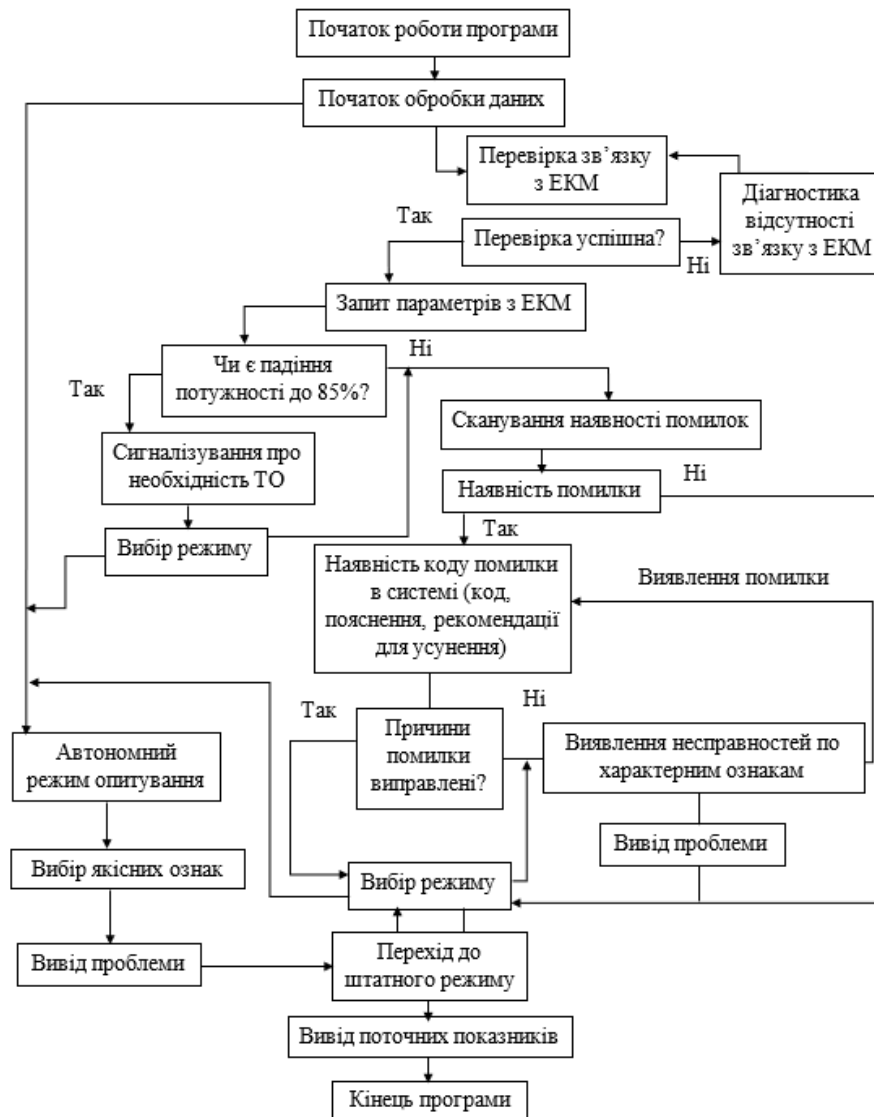


Рисунок 2.4 – Схема роботи вбудованої системи діагностування двигуна

2.2. Режими роботи діагностичного обладнання

Так як для основного пошуку несправностей використовується OBD-II сканер, який дає певний ряд стандартизованих функціональних можливостей для діагностики [53]:

Режим 1 - Зчитування актуальних параметрів роботи системи керування (Mode 1 PID Status & Live PID Information). Загалом стандартом підтримується близько 20 параметрів. Однак, кожен конкретний блок керування підтримує обмежену кількість із них (наприклад, залежно від встановлених датчиків кисню). З іншого боку, деякі автовиробники підтримують розширені набори параметрів - наприклад, деякі автомобілі концерну GM підтримують понад 100 параметрів. Через систему OBD-II діагностики можна зчитати такі основні параметри:

- режим роботи системи паливної корекції (PID 03 Fuel system status). При значенні «Closed Loop» система працює в режимі зворотного зв'язку (замкнутої петлі), при цьому дані з датчика кисню використовуються для коригування подачі палива. При значенні «Open Loop» дані з датчика кисню не використовуються для коригування подачі палива;
- тиск палива (PID 0A Fuel pressure);
- тиск у впускному колекторі (PID 0B Manifold pressure);
- оберти двигуна (PID 0C Engine speed - RPM);
- швидкість автомобіля (PID 0D Vehicle speed);
- кут випередження запалювання (PID 0E Ignition Timing Advance);
- температура всмоктуваного повітря (PID 0F Intake Air Temperature);
- витрата повітря (PID 10 Air Flow);
- положення дросельної заслінки (PID 11 Throttle position);
- розрахункове навантаження на двигун (PID 04 Calculated Load);
- температура охолоджувальної рідини (PID 05 Coolant temperature);

Зазвичай для аналізу роботи конкретної підсистеми системи керування двигуном достатньо одночасно контролювати 2-3 параметри. Однак, іноді потрібно одночасно переглядати і більшу кількість. Кількість одночасно контрольованих параметрів, а також формат їхнього виведення (текстовий та/або графічний) залежать як від можливостей конкретної програми сканера, так і від швидкості обміну інформацією з блоком керування двигуном автомобіля

Режим 2 - Отримання збереженого знімка поточних параметрів роботи системи керування на момент виникнення кодів несправностей (Mode 2 Freeze Frame).

Режим 3 - Зчитування та перегляд кодів несправностей (Mode 3 Read Diagnostic Trouble Codes (DTCs)).

Режим 4 - Очищення діагностичної пам'яті (Mode 4 Reset DTC's and Freeze Frame data) - стирання кодів несправностей, світлин поточних параметрів, результатів тестів датчиків кисню, результатів тестових моніторів.

Режим 5 - Зчитування і перегляд результатів тесту датчиків кисню (Mode 5 O2 Sensor Monitoring Test Result).

Режим 6 - Запит останніх результатів діагностики одноразових тестових моніторів (тестів, що проводяться один раз протягом поїздки) (Mode 6 Test results, non-continuosly monitored) - ці тести контролюють роботу каталізатора, системи рециркуляції вихлопних газів (EGR), системи вентиляції паливного бака.

Режим 7 - Запит результатів діагностики постійно діючих тестових моніторів (тестів, які виконуються постійно, доки виконуються умови для проведення тесту) (Mode 7 Test results, continuosly monitored) - ці тести контролюють склад паливо-повітряної суміші, пропуски запалювання (misfire), решту компонентів, що впливають на вихлоп.

Режим 8 - Керування виконавчими механізмами.

Режим 9 - Запит інформації про автомобіль, що діагностується (Mode 9 Request vehicle information) - VIN-коду і даних, що потребують калібрування. [48-50]

Одним із найбільш важливих об'єктів діагностики є двигун автомобіля. У процесі експлуатації в двигуні виникають різні несправності (рис. 2.5), пов'язані зі структурною зміною пов'язаних деталей: це зношування і задирки циліндрів, спідниць поршнів, поршневих пальців, вкладишів шатунних і корінних підшипників, а також збільшення зазорів між клапанами та штовхачами, засмічення паливопроводів, фільтрів тощо. Згідно статистичних даних, із загальної кількості відмов більша частина припадає на циліндро-поршкову групу (ЦПГ) і кривошипно-шатунний механізм (КШМ), на ремонт цих же елементів припадає до 70 % від загальних витрат на ремонт двигуна. Відповідно, вчасна діагностика вирішує низку завдань, одне з яких - це визначення гранично допустимих зазорів у сполучених деталях, що у свою чергу може завчасно попередити власника ТЗ про необхідність ремонту [54-56].

Автори [57] одним з варіантів для діагностування двигуна було побудовано алгоритм автоматизованої системи ітеративного контролю для розпізнання поточного стану ДВЗ і його окремих елементів, таких як циліндро-поршнева група (ЦПГ) та газорозподільного механізму (ГРМ); забезпечення достатньої інформації для формування необхідних операцій технічного сервісу ДВЗ; можливість прогнозування залишкового ресурсу роботи двигуна (рис.2.6).

Існують певні методи діагностики стану двигуна. ЦПГ, наприклад, діагностують за функціональними параметрами: компресія в циліндрах, чад мастила, прорив газів у картер, величина струму, який споживає стартер, перевищення рівня вібрації та шуму, а також зміна параметрів моторного мастила. Визначити технічний стан КШМ можна за тиском моторного мастила в головній масляній магістралі (мінімальний показник на холостому ході і

прогрітому двигуні дорівнює 0,1 МПа), а також за збільшенням витрати мастила за певний часовий проміжок [58].

Затруднення у запуску холодного двигуна										Ознаки несправності двигуна та їх причини
Затруднення у запуску гарячого двигуна										
Не стійкий холостий хід										
Збої в роботі двигуна під навантаженням										
Падіння потужності двигуна										
Підвищена витрата пального										
Підвищена димність, чорні ВГ										
Підвищена димність, сірі ВГ										
"Жорстка" робота двигуна										
Недостатньо обертів										
Забагато обертів										
										Підсос повітря в паливну систему
										Несправний електромагнітний клапан
										Мала пускова подача, несправність ПНВТ
										Несправність ПНВТ
										Забруднені паливні патрубки, загусте паливо
										Забруднений паливний фільтр
										Забруднений повітряний фільтр
										Забрудненні зворотні патрубки
										Ранній вприк палива
										Пізнйй вприк палива
										Порушення регулювання подачі
										Несправна(і) форсунка (форсунки)
										Несправна система передпускового підігріву
										Порушення щілин в приводі клапанів
										Низька компресія
										Пошкодження циліндру
										Несправність турбокомпресору
										Забруднення нейтралізатору вихідних газів (ВГ)

Рисунок 2.5 – Ознаки несправностей двигуна

За ознаками переліченими вище, діагностування стану ДВЗ дають змогу з різним ступенем імовірності визначати значення параметрів двигуна, але не дають змоги точно визначити момент початку аварійних явищ, що виникають під час роботи сполучених деталей. Наприклад, за перевищенням рівня вібрації можливо досить точно встановити величину зазору між вкладишем і шийкою

колінчастого валу, але встановити конкретну величину зазору, після якої починається збільшене зношення деталі практично неможливо. З цього випливає, що велика ймовірність проведення ремонту або передчасно, або вже досить пізно.

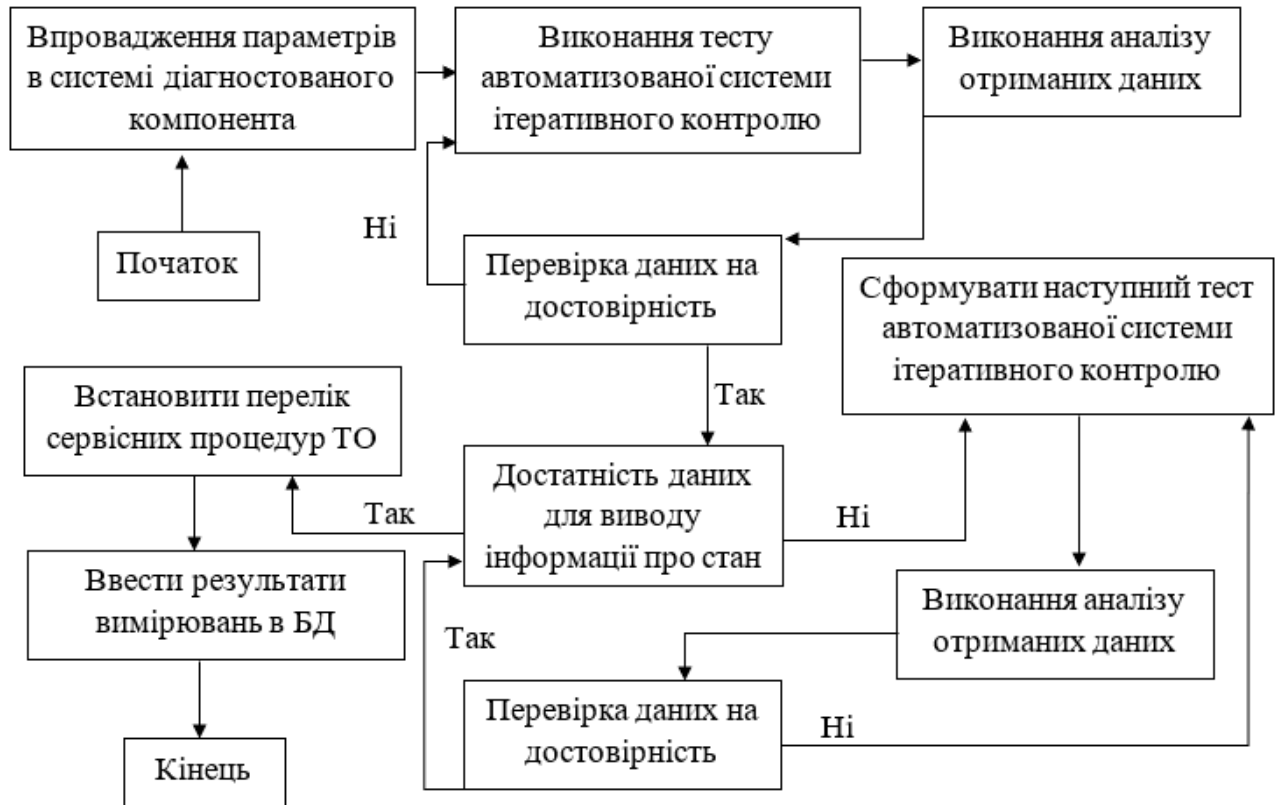


Рисунок 2.6 – Схема діагностики двигуна з АСІК (автоматизованою системою ітеративного контролю), на основі [57]

2.3. Методи пошуку несправностей

Так як при роботі ТЗ більшість несправностей проявляються у вигляді зовнішніх факторів, то різні поломки можуть мати однаковий характер. Несправності, що найбільш часто трапляються, а також їх зовнішні прояви, дозволяють виявити несправність, що виникла, не проводячи зайвих перевірок [59]. Також іноді вдаються до методів послідовного виключення. Щоб правильно і швидко зробити висновки під час перевірки складного об'єкта за

рахунок використання окремих засобів діагностування, необхідно мати у своєму розпорядженні велику кількість даних про функціональні зв'язки між можливими несправностями та їхніми симптомами (рис.2.7).

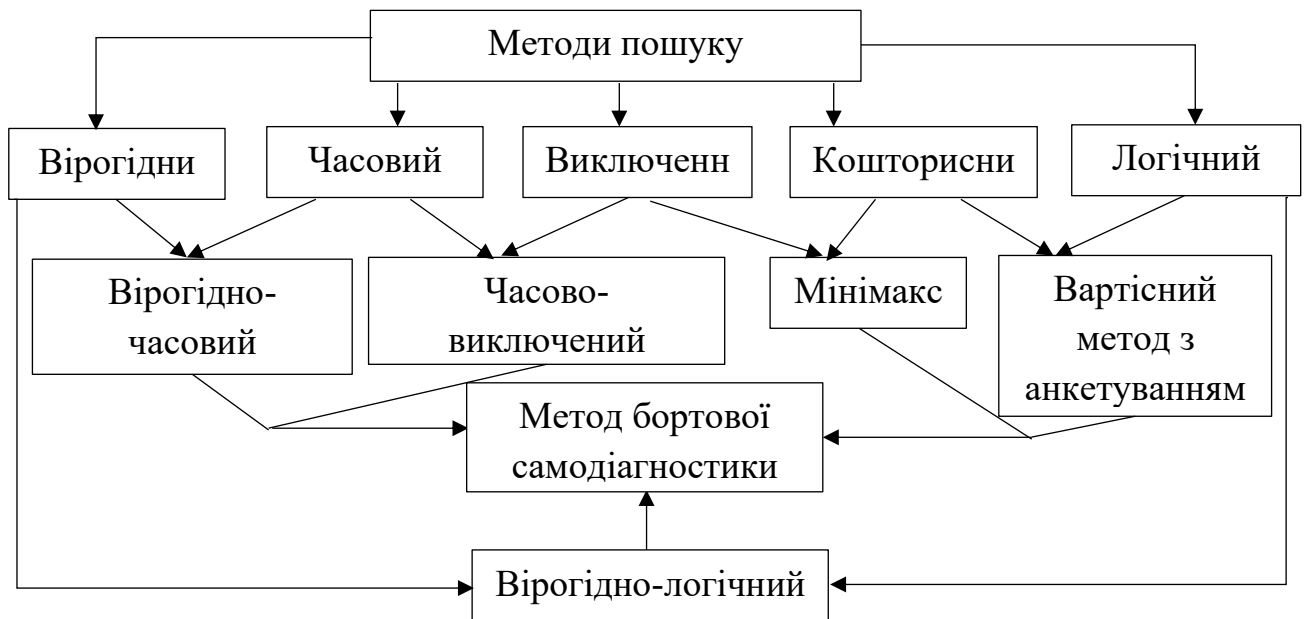


Рисунок 2.7 – Схема вибору метода бортової самодіагностики

Якщо з якої-небудь складової частини відомі лише комбінації симптомів та їхні зв'язки з відповідними несправностями, але невідомі ймовірності найчастішого виникнення хоча б деяких із них, характерних для цього симптому, то в цьому разі пошук конкретної несправності ведуть, виходячи з припущення, що при цьому симптомі всі пов'язані з ним несправності є рівноймовірними.

Для виявлення причин таких несправностей має бути розроблено систему вимірювальних перетворювачів, які фіксували б як несправності, які трапляються рідко, так і ті, які трапляються часто. Теоретично, такий метод визначення несправностей можна здійснити, але практично надзвичайно складний і дорогий. Застосування положень теорії ймовірності, зокрема теорії інформації, дає змогу значно спростити процес постановки діагнозу [60].

Сутність вірогідного підходу до визначення характеру несправності полягає в наступному: на основі статистичних даних про закономірності зміни параметрів стану залежно від напрацювання складової частини або машини загалом, про можливі комбінації симптомів та їхні зв'язки з несправностями для кожної несправності встановлюють ймовірність її виникнення та ймовірність появи кожного симптому. За даними матеріалами створюється програма пошуку цієї несправності, яка перебирає їх у порядку зменшення ймовірності виникнення різних відмов, характерних для цього симптому.

З метою ще більшої мінімізації витрат часу та коштів на пошуки несправності під час розробки такої програми-пошуку слід брати до уваги не тільки ймовірність виникнення несправності, а й час, що витрачається на виявлення кожної з них під час діагностування. Це приводить нас до вірогідно-часового методу. У цьому випадку послідовність перевірки встановлюють, виходячи з відношення часу t , необхідного на виявлення несправності, до ймовірності p появи цієї несправності. Пошук несправності починають зі складових частин, для яких зазначене відношення виходить мінімальним.

За однакової ймовірності виникнення двох або більше несправностей, характерних для будь-якого симптому, пошук буде здійснюватись, виходячи з мінімального часу, що витрачається на перевірку [61]. Якщо відношення однакове для пошуку несправностей з однаковими зовнішніми ознаками, то в цьому разі пошук за вірогідно-часовим методом неефективний, тому що він призводить до невизначеності, тобто до випадкового вибору послідовності пошуку несправності, що виникла. Один з важливих критеріїв при виборі оптимальної послідовності пошуку несправностей – мінімальна величина середньої вартості перевірки. Під час використання цього критерію потрібно досягти того, щоб максимальна вартість пошуку елемента, що відмовив, була найменшою порівняно з витратами, які отримують за інших методів перевірки. Такий метод пошуку отримав назву методу мінімаксу, який найбільш

ефективний у тих випадках, коли простої автомобілів на терміни виконання робіт не впливають [63].

Метод логічного пошуку з почерговим виключенням не потребує застосування додаткового діагностичного обладнання, має невисоку трудомісткість, не вимагає від того, хто проводить діагностування, високої кваліфікації та спеціальних знань, але має високу залежність від людського фактору, тобто діагностування здійснюється на підставі даних, отриманих від водія про непрямі ознаки виникнення несправності, події, що передували появі дефекту (проходженні ТО, переліку операцій ТО, використаних матеріалах під час ремонту та експлуатації, режимі роботи машини тощо), і подальшому аналізі.

Метод виключення - базується на послідовному виключенні робочих ланок механізму, що дає змогу, не проводячи демонтаж окремих систем та попередню перевірку, виявити несправність [63].

Вірогідність-виняток будується на основі статистичних даних про закономірності зміни параметрів стану залежно від напрацювання складової частини або машини загалом встановлюють не тільки ймовірність відмови, а й порядок перевірки методом винятку. За отриманими даними будується програма пошуку цієї несправності, яка побудована відносно порядку спадання ймовірності виникнення різних відмов, характерних для даного симптому.

З метою зменшення впливу людського фактора, можна запропонувати вірогідно-логічний метод пошуку несправностей. Для реалізації такого методу необхідно встановити на автомобіль систему самодіагностики для елементів які найчастіше виходять з ладу, що в свою чергу дасть більше інформації в онлайн режимі про стан ТЗ.

Вірогідно-логічний – такий метод дає можливість «спостерігати» інформацію про стан більшості елементів системи одночасно - «онлайн» у вигляді побудови графіка не вдаючись до перебору перевірок діагностичних

параметрів елементів окремо [64]. В свою чергу, даний метод можна реалізувати з встановленням систему самодіагностики для елементів, що найчастіше виходять з ладу елементів. При використанні даного методу на основі вбудованих засобів (інформаційні, сигналізаційні, програмовані, запам'ятовуючі) можна мінімізувати ймовірність виникнення несправності шляхом своєчасного відстеження зміни контрольного параметра.

Враховуючи це, надважлива проблема в сфері технічної діагностики автомобілів - встановлення симптомів в залежності від напрацювання складових частин або ТЗ загалом, а також виявлення залежностей між цими симптомами і відповідними їм параметрами технічного стану техніки. Однак, якщо є несправності та відмови, спочатку встановлюють можливі причини їх виникнення за характерними ознаками, а потім, виходячи з передбачуваної причини виникнення несправності, підбирають відповідні діагностичні засоби, за рахунок використання яких дають висновок про характер і сутність несправності [65].

Одним з можливих варіантів прогнозування кількості відмов є аналіз імітаційного моделювання (рис.2.8). Спочатку відбувається моделювання пробігу та оцінка необхідності проведення сервісного обслуговування. У разі проведення сервісного обслуговування здійснюється повернення в початковий стан з оновленням показників, у протилежному випадку відбувається перехід у стан моделювання пробігу до відмови і розрахунок показників, пов'язаних із ремонтом ТЗ. Після чого алгоритм перезапускається. Водночас, можна допустити використання змінних, що виробляють економічну оцінку як витрат на ремонт так і запасні частини. Такий підхід дає змогу спрогнозувати кількість як сервісних обслуговувань ТЗ, так і потребу в ремонті та в запасних частинах [66].

На даний момент OBD-II інтерфейси є широко поширеним вирішенням для збору інформації з датчиків транспортних засобів. Однак, є деякі недоліки в

застосуванні OBD-II донглів. Велика частина існуючих методів збору даних ґрунтується на моделі «запит-відповідь» для забезпечення доступу до інформації [67, 68]. Невелика швидкість зчитування при традиційному методі «запит-відповідь» (рис. 2.9 та рис. 2.10) впливає на точність результатів, призводячи до отримання неточних значень, що, в свою чергу, знижує ефективність роботи численних залежних програм верхнього рівня (наприклад, програм, що включають інформацію про поведінку водія, автоматичне виявлення аварій, допомогу при керуванні автомобілем в режимі реального часу, виявлення слизької дороги, безпеку) [69]. В цілому, для більшості випадків використання моделі «запит-відповідь» максимальна (тобто найшвидша) допустима швидкість зчитування становить лише 9 запитів на секунду, що дає 8 точок даних на секунду в [70]. Логічним варіантом рішення цієї проблеми буде доступ до даних у реальному часі, що гарантує точність і достовірність вхідних даних за рахунок зменшення часового лагу між кожною парою послідовних вимірювань (тобто, роздільної здатності запиту).



Рисунок 2.8 – Схема імітаційного моделювання відмови у ТЗ

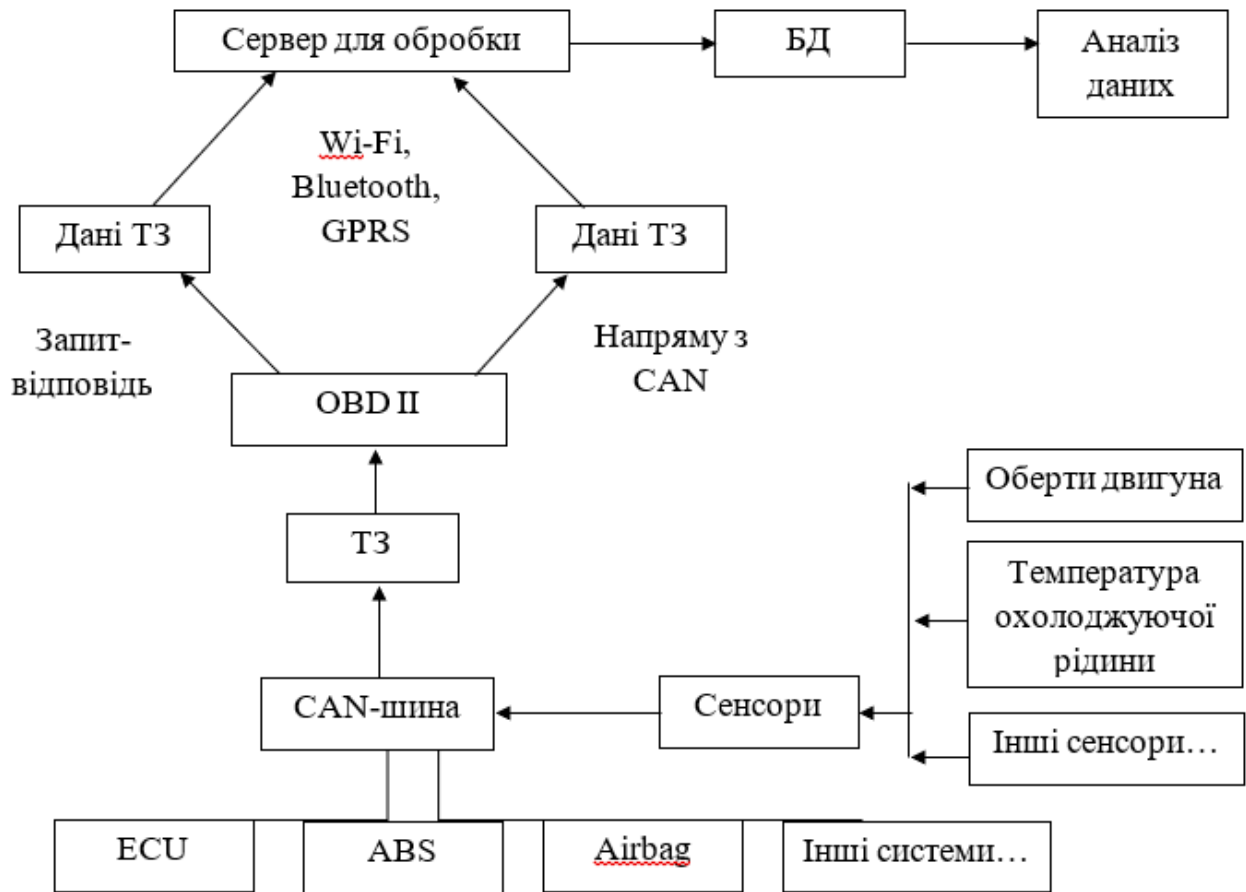


Рисунок 2.9 – Отримання інформації з сенсорів та датчиків про ТЗ

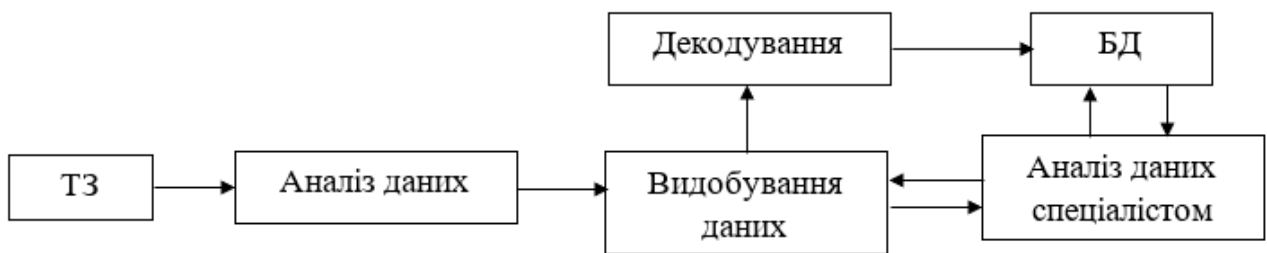


Рисунок 2.10 – Базова архітектура отримання даних

У зв'язку з тим, що останнім часом багато виробників обирають CAN в якості основної шини у внутрішній системі автомобіля, CAN може бути встановлений як основний комунікаційний протокол між автомобілем і зовнішнім OBD-інтерфейсом. Інтеграція протоколів зв'язку шини CAN передбачає ідентифікацію протоколу зв'язку, який використовується у

відповідній моделі автомобіля. Однак, при визначенні комунікаційних властивостей шини CAN для визначення режиму зв'язку, помилкові пакети можуть спричинити перехід шини CAN у вимкнений стан, що призведе до втрати зв'язку з автомобілем і, як результат, до аварійного завершення роботи системи [71, 72].

В роботах [67, 69, 70] система моніторингу транспортного засобу використовує ELM327 як зчитувач OBD-II для отримання потрібних даних (наприклад, частота обертання двигуна, швидкість транспортного засобу, температура охолоджуючої рідини та прискорення педаль) та їх передачі на смартфон користувача через Bluetooth. В роботі [68] та ж сама платформа використовує окремий додаток для отримання даних для подальшого аналізу поведінки водія на основі даних ТЗ.

Відповідно, для покращення такої системи необхідне проектування і створення чутливої до затримок і здатної до реконфігурації системи збору даних для автомобільних телеметричних систем, яка може бути використана як частина додатків інтелектуальних транспортних систем (ІТС) для підвищення ефективності, безпеки і відмовостійкості системи моніторингу транспортних засобів. Інформація, що надходить з необроблених даних шини CAN, забезпечить високі вимоги до точності для багатьох автомобільних додатків, що працюють в режимі реального часу [72, 73].

Основою для реєстрації даних ТЗ є три основних модуля які здатні обмінюватися даними з електронним блоком керування транспортним засобом, виявляти дефекти та записувати дані з OBD-II сканера підключеного до транспортного засобу (рис. 2.11).

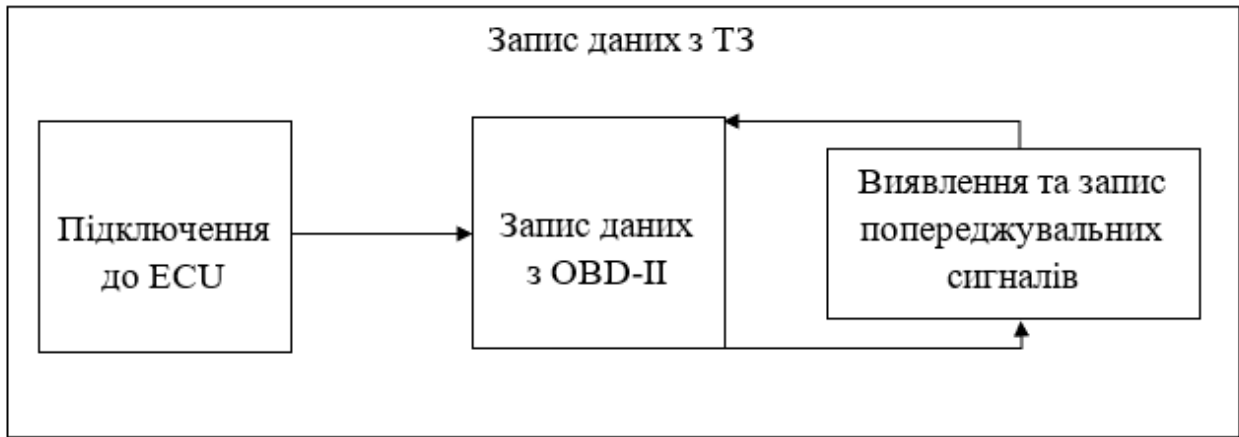


Рисунок 2.11 – Блок-схема реєстратора даних ТЗ

Як показано на рисунку 2.12, процес дослідження починається з пошуку сканера OBD-II за рахунок використання каналів Wi-Fi або Bluetooth. Для того, щоб зв'язатися з ECU, система автоматично знайде протокол OBD-II, який застосовується на транспортному засобі. Якщо автомобіль не використовує всі підтримувані стандартом OBD-II протоколи, то система буде видавати повідомлення про несумісність протоколів. Якщо підтримується, система буде чекати, поки двигун увімкнеться з індикатором обертів двигуна ($\text{Оберти} > 0$). Вже під'єднаний до мережі діагностичний сканер буде відправляти виявлені діагностичні дані (коди) на сервер з підключеною БД [74].

Одні з ключових підсистем, які мають бути в СЦ - зчитувач кодів OBD-II, та підсистема основної логіки (рис. 2.4). Функції зчитувача кодів OBD-II полягають у зчитуванні коду, поступає зі сканера OBD II, визначення причини несправностей та класифікації цих несправностей за певними категоріями. Структура формату діагностичного коду складається з п'яти символів алфавітно-цифрового коду, в якому кожному символу відповідає конкретне значення. Конкретне значення контролюється загальним стандартом (ISO/SAE) або ж виробником оригінального обладнання (OEM). Код має формат повідомлення у два байти, де перші два старші біти (MSB) вказують на перший символ, наступні два біти призначені для позначення другого символу, третій

біт – для позначення третього символу, а останній байт відведено для позначення конкретної причини несправності.

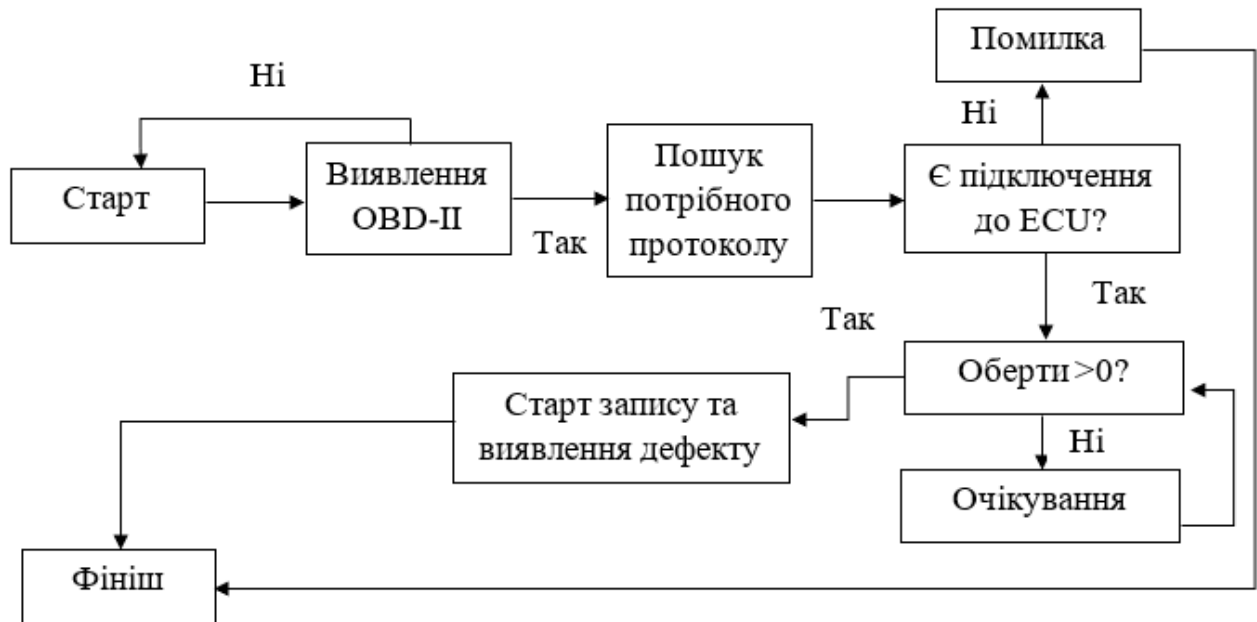


Рисунок 2.12 – Підключення ЕБК до блоку реєстрації даних

Діагностичний код має лише кілька ознак, серед яких найважливішим є ідентифікатор зчитування. Зчитування – це «віртуальна фотографія» стану транспортного засобу, яка записується в БД кожного разу, коли автомобіль підключається до діагностичного програмного забезпечення на СТО. Існує чотири системних коди, необхідних для передачі даних по OBD-II. Це коди силового агрегату (Р-коди), коди кузова (В-коди), коди шасі (С-коди) та мережеві коди (U-коди) [67, 70]. Однак одна несправність часом може викликати більше одного коду. Наприклад, втрата сигналу датчика частоти обертання коліс може спричинити як Р-код, так і В-код, якщо датчик використовується для входу в систему VSS (Vehicle speed sensor) та систему ABS. Крім того, будь-яка комунікаційна помилка, що перешкоджає РСМ (Pulse Code Modulation - імпульсно-кодова модуляція) здійснювати обмін інформацією з іншими модулями, може викликати як Р-, так і U-коди.

За всі операції в системі відповідає логічна підсистема, яка в свою чергу може бути на базі веб-серверів IIS / Apache та має всі веб-сервіси, необхідні для обслуговування запитів які отримує з бази даних. Так як бортові дані містять тисячі сигналів від датчиків і блоків управління, які передаються через мережу CAN, вони надсилаються багаторазово з визначеною періодичністю і утворюють безперервні інформаційні масиви, які використовуються для сигналізації стану різних вузлів транспортного засобу і передають в дану БД [75].

Відповідно, база даних технічного обслуговування транспортних засобів ((Vehicle Maintenance database (VSR) повинна містити відомості про всі види технічних обслуговувань, що проводяться на станціях технічного обслуговування з записами всієї інформації про вузли, датчики та сенсори ТЗ. Кожен запис включає інформацію про дату, пробіг, деталі, операції та коментарі працівників майстерні [76]. Така база даних буде досить неоднорідною, тому що рік випуску та специфікація автомобіля впливає на набір параметрів, що заносяться до бази даних. Лише невелика частина показників є спільною для всіх транспортних засобів. Загалом, новіші та досконаліші транспортні засоби надають більше статистичної інформації. Це ускладнює пошук великих наборів даних з однорідним набором параметрів.

2.4. Logged Vehicle Data (LVD) та зв'язки з функціональними блоками

База даних LVD зберігає зведену інформацію про особливості використання транспортних засобів. Дані завантажуються кожного разу, коли автомобіль заїжджає на авторизовану станцію техобслуговування та ремонту. Це буває кілька разів на рік, але з нерегулярними інтервалами, які важко передбачити апріорі. Під час експлуатації автомобіль безперервно збирає та зберігає низку показників, таких як середня швидкість або загальна витрата

пального. Загалом, це прості статистичні дані різного роду, оскільки існують дуже жорсткі обмеження на пам'ять і обчислювальну потужність, особливо для старих моделей автомобілів. Більшість параметрів належать до однієї з наступних трьох категорій:

- продуктивність;
- експлуатація транспортного засобу;
- діагностика або відлагоджування [77].

Для побудови такої бази даних, система повинна надавати попередження про компоненти з підвищеним ризиком виходу з ладу до наступного очікуваного візиту на станцію техобслуговування. Однак, якщо найближче зчитування перед відмовою становить 3-4 місяці, то менш імовірно, що знос мав видимий вплив на дані. Частота зчитування сильно варіюється між транспортними засобами і змінюється з віком транспортного засобу, і може становити лише одне зчитування на рік.

Збирання та обробка даних є найважливішим процесом у системах прогнозування. Дані з датчиків збирають і обробляють за рахунок використання OBU, а потім передають до VDS за рахунок використання бездротового модуля для подальшого зберігання у спеціально відведеному місці. Процес діагностування починається, коли транспортний засіб з невідомою несправністю підключається до VDS з метою ідентифікації дефекту. Потім відбувається порівняння фізичних даних, отриманих від датчиків, із відповідними ознаками несправності. Увага приділяється лише тим сеансам, під час яких несправність була успішно діагностованою. Тобто, не розглядається кожна сесія, несправність якої не була успішно визначена. Дані з баз даних LVD та бази технічного обслуговування об'єднані в один набір даних. Дані з LVD доповнюються змінними напрацювання на відмову на основі дат проведення ремонтів у VSR. Дані організовані у велику матрицю. Ці дані включають інформацію про напрацювання на відмову транспортних засобів, які ще не

вийшли з ладу. Показники з напрацюванням нижче заданого граничного значення, що називається горизонтом прогнозування (ГП) (Prediction horizon (PH)), отримують мітку «Несправність», тоді як показники з більшим напрацюванням у минулому отримують мітку «Нормальний стан». Це позначення застосовується для розподілу прикладів на дві групи: ті, що неминуче виходять з ладу, і ті, що справно функціонують. Але база даних LVD містить понад 25% пропущених результатів, що занадто багато для інтерполяції, і це спонукає до використання методу на основі фільтрів замість обгортки та інтегрованих методів. В дослідженні [75] пропонується два метода для заповнення даних параметрів – метод на основі фільтра і метод на основі обгортки. Метод на основі фільтрів виявляє найбільш вагомні параметри, аналізуючи їх індивідуально. Цей метод розбиває дані одного параметра на дві групи на основі міток. Функції щільності розподілу ймовірностей груп порівнюють за рахунок використання критерію Колмогорова-Смірнова [76]. Обчислюється вірогідність того, що дві групи належать до одного і того ж розподілу, і виражається за рахунок використання р-значення. Параметри, для яких обидві групи є чітко унікальними (р-значення близьке до нуля), пов'язані зі зносом перед відмовою і включені до набору даних:

$$P(\lambda) = 1 - \sum_{k=-\infty}^{\infty} (-1)^k e^{-2k^2 \lambda^2}, \quad (2.1)$$

$$\lambda = D_n \sqrt{n}, \quad (2.2)$$

$$D_n = \max |F_n(x) - F(x)|, \quad (2.3)$$

де $D_n \sqrt{n} \geq \lambda$ при обмеженні збільшення числа незалежних спостережень n

Метод обгортки базується на способі підбору транспортних засобів таким чином, щоб забезпечити послідовність набору даних і уникнути додавання

пропущених значень. У даних, що розглядаються, набори ознак відрізняються для різних транспортних засобів. Чистий перелік характеристик - це сукупність доступних показників для всіх транспортних засобів, що входять до складу набору даних. При внесенні нової характеристики до набору даних усі транспортні засоби, що не мають такої характеристики, вилючаються з набору. Додавання окремих параметрів призводить до того, що набір даних зменшиться до дуже малої частки зразків. Це робить необхідним брати до уваги факт зменшення розміру набору даних при підборі елементів. Запропонований метод дозволяє використовувати пучковий пошук для знаходження нових ознак для включення, розширюючи лише ті вузли, розмір набору даних яких не перевищує заданий поріг:

$$n_{dataset} = n_{all} * constraintFactor^{n_{params}}, \quad (2.4)$$

де $n_{dataset}$ — позначає необхідний мінімальний розмір набору даних,

n_{all} — кількість доступних зчитувань, $constraintFactor$ — коефіцієнт обмеження перебуває в діапазоні від 0 до 1,

n_{params} — кількість параметрів, які входять до обраного масиву даних. Кожен новий параметр може зменшити набір на невеликий відсоток. Це гарантує меншу граничну межу розміру набору даних.

У керованому машинному навчанні (рис. 2.13) індукційний алгоритм представляється набором навчальних даних, де кожен приклад описується за рахунок використання вектора властивостей (або атрибутів) і мітки класу [78]. Завдання алгоритму індукції, або індуктора, полягає в тому, щоб отримати такий кластер, який буде корисним для класифікації наступних випадків. Алгоритм індукції запускається на наборі даних, розділеному на внутрішню навчальну та тренувальну вибірки, при цьому з даних видаляються різні набори ознак. Набір даних з найвищою оцінкою обирається як остаточний

набір, на якому запускається індукційний алгоритм. Потім створений класифікатор оцінюється на незалежному тестовому наборі, який не використовувався під час навчання [79].

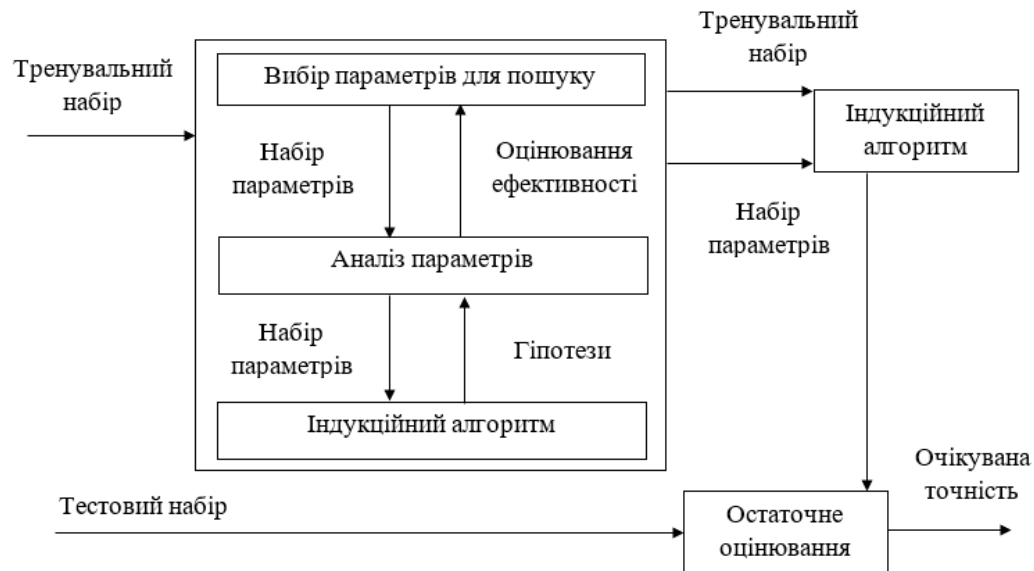


Рисунок 2.13 – Обгортковий метод для виділення елементів підмножини параметрів

Однак, чим новіший транспортний засіб, тим більше параметрів LVD доступні, але це також залежить від конфігурації транспортного засобу. Наприклад, детальні параметри коробки передач доступні лише для автомобілів з автоматичною коробкою передач. Це ускладнює отримання послідовних наборів даних для великого потоку транспортних засобів і ускладнює аналіз. Потрібно або вибрати набір даних з невідповідностями і заповнити пропущені значення, або обмежити аналіз лише тими транспортними засобами, які мають необхідні параметри. Такі бази даних містять важливу та цікаву інформацію про несправності транспортних засобів, яка іноді використовується технічним персоналом СТО для діагностики та прогнозування майбутніх несправностей.

Виходячи з цього, маємо два варіанти можливих отриманих даних:

- неповна вибірка: ці методи спрямовані на збалансування набору даних

шляхом усунення зразків класу, що становить більшість;

- надмірна вибірка: ці методи передбачають відтворення прикладів класу меншини для досягнення більш збалансованого розподілу [80].

Як неповна, так і надмірна вибірка мають певні недоліки. Неповна вибірка може відкинути потенційно корисні дані, а надмірна вибірка може збільшити ймовірність виникнення надмірної пристосованості, оскільки більшість методів надмірної генерації роблять точні копії зразків класу меншості.

В дослідженні [81] метод надмірної вибірки полягає у створенні нового прикладу класу меншості шляхом інтерполяції між кількома прикладами класів меншості, що лежать поруч. Таким чином, вдається уникнути проблеми надмірної відповідності, яка призводить до того, що границі рішення для класу меншості поширюються далі в простір класу більшості.

Часткове вирішення цієї проблеми полягає в тому, щоб забезпечити розподіл тестових і навчальних даних на основі кожного транспортного засобу, а не випадковим чином серед усіх зчитувань. Це означає, що якщо один або декілька показників з певного автомобіля належать до тестового набору, то жодні показники з цього ж автомобіля не можуть бути використані для тренування класифікатора. Набори даних для навчання і тестування повинні містити унікальні набори транспортних засобів, що не перетинаються, щоб гарантувати, що вивчаються закономірності, пов'язані зі зносом і експлуатацією, а не специфічні закономірності експлуатації для індивідуальних транспортних засобів [77, 82, 83].

На рис.2.14 показано, що VMMS має три основні шари. На першому рівні генеруються дані. Сканер OBD II підключається до автомобіля через порт OBD II. На ринку доступно багато таких сканерів; у даній системі використано ELM 327 Bluetooth (ELM 327), який в основному є мікроконтролером. Даний сканер ELM327 може відображати більше 1500 значень датчиків. Цей сканер діє як проміжний пристрій між автомобілем і портативним пристроєм, тобто

смартфоном або ноутбуком з підтримкою Bluetooth. Для цього зв'язку між автомобілем і портативним пристроєм створено крихітні ІС (інтегральні схеми), які генерують всі дані датчика у вигляді діагностичних кодів (DTC), які генеруються під час руху автомобіля, або зупинки, і надсилаються на пристрій. Фактично смартфон зв'язується з блоком управління автомобіля через бездротове з'єднання, яке використовується в якості джерела для підключення автомобіля до внутрішнього сервера.

На рівні обробки даних першим кроком є виділення ознак, в якому потік даних DTC фільтрується в процесі виділення особливостей. Потім до набору даних застосовується принциповий компонентний аналіз (PCA) для вилучення ознак. Після цього на етапі класифікації використовуються чотири алгоритми класифікації:

- K-Nearest-Neighbor (KNN);
- Support Vector Machines (SVM);
- Random Forest (RF);
- Neural Network (NN) [43].

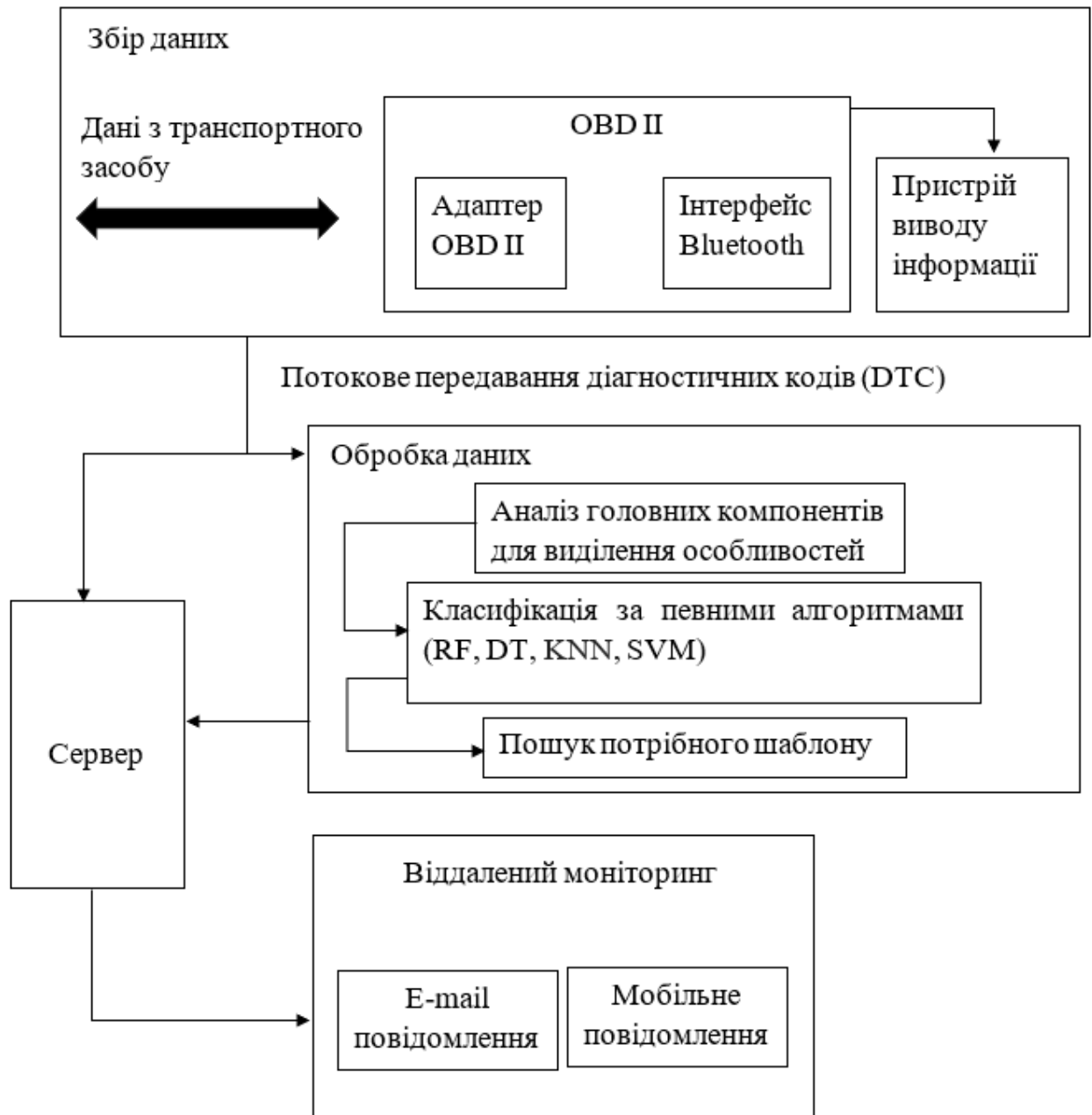


Рисунок 2.14 – Архітектура системи моніторингу віртуальних машин (VMMS) [84].

Виявляються найбільш характерні комбінації DTC або типів зв'язків, і подальша обробка виконується на сервері. Ці результати зберігаються на сервері для їх подальшого розпізнавання, яке використовується для прогнозування несправностей та дистанційного моніторингу автомобіля [84].

На рівні віддаленого моніторингу власник може дистанційно контролювати поточний стан транспортного засобу, наприклад, рівень палива, стан компонентів, поточне місцезнаходження, тощо. Та отримувати повідомлення про несправність будь-якої підсистеми транспортного засобу за рахунок використання автоматичного сповіщення.

Для отримання даних аналізуються чотири основні системи автомобіля: система запалювання, паливна система, вихлопна система та система охолодження. Ці системи відстежуються складними комп'ютерними системами управління. Датчики передають інформацію про несправності в блок управління електронікою (ECU). ECU - це мікрокомп'ютер, який складається з безлічі електронних компонентів і схем, включаючи багато напівпровідникових приладів. Його вхідний пристрій отримує вхідні дані у вигляді електричних сигналів. Ці сигнали надходять від багатьох датчиків, розташованих у різних місцях двигуна. Блок обробки порівнює вхідні дані з даними, що знаходяться в пам'яті. Блок пам'яті містить базову інформацію про роботу двигуна. Пристрій передачі імпульсів подає електричні сигнали на електромагнітний клапан впорскування. Основна функція ECU в електричній системі впорскування палива полягає в управлінні пульсацією через форсунку, частотою обертання холостого ходу, моментом запалювання і паливним насосом. Блок управління може швидко пристосуватись до зміни умов, використовуючи запрограмовану карту характеристик, що зберігається в блоці пам'яті. Мета роботи блоку управління - максимізувати потужність двигуна з мінімальним рівнем викидів вихлопних газів і мінімальною витратою палива. Інструменти сканера OBD II завантажують коди несправностей бортової мережі, зв'язуючись з блоком управління, щоб визначити, який саме датчик не відповідає [84-90].

У сучасних транспортних засобах зі збільшенням кількості електронних блоків керування зростає і рівень складності бортових мереж, оскільки кожен окремий компонент має різні вимоги до швидкості передачі даних і затримок.

Кількість електронних блоків управління в сучасних інтелектуальних транспортних засобах постійно зростає, щоб забезпечити сучасний просунутий автомобіль різноманітними новими функціональними можливостями для безпеки, захисту, зручності і т.д. [87]. Крім того, для з'єднання великої кількості електронних блоків управління було розроблено кілька автомобільних протоколів, і дослідження все ще тривають з метою створення більш досконалих функцій. Крім того, ECU (рис. 2.15) зазвичай підключаються до більш ніж однієї шини через їх диверсифіковану функціональність, включаючи управління та моніторинг транспортного засобу [88].



Рисунок 2.15 – Внутрішня конфігурація ECU [87].

У сучасних транспортних засобах для забезпечення безперебійної роботи необхідний обмін інформацією в режимі реального часу між різними модулями (рис. 2.16), і ця вимога виконується завдяки швидкому розвитку різних додатків CAN. Інші неоднорідні та складні архітектури автомобільних мереж включають такі мережі, як:

1) *CAN*. Для бортового зв'язку, як правило, використовується протокол CAN. CAN-пакети передаються між різними блоками управління через з'єднані між ними шини. У протоколі CAN використовується механізм широкомовної передачі даних. Протокол CAN має різні переваги, такі як простота, низька мережева складність і зниження витрат на електропроводку, оскільки CAN використовує архітектуру мультиплексного каналу для усунення потреби в складному надмірному монтажі проводів для зв'язку між різними блоками управління автомобілем. CAN ідентифікатори є унікальними в кожній з двох систем шин (CAN Low та CAN High). Мости в середині шини реалізують механізм збереження і пересилання повідомлень або їх частин в незалежній передачі з затримкою в часі. Мости відрізняються від повторювачів тим, що вони пересилають повідомлення, які не є локальними, в той час як повторювачі пересилають всі електричні сигнали, включаючи CAN ідентифікатор. Шлюз забезпечує з'єднання мереж з різними протоколами вищого рівня. Тому він виконує трансляцію даних протоколу між двома системами зв'язку. Відповідно, CAN не може забезпечити продуктивність в реальному часі. Відсутність механізму аутентифікації та шифрування вважається основним обмеженням протоколу CAN. До того ж, швидкий прогрес в автомобільних технологіях вимагає підтримки високошвидкісних внутрішніх протоколів, але CAN має певні обмеження щодо пропускну здатності [92].

2) *Локальна мережа з'єднання (LIN)*. LIN - це однопровідна мережа для з'єднання датчиків та виконавчих пристроїв. Надійність LIN не на висоті в порівнянні з CAN, тому вона не підходить для критично важливих завдань. LIN використовує біти парності та контрольні суми для визначення помилкових сигналів у мережі [93]. Сукупність різних комунікаційних потреб в області тіла призводить до ієрархічної мережевої архітектури, де інтегровані мехатронні підсистеми на основі недорогих мереж з'єднані між собою за рахунок використання магістралі CAN [94]. LIN-кластер складається з одного «головного» вузла і декількох «підлеглих» вузлів, підключених до спільної шини. Для досягнення низької вартості реалізації фізичний рівень визначається як один провід зі швидкістю передачі даних, обмеженою до 20 кбіт/с через обмеження щодо електромагнітних завад. Головний вузол вирішує, коли і який фрейм має бути переданий згідно з таблицею планування. Таблиця планування є ключовим елементом LIN; вона містить список пакетів, які повинні бути відправлені, і відповідні їм слоти, забезпечуючи таким чином визначеність у порядку передачі. У момент, коли фрейм заплановано на передачу, головний вузол надсилає запит на передачу (свого роду командний фрейм), що містить запит на передачу, запрошуючи підлеглий вузол надіслати свої дані у відповідь. Будь-який зацікавлений вузол може прочитати пакет даних, переданий по шині. Як і в CAN, кожне повідомлення має бути ідентифіковане з доступних 64 різних ідентифікаторів повідомлень [95].

3) *CAN FD*. Протокол керування з гнучкою швидкістю передачі даних забезпечує передачу даних по двох паралельних каналах як в синхронному, так і в асинхронному режимі. FD може бути застосований для критично важливих за часом задач. Хоча FD має функції підвищення надійності та відмовостійкості, але вартість його впровадження дуже висока. FD працює з логічними помилками, використовуючи контрольні суми та механізми резервного кодування [96]. FD – це двоканальний протокол з високою швидкістю передачі

даних до 20 Мбіт/с (10 Мбіт/с на канал). Він є стійким до відмов і визначеним, і призначений для розширених додатків, таких як X-by-Wire. На відміну від CAN, FD не має єдиної топології. Замість цього мережі можуть бути налаштовані різними способами, такими як шина, зірка або їх комбінації. Щоб ще більше розширити можливості конфігурації, FD має два канали, які можна задавати окремо. Це дозволяє здійснювати передачу даних по одному каналу, не використовуючи інший, тим самим зберігаючи пропускну здатність. Для додатків, відповідальних за безпеку, дані можуть бути передані одночасно по обох каналах, забезпечуючи вбудовану функціональну відмовостійкість мережі. Якщо один канал буде пошкоджено, передача продовжуватиметься без перерви по іншому каналу. Це дуже важливо, якщо необхідно впровадити систему дистанційного керування, щоб у разі виходу з ладу одного з каналів у водія залишався повний контроль над гальмами і рульовим керуванням автомобіля [97].

4) *MOST*. Це мультимедійна мережа, розробка якої була ініційована в 1998 році MOST Cooperation. MOST забезпечує передачу аудіо- та відеоданих за схемою "точка-точка" зі швидкістю 24,8 Мбіт/с. Це підтримує такі додатки кінцевого користувача, як радіоприймачі, навігаційні системи глобального позиціонування (GPS), дисплеї для відтворення інформації та мультимедійні системи, радіо. Фізичний рівень MOST - це пластикове оптичне волокно (POF), яке забезпечує набагато кращу стійкість до електромагнітних завад і вищу швидкість передачі, ніж класичні мідні дроти [98]. Хоча протоколи MOST задовольняють інформаційно-розважальні вимоги, але протоколи MOST не можуть забезпечити вимоги до ширини смуги пропускання при експоненційному зростанні вимог. MOST має дві основні вимоги до функціональності системи шин:

- Проста структура системи, що базується на функціонально-орієнтованій концепції.

- Передача поточкових і пакетних даних, а також управляючої інформації.

Зв'язок з функціональними блоками відбувається за рахунок використання прикладного протоколу, який складається з графічних символів і не потребує адреси пристрою. Таким чином, забезпечується просте і швидке проектування системи інформаційно-розважальної сфери з високим рівнем узагальнення. Друга вимога досягається за рахунок використання фреймової структури. Вона синхронно передає мультимедійні дані, може здійснювати асинхронну передачу великої кількості даних, не впливаючи на їх синхронну передачу, а також пропонує канал управління для команд управління і сигналів про стан. Цей протокол належить до прикладного рівня. Канальний рівень заснований на контролері мережевого інтерфейсу MOST.

Однак у практичних випадках діагностики існує багато різних станів несправності для одного і того ж механічного обладнання. Тоді для побудови послідовної системи діагностування для ідентифікації несправностей і розпізнавання типів несправностей застосовується теорія можливостей, як показано на рис. 2.17 [99]. Теорія можливостей - це математична теорія для отримання нечіткого висновку при обробці нечіткої інформації [100], яку можна представити у вигляді:

$$p(x_i) = \sum_{k=1}^N \min\{\lambda_i, \lambda_k\}, (i, k = 0, 1, 2, \dots, N), \quad (2.5)$$

$$\lambda_i = \int_{x_{i-1}}^{x_i} \frac{m}{\eta} \left(\frac{x-x_0}{\eta}\right)^{m-1} \exp\left\{-\frac{(x-x_0)^m}{\eta}\right\} dx, x_i = \bar{x} + \frac{i-N}{N} \sigma, \quad (2.6)$$

$$\lambda_k = \int_{x_{k-1}}^{x_k} \frac{m}{\eta} \left(\frac{x-x_0}{\eta}\right)^{m-1} \exp\left\{-\frac{(x-x_0)^m}{\eta}\right\} dx, x_k = \bar{x} + \frac{k-N}{N} \sigma, \quad (2.7)$$

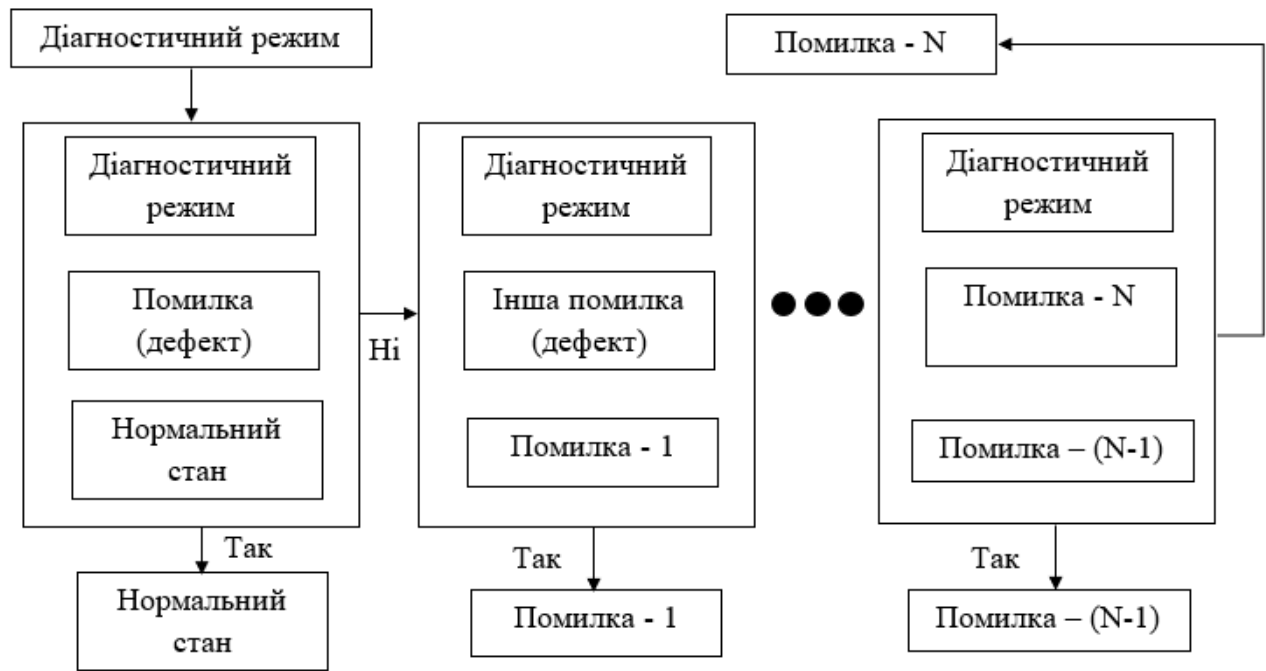


Рисунок 2.17 – Система нечіткої діагностики [99]

Звідси можна отримати функції можливостей для відомих станів та стану діагностування. Ці стани, такі як нормальний стан і різні види несправностей, які були точно діагностовані, будуть використовуватися для визначення стандартів діагностування в якості зразкових станів. Звичайно, є деякі несправності, які взагалі ніколи не виникали, і тоді ці стани називаються «невідомими станами». Таким чином, не існує жодного способу обчислити безпосередньо функцію ймовірності невідомого стану. Для того, щоб підвищити діагностичні можливості та покращити чутливість ідентифікації, в роботі [99] був запропонований наступний непрямий метод для отримання функції ймовірності невідомого стану:

$$M_{un} = \max\{1 - \sum_{i=1}^N M_i, 0\}, (i = 1, 2, \dots, N), \quad (2.8)$$

де

M_i – функції можливості стану,

N – загальна кількість станів.

Принцип дії зв'язку між транспортними засобами полягає в отриманні даних, які зчитуються з бортового комп'ютера автомобіля. На рисунку 2.18 зображено блок-схему системи OBD-II для отримання даних. Мікропроцесор періодично надсилає запити на отримання ідентифікатора параметрів до електронного блоку керування та отримує відповіді і дозволяє завантажувати дані на сервер для моніторингу в режимі реального часу. Для перетворення повідомлень між протоколом OBD-II і послідовними даними можливо використати мікросхему STN 2120, яка абсолютно стабільно працює з набором команд ELM AT 327. Кабель живлення OBD-II був застосований для конвертації 12 В з порту OBD-II в 5 В для роботи мікрокомп'ютера [65]. Деякі датчики функціонують як вхідні дані від системи ECU, а дані від ECU підключаються за рахунок використання лінії CAN - шини.

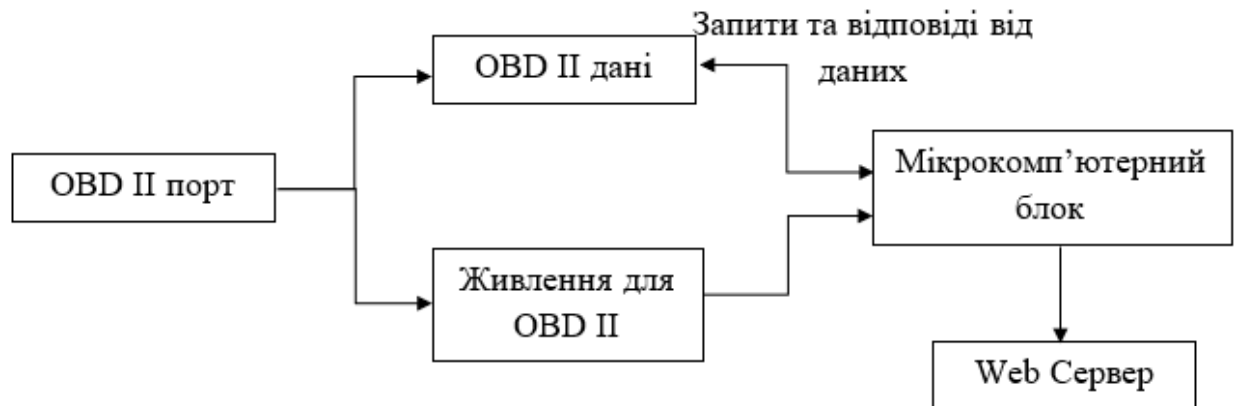


Рисунок 2.18 – Структурна схема отримання даних з OBD II

2.5. Побудова сховища даних

Однак, для обробки основних числових даних (категоріальних змінних) в історичних даних про технічне обслуговування необхідне одномоментне кодування. Проте, коли кількість категорійних змінних значно перевищує кількість числових змінних, набір даних стає малочисельним і його важко аналізувати [102]. Тоді алгоритм глибокого навчання використовується для

навчання нейронної мережі на основі одномоментно зашифрованих даних, щоб отримати низькорозмірні та надійні дані [103]. Також можна використати географічні дані (рис. 2.19). Ці дані поєднуємо з обробленими історичними даними технічного обслуговування як новий набір даних. Новий набір цих даних нормалізується і рандомізується перед тим, як він буде використаний для моделювання. Далі створюється модель глибокого навчання і визначити тип нейронної мережі відповідно до типу даних. У нейронній мережі вага вхідного шару вказує на значимість параметра. Отже, виділення ваги вхідного шару нейронної мережі є корисним для визначення відповідності між ознаками та результатами.

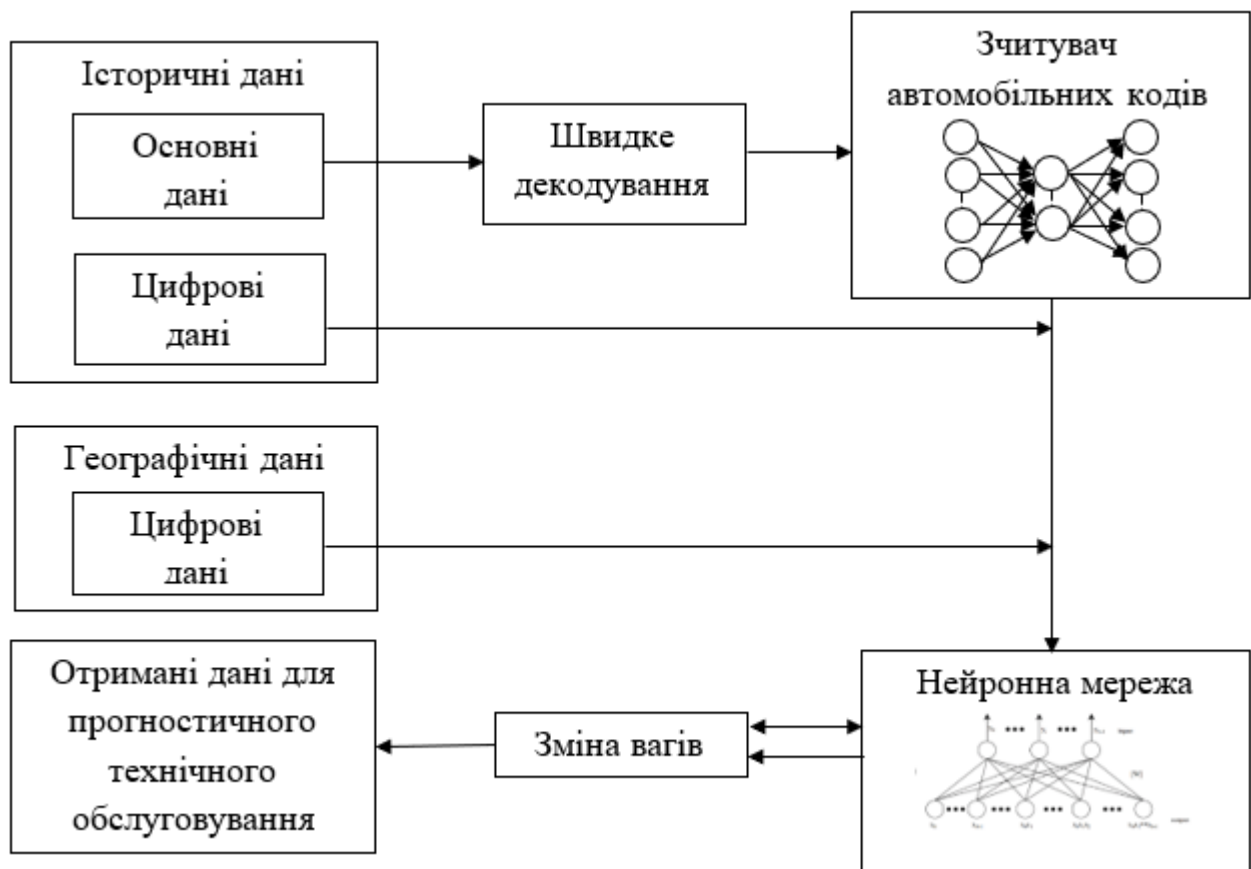


Рисунок 2.19 – Доповнена схема отримання даних

Основними етапами при зборі даних є:

- налаштування ініціювання зв'язку – встановлення конкретного шляху до com-порту, до якого підключено зчитувач даних, а також швидкості передачі даних;

- налаштування конфігурації – встановлення формату відповідей на запити даних;

- запити параметрів і фіксація відповідей – зациклення кожні декілька секунд для безперервного надсилання запитів параметрів до зчитувача і збереження відповідей необроблених даних [104].

Коли мікропроцесор (мікрокомп'ютерний блок) виявляє наявність інтернет-з'єднання, необроблені дані з EBU автоматично передаються на веб-сервер за рахунок використання пост-запитів. Відповідно, потрібно розробити підпрограму, яка у фоновому режимі роботи основної програми реєстрації даних періодично здійснює перевірку наявності з'єднання з мережею, оскільки мікропроцесор не завжди може бути під'єднаний до Інтернету.

Дані, які збираються та вивантажуються на сервер з кожного параметричного запиту закодовані. При розміщенні на веб-сервері потрібна подальша обробка для перетворення необроблених даних в інформацію, яку можна відстежувати і виводити на користувацький інтерфейс. За рахунок використання швидкого декодування частина номінальних даних були перетворені для збільшення розмірності та значно збільшило розрідженість даних. Для того, щоб зменшити розрідженість даних доцільно використовувати автокодер, алгоритм глибокого навчання, для кодування розріджених даних до надійного представлення. Автокодер представляє собою модель, що містить три повністю з'єднані шари та використовується як базовий структурний блок для побудови глибокої структури. Однак, навчання з великою кількістю нерелевантних ознак у даних все ще є складним завданням, а сучасні версії

автокодерів ще не розділяють приховані одиниці на дві частини, одна з яких релевантна задачі, а інша - нерелевантна задачі [105, 106].

Разом з базою даних такі дані повинні зберігатись в сховищі даних (СД) [101] з можливістю постійного підключення, наповнення та отримання з неї інформації. Сховище даних – це комбінація різних концепцій і технологій, які допомагають підприємствам управляти і зберігати історичні дані, отримані з операційних і прикладних програм [107]. Сховище даних містить набір логічних даних, відокремлених від оперативної бази даних, і є їх зведенням. Це забезпечує єдиний механізм доступу для отримання інформації та її подальшого аналізу для прийняття рішень. СД використовує техніку моделювання даних, яка називається технікою розмірного моделювання. Димензійнальне моделювання - це модель на основі викликів, яка підтримує високорівневий доступ до запитів. Атрибути для аналізу прийняття рішень включають показники ефективності, експлуатаційні метрики, розміри агрегатів та всі інші метрики, необхідні для аналізу ефективності роботи ТЗ.

Однак створити та керувати системою СД складно. Інструменти планування та аналізу використовуються для проектування та редагування схем, уявлень, скриптів, сценаріїв, правил, запитів і звітів. Інструменти планування та аналітики використовуються для моделювання сценаріїв «що, якщо», наприклад, для розуміння впливу змін у схемі або частоти оновлення, а також для планування ресурсів [108]. Інструменти управління сховищем (наприклад, HP Intelligent Warehouse Advisor, IBM Data Hub, Prism Warehouse Manager) призначені для моніторингу сховища, надання статистичних даних і рекомендацій: використання розділів і зведених таблиць, час виконання запитів, типи і періодичність деталізації або згортання, тощо. Скрипти, що зберігаються в сховищі, і можуть запускатися на виконання з певною періодичністю, на вимогу або при настанні певних подій. Механізм робочого процесу забезпечує успішне завершення процесу, постійно записує інформацію про успіх або

невдачу кожного кроку і забезпечує відновлення після невдачі шляхом часткового відкату, повторної спроби або переходу до наступного кроку.

Сховище даних складається з Datamart (частина сховища даних, яка містить вміст декількох невеликих наборів сховищ даних), OLTP, OLAP (аналітична обробка даних в режимі онлайн) і Star Schema (багатовимірна модель, яка забезпечує загальний огляд бази даних структури даних та оптимізує її для використання у сховищах даних). Підходи до створення сховищ даних прийнято поділяти на дві категорії: підходи, що базуються на даних, та підходи, що базуються на технічних вимогах. Перший підхід передбачає глибокий аналіз даних, що містяться у внутрішніх та/або зовнішніх базах даних, і виводить на основі цього багатовимірну схему віртуального сховища даних [107, 109, 110]. Підходи, керовані вимогами, починаються з більш раннього етапу вимог, який фокусується на моделюванні користувацького аналізу, а на подальших етапах здійснюється узгодження між джерелами даних і вимогами [74].

Архітектура сховища даних (рис. 2.20) – це структура, побудована на принципах розуміння того, як дані переміщуються в системі. Архітектурними характеристиками сховища даних є [76]:

- дані беруться з наявних інформаційних систем, баз знань та файлів;
- дані об'єднуються та перетворюються перед тим, як зберігатися у сховищі даних;
- сховище даних - це база даних тільки для читання, створена для прийняття рішень;
- доступ користувачів до сховища даних здійснюється за рахунок використання інтерфейсного інструменту або додатку.



Рисунок 2.20 – Архітектура інтеграції даних

Автори [112] визначають, що методологія проектування сховища даних складається з 9 етапів: вибір технологічного процесу, вибір елементів, ідентифікація та налаштування розмірності, вибір фактичних даних, збереження первинних обчислень у базі даних, заповнення таблиць розмірності, вибір періоду функціонування бази даних, повільне відстеження змін розмірності, а також визначення пріоритету запитів та режиму роботи з даними.

Процес збору даних здійснювався за рахунок використання аналізу первинних даних (дані які передаються напряму з датчиків), вивчення вторинних даних (історичні дані) та стандартизації інформації в сховищі даних. Після встановлення стандартизованих форматів необхідно упорядкувати дані за рахунок використання раціональної структури даних. Завдання щодо структури даних включають в себе наступні [113]:

- проста для розуміння і використання;
- легкість в керуванні;
- легкість розповсюдження та обміну великими обсягами даних;
- швидке та легке отримання великих обсягів даних;
- адаптивність до змін у розташуванні пристроїв збору даних;
- надійність.

Один з можливих способів організації даних - деревоподібна структура (рис. 2.21), яка зазвичай використовується у файлових системах. Ця модель є надійною та ефективною, що підтверджено багатьма файловими системами в різних операційних системах. У деревоподібній структурі файли організовані у вигляді каталогів і підкаталогів.

Сховище даних передбачає зберігання як неструктурованих даних (сирих), так і структурованих, тобто оброблених даних. На цьому етапі ми зіштовхуємося з типовою проблемою інтеграції різнорідних даних. Відомо, що використання онтологій є найкращим способом вирішення семантичної неоднорідності та забезпечення інтеграції даних [114]. Коли дані переходять з операційного середовища до Сховища, можливі неузгодженості і дублювання повинні бути усунені, щоб Сховище могло забезпечити інтегроване і узгоджене уявлення про дані ТЗ. Таким чином, у рамках Сховища даних інтегрування та вирівнювання даних - це процес отримання даних з джерел і забезпечення їх доступності у Сховищі [115].

Запропонована модель сховища даних (DW – Data Warehouse) визначає модель контексту як список пар елементів {(Атрибут, Значення)}. Оскільки концепція описується як набір пар даних, модель легко реалізується і дозволяє інтегрувати контекстні дані в модель DW. Вона також дозволяє виконувати навігацію по контекстних списках за кількома вимірами. Фактично, кожен параметр моделі може бути перетворений на вимір DW. Крім того, деякі атрибути, що визначають зміст, містять ієрархію. Ієрархія вимірів описує логічну структуру, що використовує впорядковані рівні для організації та групування даних.

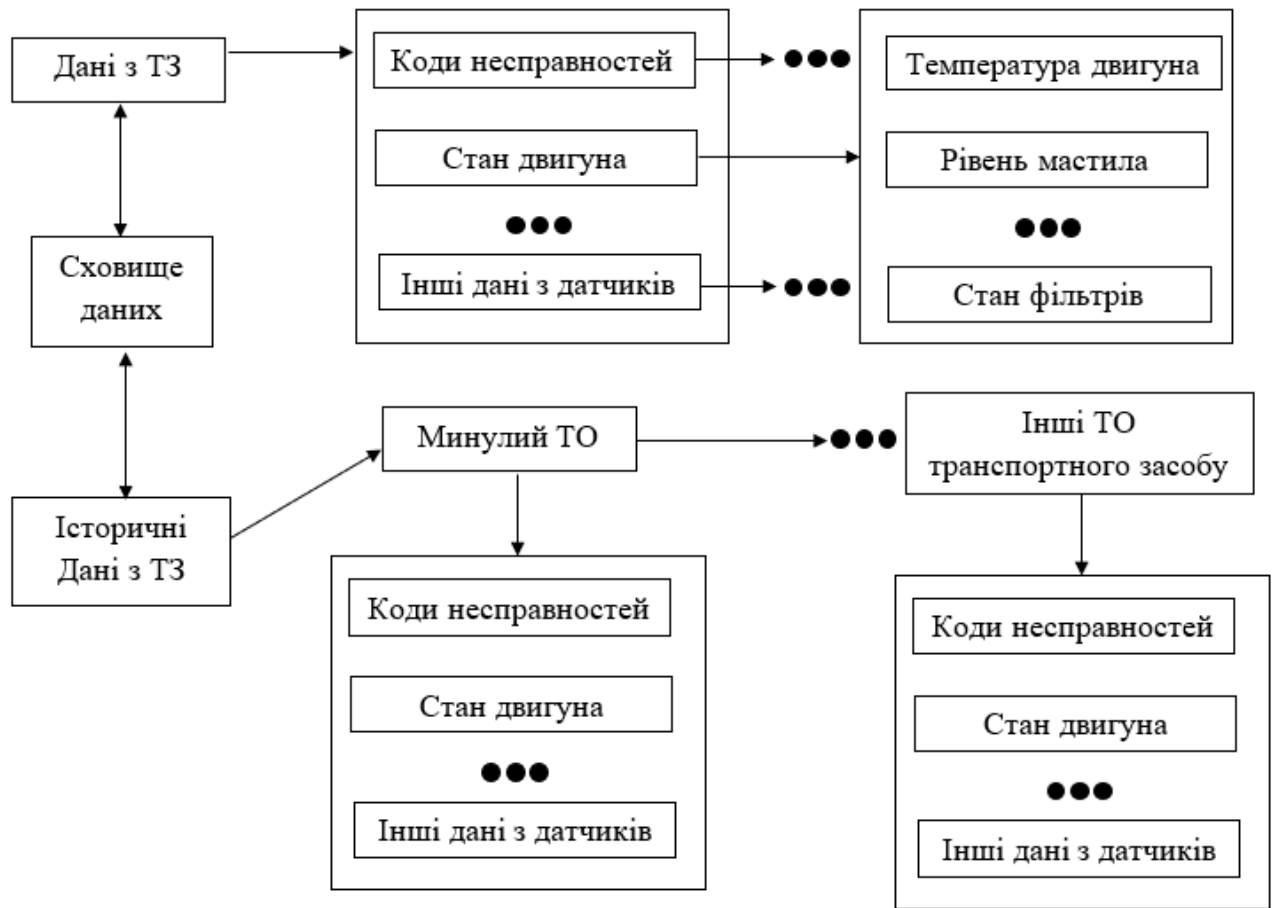


Рисунок 2.21 – Деревоподібна структура завантаження даних в СД

Основою сховищ даних є отримання, перетворення і завантаження даних з різноманітних вихідних систем в інтегрований репозиторій. Оскільки дані розподіляються між різноманітними системами-джерелами, виникають різні проблеми інтеграції (наприклад, різні формати або представлення одних і тих самих об'єктів), які вирішуються за рахунок використання процесу ETL [116]. Дані можна представити і дослідити за рахунок використання інструментів звітування та інформаційних таблиць або ввести в моделі інтелектуального аналізу даних для отримання прогнозів і висновків на основі аналітичних матеріалів

Для переважної більшості систем управління даними існують дві основні форми [117]:

- 1) системи управління для обробки великих обсягів даних про операції в режимі реального часу;
- 2) сховища даних для полегшення доступу до інформації шляхом створення централізованої бази даних для всіх даних у вигляді сховища, організованого спеціально для запитів.

Однак через неможливість впоратись традиційними методами ETL-процесів з великими обсягами інформації, потрібно застосовувати спеціальні методи для роботи з величезними обсягами вихідних даних, напівструктурованих і неструктурованих даних, складною обробкою даних, швидким прийняттям рішень і якістю наданих даних [118].

В дослідженнях [119-122] прагнули вирішити проблему отримання даних Big Data, проте вони переважно зосереджувалися на розгортанні технологій, не беручи до уваги значення стратегічного моделювання. Відповідно, для такої роботи потрібні методи, які засновані на розгортанні паралельних і розподілених обчислень (HDFS (Hadoop Distributed File System) для зберігання і MapReduce для обробки) [118], на використанні систем управління базами даних NoSQL або на інших технологіях, які працюють з Big Data.

Для накопичення та керування великими масивами даних потрібен спеціальний підхід до підвищення продуктивності ETL-процесів. Одним зі засобів швидкого запуску аналізу Big Data за рахунок скорочення часу, необхідного для трансформації даних є платформа Hadoop [123, 124]. Через багатопланову структуру трансформацій ефективність ETL-процесів здебільшого залежатиме від того, наскільки ефективно виконуються перетворення, що є етапом, який обмежує швидкість ETL-процесу. MapReduce спрощує одночасну обробку, розділяючи дані (рис. 2.22) на менші фрагменти і обробляючи їх паралельно на серверах Hadoop. Зрештою, він об'єднує всі дані з кількох серверів, щоб повернути в систему зведений результат.

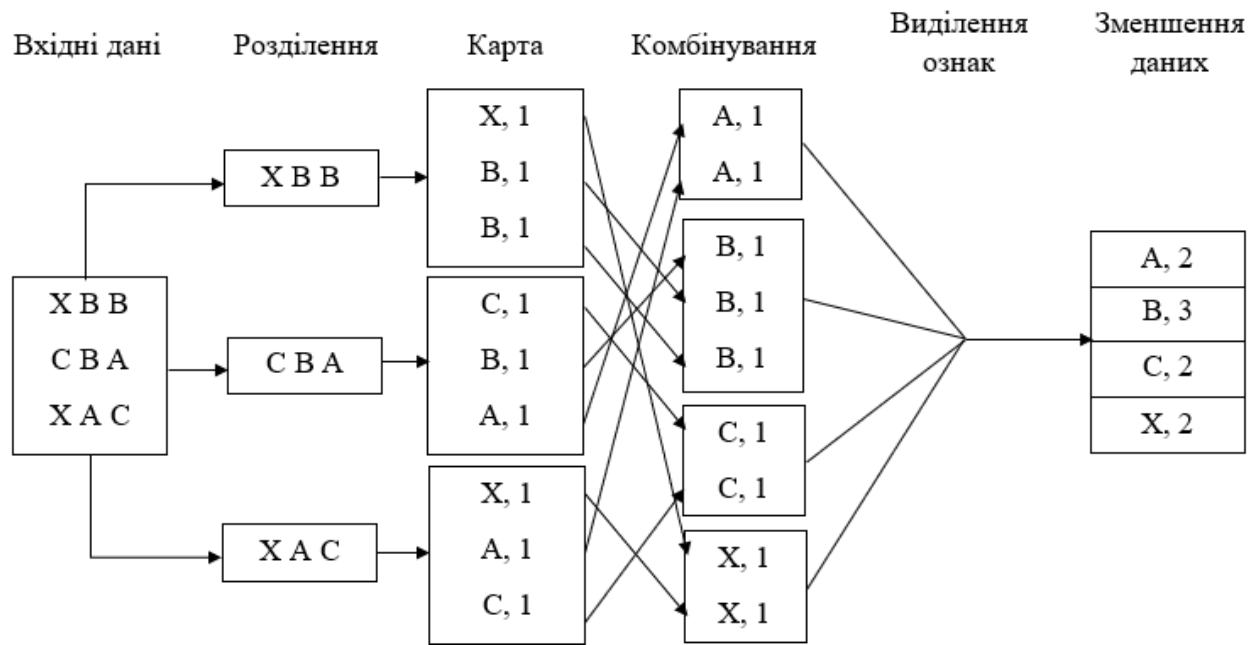


Рисунок 2.22 – Приклад роботи MapReduce [125]

Hadoop [126, 127] – це програмна реалізація фреймворку MapReduce з відкритим вихідним кодом для запуску додатків на великих кластерах, побудованих на товарному обладнанні від Apache [128]. Характерними рисами Hadoop є розділення даних та паралельні обчислення великих наборів даних. Зберігання та обчислювальні можливості масштабуються з додаванням обчислювальних кластерів, що складаються з тисяч вузлів, до кластера Hadoop.

В основі MapReduce [129] лежать дві функції: Map і Reduce. Вони виконуються послідовно одна за одною:

- функція map приймає з нього дані у вигляді пар <ключ, значення>, обробляє їх і видає на виході інший набір проміжних пар <ключ, значення>;
- функція reduce також отримує вхідні дані у вигляді пар <ключ, значення> і видає на виході пари <ключ, значення>.

Типи ключів і значень можуть відрізнятися залежно від конкретного сценарію застосування. Всі вхідні та вихідні дані зберігаються у HDFS. У той час як карта є необхідним кроком для фільтрації та сортування вихідних даних,

функція скорочення є необов'язковою. Ключ може бути штучним або автоматично згенерованим, а значенням може бути рядок, json, BLOB (двійковий великий об'єкт) і т.д. Тип «ключ-значення», в принципі, полягає у використанні геш-таблиці, де є унікальний ключ і вказівник на конкретний елемент даних. Продуктивність буде значною мірою зростати завдяки використанню механізмів кешування. Щоб прочитати значення, потрібно знати і ключ, і сховище де він є, тому що справжній ключ - це хеш (сховище та ключ) [130, 131].

Модель «ключ-значення» має найпростіше представлення. Її структура складається зі списку пар, що складаються з ключа та значення. Така структура бази даних типу «ключ-значення» може бути корисною, але вона також має певні недоліки. По-перше, ця структура не забезпечує жодних традиційних функцій баз даних (наприклад, атомарність операцій або узгодженість, коли кілька операцій виконуються одночасно). По-друге, зі збільшенням обсягу даних може стати більш проблематичним збереження унікальних значень, таких як ключі; вирішення цієї проблеми вимагає певної складності у створенні рядків символів, які залишатимуться унікальними серед величезного набору ключів.

Загалом, системна структура складається з трьох основних частин (рис. 2.23):

1. підсистема збору даних,
2. підсистема зберігання даних
3. підсистема аналізу даних.

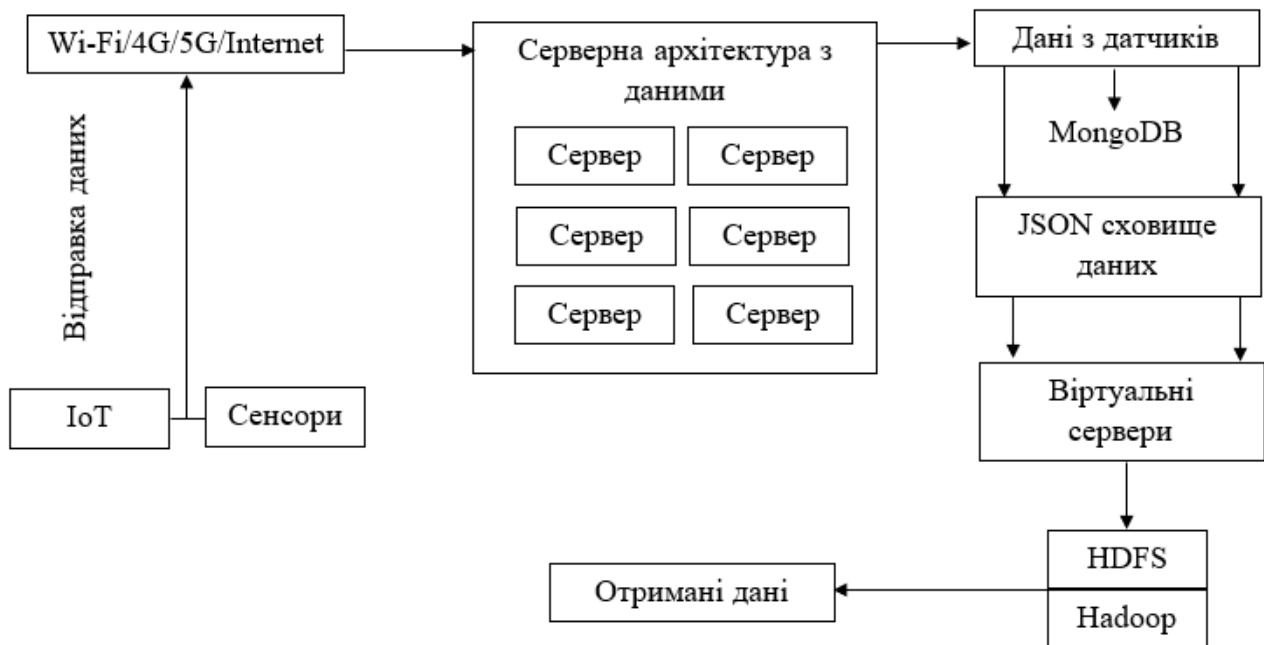


Рисунок 2.23 – Архітектура системи

Система базується на програмному забезпеченні з відкритим програмним кодом і забезпечує розподілену та масштабовану структуру для підтримки будь-якої кількості датчиків за потреби. За рахунок використання IoT можливо проводити моніторинг та збір даних в реальному часі в ТЗ, що в свою чергу забезпечує можливість взаємопов'язаним транспортним засобам спільно обмінюватися даними та обробляти їх за рахунок використання підключення до мережі. Датчики IoT в транспортних засобах найчастіше використовують в режимі реального часу для моніторингу стану автомобіля, а також для перегляду попередніх результатів експлуатації [132]. Датчики, які встановлені на транспортних засобах, мають можливість передавати дані про свій стан через Wi-Fi/4G/5G/Internet. Проте в багатьох випадках автомобільні IoT-додатки не передбачають глибокого аналізу отриманих даних [133].

З боку фреймворку на етапі отримання вхідного потоку журналів даних (логів) необхідне програмне забезпечення, що керує цим процесом. Ними можуть бути:

- Apache Flume - Apache Flume - це розподілене програмне забезпечення

для ефективного збору, об'єднання та транспортування великих обсягів даних логів [134]. Він має архітектуру, що базується на потоковому передаванні даних. Він базується на розширюваній моделі даних, що дозволяє проводити аналіз в режимі онлайн. Flume може використовуватися для надійного транспортування великих обсягів даних про події: Вхідне джерело Flume отримує дані про події від зовнішнього джерела; після цього дані поміщаються в канал і вилучаються з нього тільки після їх збереження в наступному каналі або в серверному репозиторії. Це гарантує, що набір даних надійно передаватиметься від однієї точки до іншої в потоці.

- Apache Kafka - Apache Kafka є розподіленою потоковою платформою для управління інформаційними потоками [135]. Платформа може розміщувати та отримувати підписки на потоки записів, подібно до черги повідомлень, читаючи вхідні дані та виводячи дані в тому ж потоці. Програмне забезпечення також дозволяє працювати з даними в міру їх надходження, що дає змогу трансформувати дані в процесі їх отримання.

Поєднання двох фреймворків забезпечує можливість виконувати операції з затримкою до секунди і значно полегшує масштабування при великому обсязі даних [131].

Також можна зазначити, що переважна більшість баз даних SQL розроблена для підтримки чотирьох основних характеристик управління даними, відомих як ACID (атомарність, узгодженість, ізолюваність, надійність). Ці властивості забезпечують збереження даних навіть у випадку паралельного доступу до них та/або у випадку можливих збоїв. Властивості ACID дозволяють забезпечити, що коли операція виконується над узгодженою базою даних, вона(і) або завершиться з правильним кінцевим результатом, або завершиться без наслідків, але в обох випадках збережеться узгоджений стан даних. З іншого боку, бази даних NoSQL, включаючи бази документів, жертвують

властивостями ACID на користь підходу BASE (в цілому доступний, легкий доступ, з можливою узгодженістю) [136].

Зібрані з допомогою датчиків дані мають зберігатись у певному сховищі даних. Однак зі зростанням кількості датчиків, а отже і масиву даних, стає все більш нетривіальним завданням їх безперервного зберігання. Проте дані, отримані з датчиків, є цінними, оскільки можуть містити в собі приховані ознаки несправностей або ж діагностичну інформацію. Бази даних NoSQL з відкритим вихідним кодом є ефективною основою для збереження великих обсягів даних з датчиків

Бази даних NoSQL мають багато переваг над реляційними базами даних, але є й кілька недоліків. Наприклад, ці бази даних не мають стандартизованого інтерфейсу та мови запитів для забезпечення ефективності управління даними. Крім того, гнучкість структури даних створює труднощі для методів «вилучення та перетворення» для завдань ETL. Бази даних NoSQL не підтримують об'єднання і не мають підтримки OLTP (онлайн обробка операцій), як бази даних SQL, що ускладнює виконання складного аналізу [137]. Зі зростанням обсягів даних з'являються сховища великих даних, які потребують даних з NoSQL-джерел, а разом з ними зростає і потреба в засобах ETL-інструментів. Існує безліч програм, які здійснюють перетворення з SQL-бази в базу даних NoSQL; однак існує дуже мало інструментів, які можуть бути використані для зворотного процесу.

MongoDB – являє собою документно-орієнтовану базу даних з можливістю зберігати документи у форматі JSON. Це забезпечує високу швидкодію, постійну доступність і зручну масштабованість. Документи, що зберігаються в MongoDB, можуть бути приведені до типів даних мови програмування. Сервери MongoDB можна дублювати за рахунок використання функції автоматичного обходу відмов на головному сервері [138]. Для

масштабування баз даних автоматично використовується кластеризація, яка дозволяє рівномірно розташовувати масиви даних на різних серверах [139].

В практичному дослідженні [140] було розглянуто різні алгоритми для перетворення. В них для автоматичного виявлення схем баз даних NoSQL використовують інструмент з відкритим вихідним кодом Variety [141], аналізатор схем для MongoDB та MySQL [142]. Variety розроблений для отримання схеми однієї окремої вибірки за один раз. Далі аналізатор схем генерує нову схему у форматі JSON, яку система аналізує для формування SQL-запитів для побудови бази даних і складання таблиць. Після створення таблиць решта ключових ідентифікаторів аналізується як окремі стовпці таблиць, а первинним ключем встановлюється `_id`. Всі таблиці, створені для внутрішніх документів і масивів, мають первинні ключі у вигляді цілих ідентифікаторів з автоінкрементом разом з ідентифікатором основного документа.

З цього можемо зробити висновок, що для побудови системи DW з гетерогенними (семантичними і несемантичними) джерелами даних інтеграційний процес повинен бути здатним обробляти семантику даних. Технологія семантичних мереж (Semantic Web, SW) відповідає цим потребам, адже вона забезпечує додавання в процесі інтеграції семантики як на рівні даних, так і на рівні схем [143]. У SW для відображення та обміну даними в зрозумілому для машини форматі використовується RDF (Resource Description Framework - фреймворк опису ресурсів). У моделі даних RDF [144]. дані відображаються у формі набору операторів, які називаються RDF-трійками (RDF triples). Одним з недоліків семантичних даних є те, що вони часто описуються без будь-якої схеми (тобто, надаються лише екземпляри, а не схема) або з неповною схемою. Більше того, різні джерела описують один і той самий предметний клас за власною неявною схемою, що призводить до проблем семантичної неоднорідності [145].

Однак, зберігання даних з сенсорів є дуже важливою функцією для системи, проте ці дані необхідно аналізувати, щоб знайти важливу інформацію, таку як сигнали раннього попередження та повідомлення про несправності. Для обробки великих даних необхідна не лише обчислювальна потужність, але й система зберігання, що зазвичай реалізується за рахунок використання використання хмарних кластерів [146]. Хмарні кластери створюються з використанням декількох серверів, однак віртуалізація дозволить максимально ефективно використати ресурси та зменшити витрати на створення кластерів.

Щоб заповнювати SDW (Semantic Data Warehouse) з різноманітних джерел даних, процес інтегрування повинен передбачати врахування специфіки кожного джерела даних [147]. Крім семантичних джерел, SDW має також підтримувати несемантичні джерела, визначаючи семантичний шар над несемантичними джерелами. Окрім очистки та інших традиційних перетворень (перейменування, об'єднання, групування, упорядкування, трансформації рядків/чисел тощо), процес інтегрування має супроводжуватися описом семантики БД на рівні даних. Крім того, слід фіксувати зміни в даних, а семантичні ресурси слід пов'язувати із зовнішніми базами знань.

Відповідно, процес створення SDW з гетерогенних джерел даних вимагає процесів інтеграції. Концепція ETL-процесів для SDW об'єднує технології DW і SW з метою забезпечення інтеграції, упорядкування та керування всіма інтеграційними завданнями на семантичному рівні для створення багатовимірних SDW з різних джерел і реалізації OLAP-запитів до цих даних [145]. Таким чином, семантичний ETL покриває наступні аспекти [144]:

1. Визначення багатовимірних SDW.
2. Вилучення даних з джерел семантичних і несемантичних даних.
3. Створення відображень «Атрибут, Значення» для забезпечення автоматичної параметризації ETL-операцій.
4. Визначення сукупності високорівневих концептуальних ETL-

конструкцій (завдань/операцій) для об'єднання семантичних джерел у SDW.

5. Перетворення інформації відповідно до семантики (MD), закодованої в цільовій предметній області та файлі відображення «Атрибут, Значення».

6. Актуалізація SDW для відображення даних, змінених у вихідному джерелі з плином часу.

7. Поєднання внутрішніх даних із зовнішніми базами знань.

8. Перенесення трансформованих даних до тривимірного сховища.

9. Включення OLAP запитів на SDW.

В MongoDB дані зберігаються в кластерному вузлі (рис 2.24), і кластер MongoDB не може бути побудований без одного або декількох вузлів. Якщо вузол зазнає збою, інформація для цього кластера зберігається в його резервних копіях. Операції з даними (читання/запис) дозволяють визначити, який шар використовується. Один або декілька серверів використовують реплікацію вузла, а резервний вузол змодельований так само, як і первинний канал даних. Якщо основний сервер виходить з ладу, його функції виконуватиме один з резервних серверів. Головний сервер – це центр, який забезпечує функціонування всіх операцій (читання/запис). Зрештою, кластер синхронізується з усіма транзакціями розподіленого читання. За зберігання метаданих у кластері керує набір конфігураційних серверів [149]. Ці сервери визначають фрагменти даних, а також інформують про те, який саме фрагмент інформації належить до того чи іншого кластера. MongoDB призначає кожне завдання певному вузлу на базі відповідного типу завдання, а потім об'єднує отримані дані. Цей фреймворк використовує файли з розподіленою пам'яттю, щоб максимально ефективно використовувати наявну пам'ять, що, в кінцевому підсумку, підвищує продуктивність [150].

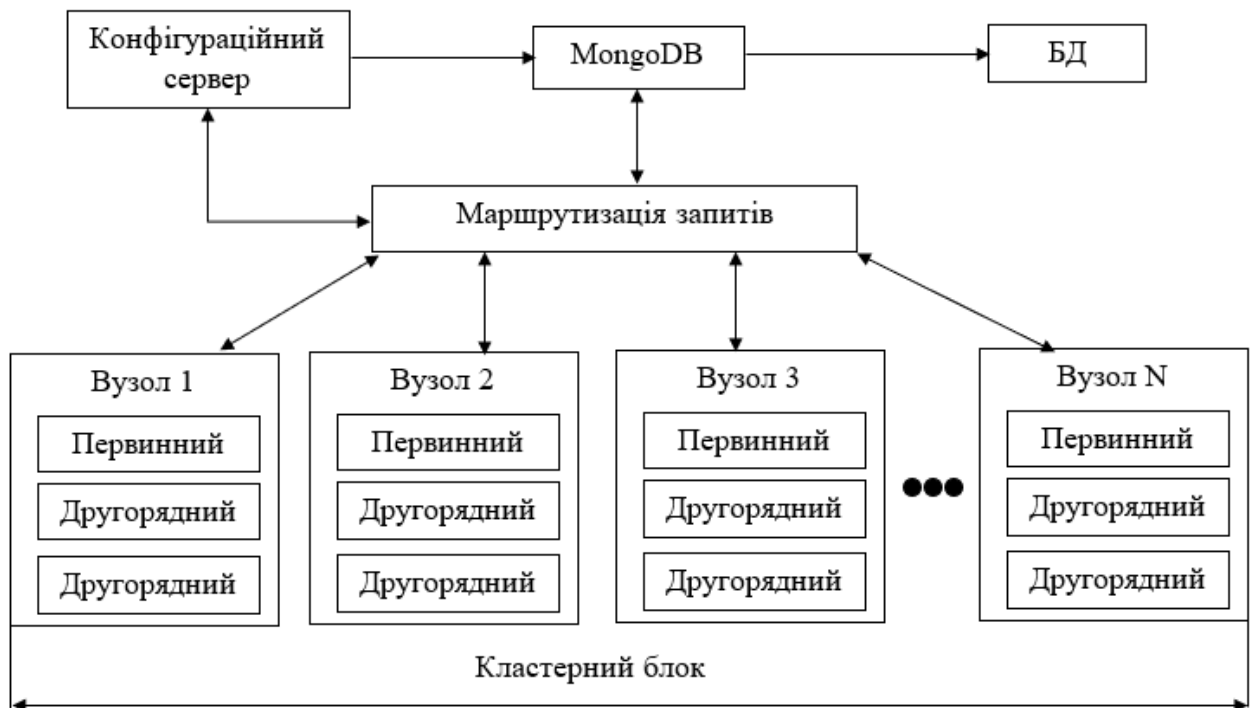


Рисунок 2.24 – Архітектура MongoDB

База даних в MongoDB представляє собою сукупність колекцій даних (рис. 2.25), які містять однорідні дані. Ці документи є JSON-об'єктами (рис. 2.26), що містять пари ключ-значення. Поля можуть підтримувати всі типи даних, що підтримуються реляційною базою даних, більше того, поле може мати масив значень, або JSON-документ як своє значення, або масив JSON-документів як своє значення. MongoDB потребує наявності первинного ключа під назвою ObjectID для кожного документа. ObjectID визначає кожен документ у базі даних MongoDB в колекції. Цей первинний ключ має ім'я поля `_id` і його значення у вигляді 24-байтового шістнадцяти-символьного рядка, які є унікальними для окремих файлів. Також можна створити додаткові вторинні індекси на основі полів, які підтримують зростаючі, спадні, унікальні, комбіновані та геопросторові ключі [151].

TIME_RUN_THRUTTE_RPM	()	VEHICLE_SPEED	()	THRUSTLE	()	ENGINE_LOAD	()	COOLANT_TEMPERATURE	()	LONG_TERM_FUEL_TRIM_BANK_1	()	SHORT_TERM_FUEL_TRIM_BANK_1	()	INTAKE_MANIFOLD_PRESSURE	()	FUEL_TANK	()	ABSOLUTE_THROTTLE_B	()	PEDAL_D	()	PEDAL_E	()																												
COMMANDED_THROTTLE_ACTUATOR	()	FUEL_AIR_COMMANDED_EQUIV_RATIO	()	ABSOLUTE_BAROMETRIC_PRESSURE	()	RELATIVE_THROTTLE_POSITION	()	INTAKE_AIR_TEMP	()	TIING_ADVANCE	()	CATALYST_TEMPERATURE_BANK1_SENSOR1	()	CATALYST_TEMPERATURE_BANK1_SENSOR2	()	CONTROL_MODULE_VOLTAGE	()	COMMANDED_EVAPORATIVE_PURGE	()	TIME_RUN_WITH_MIL_ON	()	TIME_SINCE_TROUBLE_CODES_CLEARED	()	DISTANCE_TRAVELED_WITH_MIL_ON	()	WARM_UPS_SINCE_CODES_CLEARED	()																								
0.000000		0.000000		0.000000		17.647058		0.000000		81.000000		-4.687500		0.000000		101.000000		60.392159		49.803921		16.078432		31.764706		17.254902		0.000000		100.000000		0.000000		45.000000		5.000000		424.799988		251.500000		12.382000		0.000000		0.000000					
8260.000000		0.000000		255.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000			
0.000000		0.000000		0.000000		17.647058		0.000000		81.000000		-4.687500		0.000000		101.000000		60.392159		49.803921		16.078432		31.764706		17.254902		0.000000		100.000000		0.000000		45.000000		5.000000		424.799988		251.500000		12.382000		0.000000		0.000000		0.000000			
8260.000000		0.000000		255.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000			
0.000000		0.000000		0.000000		17.647058		0.000000		81.000000		-4.687500		0.000000		101.000000		60.392159		49.803921		16.078432		31.764706		17.254902		0.000000		100.000000		0.000000		45.000000		5.000000		424.799988		251.500000		12.382000		0.000000		0.000000		0.000000		0.000000	
8260.000000		0.000000		255.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000	
0.000000		0.000000		0.000000		17.647058		0.000000		81.000000		-4.687500		0.000000		101.000000		60.392159		49.803921		16.078432		31.764706		17.254902		0.000000		100.000000		0.000000		45.000000		5.000000		424.799988		251.500000		12.382000		0.000000		0.000000		0.000000		0.000000	
8260.000000		0.000000		255.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000		0.000000	
0.000000		0.000000		0.000000		17.64																																													

Рисунок 2.25 – «Сирі» отримані дані

Значення для первинного ключа `_id` може бути створене безпосередньо під час створення файлу. Але оскільки це первинний ключ та індекс, воно має бути унікальним. Якщо документ вставляється в колекцію без поля `_id`, то MongoDB створює унікальне значення для цього поля. Якщо документ вставляється з повторюваним значенням у полі `_id`, то виникає помилка дублікату ключа.

```
[
  {
    "ENGINE_RUN_TIME ()": 0,
    "ENGINE_RPM ()": 0,
    "VEHICLE_SPEED ()": 0,
    "THROTTLE ()": 17.647058,
    "ENGINE_LOAD ()": 0,
    "COOLANT_TEMPERATURE ()": 48,
    "LONG_TERM_FUEL_TRIM_BANK_1 ()": -3.90625,
    "SHORT_TERM_FUEL_TRIM_BANK_1 ()": 0,
    "INTAKE_MANIFOLD_PRESSURE ()": 101,
    "FUEL_TANK ()": 0,
    "ABSOLUTE_THROTTLE_B ()": 49.803921,
    "PEDAL_D ()": 16.078432,
    "PEDAL_E ()": 31.764706,
    "COMMANDED_THROTTLE_ACTUATOR ()": 18.431372,
    "FUEL_AIR_COMMANDED_EQUIV_RATIO ()": 0,
    "ABSOLUTE_BAROMETRIC_PRESSURE ()": 100,
    "RELATIVE_THROTTLE_POSITION ()": 0,
    "INTAKE_AIR_TEMP ()": 36,
    "TIMING_ADVANCE ()": 5,
    "CATALYST_TEMPERATURE_BANK1_SENSOR1 ()": 200.899994,
    "CATALYST_TEMPERATURE_BANK1_SENSOR2 ()": 41.800003,
    "CONTROL_MODULE_VOLTAGE ()": 12.949,
    "COMMANDED_EVAPORATIVE_PURGE ()": 0,
    "TIME_RUN_WITH_MIL_ON ()": 0,
    "TIME_SINCE_TROUBLE_CODES_CLEARED ()": 2648,
    "DISTANCE_TRAVELED_WITH_MIL_ON ()": 0,
    "WARM_UPS_SINCE_CODES_CLEARED ()": 91
  },
]
```

Рисунок 2.26 – JSON представлення отриманих даних (пара ключ-значення)

При початкову отримані даних від ТЗ на сервери передається інформація про стан кожного вузла, такі як: робочий хід двигуна, оберти двигуна, швидкість автомобіля, положення дросельної заслінки, навантаження на двигун, температура охолоджувальної рідини, довгостроковий запас палива, короткостроковий запас палива, тиск у впускному колекторі, залишок палива в паливному баку, абсолютне положення дросельної заслінки, положення педалі, положення педалі, привід дросельної заслінки, співвідношення паливо-повітря, абсолютний барометричний тиск, відносне положення дроселі, температура впускного повітря, час запалювання, датчик температури каталізатора, датчик температури каталізатора, датчик температури каталізатора, напруга керуючого модуля, кероване випаровування, час роботи з індикацією рівня мастила, час з моменту усунення кодів несправностей, пройдена відстань із увімкненою лампою індикатора несправності, час роботи з моменту виникнення кодів несправностей та інші.

Далі за рахунок використання фреймворку MapReduce дані з файлової системи можуть миттєво зчитуватись і фреймворк може надавати відповіді на запити «на льоту», без будь-якого завантажувального етапу. MapReduce дозволяє автоматизувати процес паралельного сканування даних і застосування визначених функцій під час сканування даних. Відсутність фази завантаження в MapReduce має додатковий ефект на продуктивність, який не обмежується використанням допоміжних структур даних. Під час читання дані можуть бути стиснуті на сервері [152]. Це може підвищити продуктивність, навіть при скануванні методом перебору, за рахунок зменшення часу вводу/виводу при наступних зверненнях до даних. Крім того, так як дані не завантажуються заздалегідь, MapReduce повинен виконувати перевірку даних під час виконання при кожному зверненні до даних, замість того, щоб аналізувати дані лише один раз під час завантаження.

В роботі [153] представляють структуру для ефективного керування репозиторієм схем для сховищ JSON-документів. Пропонований підхід базується на понятті схеми JSON, яка має назву «скелет» (skeleton). Простіше кажучи, скелет - це набір дерев, що описують конструкції, які найчастіше з'являються в об'єктах збору JSON-даних. Однак разом з цим наявність повного структурного опису має важливе застосування для багатьох задач, таких як оптимізація запитів, встановлення та впровадження безпекових заходів контролю доступу, і, що важливо, надання користувачеві повного структурного уявлення про базу даних, яке може допомогти йому ефективно виконувати запити та досліджувати дані. В цій роботі складністю при проектуванні системи керування схемами для сховищ даних є пошук схеми в даних, що зберігаються в сховищі. Без чіткого визначення поняття схеми, потреба у вилученні схеми зі сховища даних призводить до необхідності проведення комплексного процесу вилучення даних, а не простого пошуку. Через відсутність вимог гарантувати лише один тип об'єкта в одній колекції, колекція може містити записи, що відповідають більш ніж одному типу об'єкта. Крім того, структури одного типу об'єктів у колекції можуть відрізнятися через розрідженість типів атрибутів у NoSQL, а також еволюцію моделі даних, спричинену високоінтерактивним додаванням і вилученням властивостей.

2.6. Висновки до другого розділу

1. Доведено обмеженість класичної моделі збору телеметричної інформації для завдань Big Data. Встановлено, що традиційний метод циклічного опитування (polling) через стандартизований інтерфейс OBD-II та CAN-шину за моделлю «запит-відповідь» має критично низьку пропускну здатність (близько 9 запитів на секунду). Обґрунтовано необхідність переходу сучасних інтелектуальних транспортних систем до потокового збору даних

(Data Streaming) у реальному часі з використанням спеціалізованих IoT-мікроконтролерів та протоколів бездротової передачі. Це запобігає втраті даних про швидкоплинні процеси та забезпечує формування якісних наборів даних для прогностичної аналітики.

2. Систематизовано функціональні можливості протоколу OBD-II як базису для автоматизованого моніторингу. Проаналізовано стандартизовані режими (Modes) з позиції їх придатності для машинного навчання. Визначено, що агрегація поточних динамічних параметрів (Live Data), «заморожених кадрів» (Freeze Frame) та діагностичних кодів несправностей (DTC) створює необхідний інформаційний простір для систем ітеративного контролю. Безперервний збір цих даних є критично важливим для моніторингу стану двигуна, дозволяючи математично оцінювати динаміку деградації та прогнозувати момент початку незворотного зношення.

3. Обґрунтовано доцільність застосування вірогідно-логічного методу локалізації дефектів на основі даних OBD-II. На основі аналізу існуючих підходів до діагностики (вірогідного, часового, мінімаксного, логічного та методу виключень) доведено перевагу їх інтеграції з потоковими даними систем бортової самодіагностики. Встановлено, що такий вірогідно-логічний підхід дозволяє відстежувати зміну контрольних параметрів онлайн та алгоритмізувати пошук аномалій, мінімізуючи вплив людського фактора. Це перетворює інтерфейс OBD-II в інструмент активного сенсорного рівня для проактивних прогностичних моделей.

4. Розроблено методологію подолання проблеми розрідженості в базах логованих даних (Logged Vehicle Data, LVD). Встановлено, що масиви LVD характеризуються високим рівнем розрідженості (понад 25% пропущених значень), що зумовлено нерегулярністю сервісних візитів та відмінностями у апаратних комплектаціях транспортних засобів. Доведено, що застосування спеціалізованих статистичних фільтрів (зокрема на основі критерію

узгодженості Колмогорова-Смірнова) у поєднанні з обгортковими алгоритмами селекції ознак (пучковий пошук) дозволяє математично обґрунтовано відновлювати пропуски або відсіювати неповні записи. Це суттєво підвищує інформативність та якість навчальних вибірок для прогностичних моделей без втрати критично важливої телеметрії.

5. Обґрунтовано архітектурний перехід до нереляційних документо-орієнтованих баз даних для збереження гетерогенних масивів LVD. Доведено, що класичні реляційні БД (SQL) з жорстко заданими схемами є неефективними при роботі з розрідженими даними різноманітних автомобільних специфікацій. Запропоновано використання NoSQL рішень (на прикладі MongoDB), де гнучкий формат документів (JSON/BSON) дозволяє динамічно зберігати гетерогенну інформацію, забезпечуючи стійкість до відмов, високу швидкість транзакцій та зручне горизонтальне масштабування кластерів при зростанні обсягів Big Data.

6. Запропоновано екосистему розподілених обчислень для усунення «вузьких місць» у процесах попередньої обробки даних (ETL). Для того щоб процеси фільтрації та екстракції розріджених даних не знижували загальну продуктивність системи, обґрунтовано застосування технологій розподілених обчислень (HDFS) та паралельної обробки (MapReduce). Це дозволяє фрагментувати величезні масиви телеметрії та обробляти їх «на льоту». Додатково доведено ефективність інтеграції брокерів повідомлень (Apache Kafka, Apache Flume) для безперебійного керування потоками телеметрії, а також використання концепції семантичних сховищ (Semantic Web) для автоматизованого узгодження семантичної неоднорідності форматів даних від різних виробників транспортних засобів.

РОЗДІЛ 3 МОДЕЛІ МАШИННОГО НАВЧАННЯ ДЛЯ СИСТЕМИ ПРОГНОСТИЧНОГО МОНІТОРИНГУ

3.1. Підготовка датасету

Вхідні дані проходять шлях від сирих розріджених логів бортових систем до збалансованого структурованого датасету. Первинним джерелом даних виступає база логованої телеметрії LVD (Logged Vehicle Data), що накопичується під час візитів автомобілів на СТО. Для підготовки враховуємо:

- наявність понад 25% пропусків через нерегулярність сервісних інтервалів та відмінності в комплектаціях ТЗ.
- для відбору найбільш інформативних ознак застосовується непараметричний критерій Колмогорова-Смірнова. Він порівнює емпіричні функції розподілу справного $F_{no_repair}(x)$ та дефектного $F_{need_repair}(x)$ станів автомобіля:

$$D_n = \max |F_{need_repair}(x) - F_{no_repair}(x)| \quad (3.1)$$

- для формування кінцевого простору ознак використовується пучковий пошук, який додає параметри, суворо контролюючи мінімальний розмір результуючої вибірки $n_{dataset}$ через коефіцієнт обмеження $constraintFactor$:

$$n_{dataset} = n_{all} \cdot constraintFactor^{N_{params}} \quad (3.2)$$

При аналізі інформації, яка поступає з OBD-II сканера потрібно побудувати інформаційну БД, в яку будуть завантажуватись всі параметри з ТЗ. Після цього за рахунок використання по-code засобів можливо побудувати прогностичну систему, яка за рахунок використання декількох

алгоритмів (Random Forest, Neural Network, Logistic Regression). Для визначення методу який буде використовуватися в системі прогностичного моніторингу та проведення експерименту використовуємо платформу машинного навчання та візуального аналізу даних Orange Data Mining [154]. На основі даних про технічний стан ТЗ [155, 156] будуємо аналітичну і прогностичну модель.

Точне прогнозування виходу з ладу та залишкового ресурсу обладнання дозволяє розробити адекватне планування технічного обслуговування. Для прогнозування використовуються джерела даних і різноманітні методи для моніторингу продуктивності обладнання, якості продукції та прогнозування можливої відмови обладнання. Моніторинг дозволяє забезпечити раннє виявлення несправностей і дозволяє виявити проблеми, що виникають, до того, як станеться відмова [157].

Для початку роботи дані приводимо до необхідного вигляду (виконується їх очищення). Самі по собі дані часто можуть мати відсутність значення або неправильне форматування. Саме попередня обробка даних дозволяє провести їх очищення, ідентифікацію, видалення дублікатів та знаходження відповідностей.

До набору даних можуть входити різні ознаки та параметри, пов'язані з роботою двигуна транспортного засобу, такі як оберти двигуна, температура, тиск та інші дані з датчиків. Також він може включати метадані про транспортний засіб, такі як марка, модель, рік випуску і пробіг. Візуально дані виглядатимуть наступним чином (рис. 3.1):

Після проходження препроцесингу вхідні дані для навчання ML-моделей представляються у вигляді статичного вектора ознак низької розмірності. Кожен екземпляр даних i ($i=1, \dots, N$, де $N=99$ інстансів) описується вектором $x_i \in \mathbb{R}^6$, який містить 6 ключових числових параметрів роботи ДВЗ:

$$x_i = [x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5}, x_{i6}]^T \quad (3.3)$$

де:

x_{i1} — оберти двигуна за хвилину (Engine rpm, RPM);

x_{i2} — тиск мастила в системі мащення (Lub oil pressure, bar);

x_{i3} — тиск палива в паливній рампі (Fuel pressure, bar);

x_{i4} — тиск в системі охолодження (Coolant pressure, bar);

x_{i5} — температура моторного мастила (lub oil temp, °C);

x_{i6} — температура охолоджувальної рідини (Coolant temp, °C).

Info				
99 instances				
7 features (no missing values)				
Data has no target variable.				
0 meta attributes				
Columns (Double click to edit)				
	Name	Type	Role	Values
1	Engine rpm	N numeric	feature	
2	Lub oil pressure	N numeric	feature	
3	Fuel pressure	N numeric	feature	
4	Coolant pressure	N numeric	feature	
5	lub oil temp	N numeric	feature	
6	Coolant temp	N numeric	feature	
7	Engine Condition	C categorical	target	need repair, no repair
Reset Apply				

Рисунок 3.1 – Вибір необхідних даних (ознаки та цільова змінна)

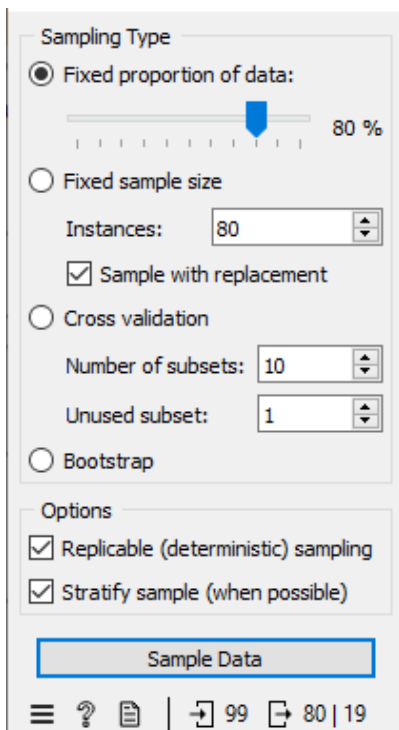
Навчальний датасет D представляє собою матрицю розмірністю 99×7 , де кожен рядок містить вектор ознак та відповідну мітку класу:

$$D = \{(x_1, y_1), (x_2, y_2), \dots, (x_{99}, y_{99})\} \quad (3.4)$$

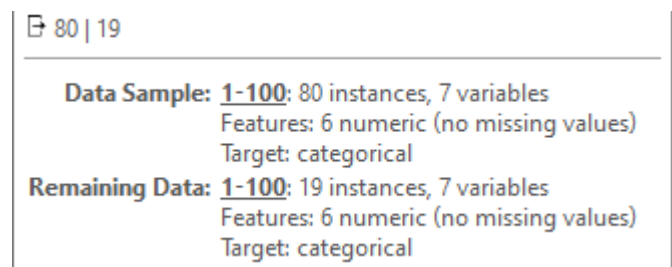
Для проведення експерименту датасет (рис.3.2) розділяється на тренувальний (80% – 80 інстансів) та тестовий (20% – 19 інстансів) набори за принципом унікальності транспортних засобів.

Як показано раніше, після отримання даних їх необхідно обробити за рахунок використання різних алгоритмів. Для дослідження використовуємо декілька для отримання більш точних результатів:

- Random Forest [157],
- Neural Network,
- Logistic Regression.



а)



б)

Рисунок 3.2 – а) Процес розділення даних, б) Результат розділення даних

Прогностичні алгоритми на основі машинного навчання здатні вдосконалити моделі прогнозування, які використовувалися в минулому. Один з цих алгоритмів, випадковий ліс (RF). Цей метод дозволяє покращити управління технічним обслуговуванням і процесом прийняття рішень завдяки введенню класифікаторів для відмов інтегрального типу – несправностей

обладнання, що виникають внаслідок зносу в процесі експлуатації та впливу навантаження на деталі обладнання. Методика використовує дані про технічне обслуговування, які засновані на інформації про кількість циклів функціонування машини, що передують кожному циклу технічного обслуговування [157, 158].

RF – це ансамблевий алгоритм навчання. Дерева будуються незалежно, використовуючи вибірки з набору даних, і для прогнозування береться більшість отриманих оцінок. RF додає до цього методу дерев класифікації елементи випадковості [159, 160]. Ця рандомізація змінює спосіб побудови як дерев для класифікації, так і дерев для регресії. Кожен вузол розбивається на основі найкращих прогнозів з випадково обраної підмножини дерев. Алгоритм витягує з масиву даних вибірки завантаження, будує дерево класифікації або регресії, а потім прогнозує нові дані, об'єднуючи прогнози з розробленого «лісу дерев».

На наступному етапі визначаємось з моделлю штучної нейронної мережі та способом її навчання. Дослідження [161-172] показують, що для рішень задач прогнозування найчастіше застосовують такі архітектури штучних нейронних мереж як MLP та LSTM.

3.1.1. Формалізація цілі класифікації. Ціллю машинного навчання є побудова алгоритму (класифікатора) $f(x, \theta) \rightarrow y$, який відображає простір ознак R^6 у множину бінарних цільових класів $Y = \{0, 1\}$:

- $y = 1$ (позитивний клас) – стан «need repair» (двигун має дефект або потребує превентивного технічного втручання до наступного запланованого ТО).

- $y = 0$ (негативний клас) – стан «no repair» (двигун є повністю справним)

З огляду на екстремальний дисбаланс класів та вимоги до безпеки експлуатації ТЗ, головним критерієм налаштування параметрів моделі θ є

максимізація повноти (Recall) при жорсткому обмеженні на хибно-негативні прогнози ($FN \rightarrow 0$):

$$Recall = \frac{TP}{(TP+FN)} \rightarrow 1.0 \quad (3.5)$$

– Хибно-негативний прогноз (FN) означає, що модель класифікувала несправний двигун як «здоровий». Це призведе до виїзду аварійного автомобіля на маршрут та ризику раптової відмови в дорозі. Хибно-позитивний прогноз (FP – «помилкова тривога») призводить лише до додаткової діагностики на СТО, що економічно менш небезпечно.

– Для інтегральної оцінки якості класифікатора в умовах незбалансованих даних модель максимізує F1-score та коефіцієнт кореляції Метьюса:

$$MCC = \frac{TP*TN-FP*FN}{\sqrt{(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)}} \rightarrow 1.0 \quad (3.6)$$

Безпосереднім результатом роботи аналітичного ядра системи (AnalysisModule) після обробки вхідного вектора ознак є генерація об'єкта класу PredictionResult. Відповідно до розробленої діаграми класів, цей об'єкт має типізовану структуру:

– defectLikelihood: Кількісна оцінка ймовірності виникнення відмови або критичної деградації вузла (наприклад, циліндро-поршневої групи чи кривошипно-шатунного механізму) в діапазоні $[0.0;1.0]$. Цей показник розраховується нейронною мережею MLP на основі виявлених нелінійних аномалій;

– recommendedActions: Динамічно сформований вектор (список) превентивних технічних рекомендацій. Він не є статичним шаблоном, а

генерується індивідуально на основі класифікованого коду несправності (DTC) та ступеня відхилення фізичних параметрів ДВЗ від норми.

Для забезпечення переходу від реактивного ремонту до проактивного предиктивного технічного обслуговування (Predictive Maintenance) в архітектурі ядра AnalysisModule реалізовано двоступеневий конвеєр обробки даних. Перший ступінь вирішує задачу бінарного експрес-скринінгу технічного стану («справний» / «не справний»), а другий ступінь забезпечує локалізацію, багатокласову класифікацію та довгострокове прогнозування конкретного типу механічного зносу.

3.2. Вибір архітектури нейронної мережі для системи прогностичного моніторингу

3.2.1 Багатошаровий перцептрон. Правильна ініціалізація виступає важливою передумовою швидкої збіжності нейронних мереж прямого поширення, таких як перцептрони високого порядку та багатошарові нейронні мережі. Вибір перцептронів є логічним, оскільки вони прості, адаптивні, мають можливість масштабуватись і виступати загальними апроксиматорами функцій.

Багатошаровий перцептрон (БП) є простою моделлю біологічних нейронних мереж і базується на принципі прямого потоку інформації, тобто мережа має ієрархічну структуру. БП складається з різних шарів, де інформація передається тільки від одного шару до іншого.

$$(h, n, \sigma, P), \quad (3.7)$$

де

R – натуральні числа;

$h \in N$ - кількість прихованих шарів;

$n = (n_0, n_1, \dots, n_h) \in \mathbb{N}^{h+1}$ - кількість одиниць у кожному прихованому шарі (нульовий прихований шар є вхідним);

σ - активаційна функція

Шари між вхідним і вихідним шарами є прихованими. З теоретичної точки зору, не обов'язково розглядати більше одного вихідного блоку, оскільки два або більше вихідних блоків можуть бути реалізовані шляхом паралельного розгляду двох або більше БП. Однак, якщо показники результату є корельованими, то можна досягти тих самих результатів апроксимації з меншою кількістю прихованих одиниць [161]:

$$P = (W^1, \dots, W^h, w^y), \quad (3.8)$$

де,

для $i = 1, \dots, h$, $W^i = (w^{i,1}, \dots, w^{i,n_i}) \in (\mathbb{R}^{n_{i-1}+1})^n$ - параметри (ваги та зсуви) між $(i-1)$ -м та i -м прихованими шарами,

$w^y \in \mathbb{R}^{n_h}$ - вектор параметрів між останнім прихованим шаром та вихідним блоком

Швидкість навчання багат шарових перцептронів високого порядку залежить головним чином від початкових значень ваг і зсувів, швидкості навчання, топології мережі та покращення правил навчання, таких як імпульсний член. Оптимальні значення цих параметрів зазвичай невідомі апріорі, оскільки вони залежать головним чином від використаного набору навчальних даних. На практиці неможливо здійснити глобальний пошук оптимальних значень цих параметрів, оскільки поведінка збіжності мережі може значно змінитися при невеликих змінах початкових ваг [162].

В роботі [163] було запропоновано кілька методів ініціалізації ваг для багат шарових перцептронів. Найпростішим з них є ініціалізація випадковими вагами, якій часто надають перевагу через її простоту та здатність давати безліч

розв'язків, оскільки ваги можуть через свою початкову випадковість сходитися до різних атрибутів. Інші методи [164] передбачають великий статистичний та/або геометричний аналіз даних і тому є дуже трудомісткими. Деякі інші методи ініціалізації ваг ґрунтуються на спеціальних властивостях мережі, які не можуть бути застосовані до перцептронів високого порядку або багат шарових перцептронів, як, наприклад, метод ініціалізації ваг для радіальних базисних функцій мережі [165].

Існують спеціальні функції, які можна апроксимувати меншою кількістю прихованих одиниць. В роботах [161, 166] основна увага приділяється глобальній точності апроксимації, але результати мають якісний характер, тобто формули включають невизначені константи і тому не можуть бути використані безпосередньо для обчислення необхідної кількості прихованих одиниць.

Вхідні елементи не відіграють активної ролі в обробці інформаційного потоку, оскільки вони лише розподіляють сигнали між елементами першого прихованого шару. Всі приховані шари працюють однаково, а вихідний елемент є спрощеною версією прихованого шару. У БП кожен прихований елемент перетворює сигнали з попереднього шару в один вихідний сигнал, який розподіляється на наступний шар. Кожен прихований компонент має, як правило, нелінійну функцію активації. Функція активації є результатом перетворення за модулем індивідуального зсуву, однакового для всіх прихованих блоків. Вихід прихованого елемента визначається зваженою сумою сигналів з попереднього шару, яка потім трансформується за рахунок використання функції активації. У вихідному блоці функція активації є функцією тотожності [167].

БП використовуються для наближення (апроксимації) функції. Необхідно точно визначити, що слід розуміти під «наближенням», зокрема, коли одне наближення краще за інше. Однією з можливих мір точності апроксимації може бути найбільша похибка між функцією та її апроксиматором. Відомо, що БП

можуть апроксимувати будь-яку неперервну функцію з як завгодно високою точністю наближення у зазначеному вище сенсі [166], але цього результату важко практично досягти, якщо структура БП є фіксованою. Часто загальна точність є менш важливою, ніж хороше локальне наближення; така точка зору дає поняття «порядок наближення» [167]. Основна ідея розрахунку необхідної кількості прихованих блоків у БП для досягнення заданого порядку апроксимації полягає в тому, щоб, з одного боку, з'ясувати, скільки незалежних величин необхідно скоригувати, щоб досягти заданого порядку апроксимації для довільно достатньо гладкої функції, а з іншого боку, скільки незалежних величин можна скоригувати, змінюючи параметри мережі.

3.2.2 Рекурентна мережа. Рекурентні нейронні мережі (RNN) – це аналог нейронних мереж прямого поширення на основі часових рядів. ШНМ моделюють відображення вхідної послідовності у вихідну послідовність і мають зв'язки зворотного зв'язку у своїх прихованих блоках, що дозволяє їм використовувати інформацію про минулі входи для прогнозування майбутніх виходів. Їх також можна розглядати як особливий вид мережі прямого поширення з «шаром» для кожного часового кроку послідовності. Але на відміну від мережі прямого поширення, де кожен шар має свої власні параметри, «шари» RNN мають спільні параметри [168].

Базові RNN описуються трьома матрицями (і деякими векторами зміщення) та спеціальним початковим вектором прихованого стану, так що $\theta \equiv (W_{xa}, W_{aa}, W_{oa}, b_a, b_o, a_o)$, де W_{xa} - це зв'язки від входів до прихованих блоків, W_{aa} - це рекурентні зв'язки, а W_{oa} - це зв'язки між прихованими блоками та виходами. Нехай задано послідовність векторних входів $x = (x_1, \dots, x_T)$ та векторних цільових виходів $y = (y_1, \dots, y_T)$, RNN обчислює послідовність прихованих станів і виходів $f(\theta, x) = (o_1, \dots, o_T)$ відповідно до:

$$a_t = \phi_a(W_{xa}x_t + W_{aa}a_{t-1} + b_a) \quad (3.9)$$

$$a_t = \phi_o(W_{aa}a_t + b_o) \quad (3.10)$$

де

a_o - спеціальний вектор параметрів початкового стану,

а $\phi_a(-)$ і $\phi_o(-)$ - нелінійні функції активації (оцінюються за координатами).

Цільова функція, що оптимізується при навчанні RNN, є значенням функції втрат, усередненим по навчальній множині S . Вона задається формулою:

$$h(\theta) = \frac{1}{|S|} \sum_{(x,y) \in S} L(y, f(x, \theta)) \quad (3.11)$$

Функція втрат $L(y, z)$ зазвичай визначається як сума специфічних для кожного кроку втрат $L_t(y_t, z_t)$, які вимірюють ступінь, до якого z_t не може передбачити y_t . Вона задається формулою:

$$L(y, z) \equiv \sum_{t=1}^T L_t(y_t, z_t) \quad (3.12)$$

Рекурентна нейронна мережа отримує послідовні входи $u_p(t)$ і виробляє вектор $(\phi_1(t), \phi_2(t), \dots, \phi_n(t))$ на кожному кроці t . Ці значення лінійно комбінуються з ваговою матрицею W , щоб отримати вихідний вектор мережі $y_m(t)$. Під час навчання RNN ваги вихідного шару обчислюються за рахунок використання лінійної регресії або квадратичного програмування.

Вихід мережі в момент часу t , $y(t) \in R^m$, обчислюється за наступними формулами:

$$y(t) = W_\phi(t) \quad (3.13)$$

$$\phi(t) = f(u(y), u(t-1), \dots, u(0)) \quad (3.14)$$

де

$\phi(t) \in R^n$ - вихід рекурентної нейронної мережі $f(-)$,

W - вагова матриця.

У той час як приховані моделі, як правило, ігнорують цю інформацію, рекурентні нейронні мережі (RNN) (рис.3.3), в принципі, можуть її використовувати. Тимчасова відстань між подіями передає інформацію, необхідну для численних послідовних завдань, таких як керування двигуном і виявлення ритму. Якщо зосередитись на довготривалій короткочасній пам'яті (LSTM), оскільки було доведено у роботі [170], що вона перевершує інші RNN у завданнях з великими часовими затримками, можна виявити, що LSTM може навчитися тонко розрізняти послідовності піків без допомоги будь-яких коротких навчальних прикладів. Без зовнішніх скидань або примусу з боку вчителя наш варіант LSTM також навчається генерувати стабільні потоки точно синхронізованих піків та інших високо нелінійних періодичних патернів. Це робить LSTM перспективним підходом для задач, які вимагають точного вимірювання [170].

Як варіант, побудувати мережу на основі LSTM (рис. 3.4) із затворами, що забуваються [171]. LSTM – це рекурентна нейронна мережа спеціально розроблена для навчання довготривалих залежностей за рахунок використання градієнтного спуску. Унікальною особливістю архітектури LSTM є комірка пам'яті, яка здатна зберігати свою активацію нескінченно довго. Базовою одиницею мережі LSTM є блок пам'яті, що містить одну або більше комірок пам'яті і три адаптивні мультиплікативні елементи, які використовуються всіма комірками блоку. Кожна комірка пам'яті має в основі рекурентно самопідключений лінійний блок, який ми називаємо «Карусель постійних помилок» (Constant Error Carousel - CEC). Рециркулюючи сигнали активації та помилки нескінченно, CEC забезпечує короткочасне зберігання пам'яті протягом тривалих періодів часу. Вхідні, вихідні та забуваючі вентиля можуть

бути навчені, щоб дізнатися, відповідно, яку інформацію зберігати в пам'яті, як довго її зберігати і коли її зчитувати. Об'єднання комірок пам'яті в блоки дозволяє їм використовувати одні й ті ж вентиля (за умови, що це дозволяє задача), таким чином зменшуючи кількість адаптивних параметрів [170].

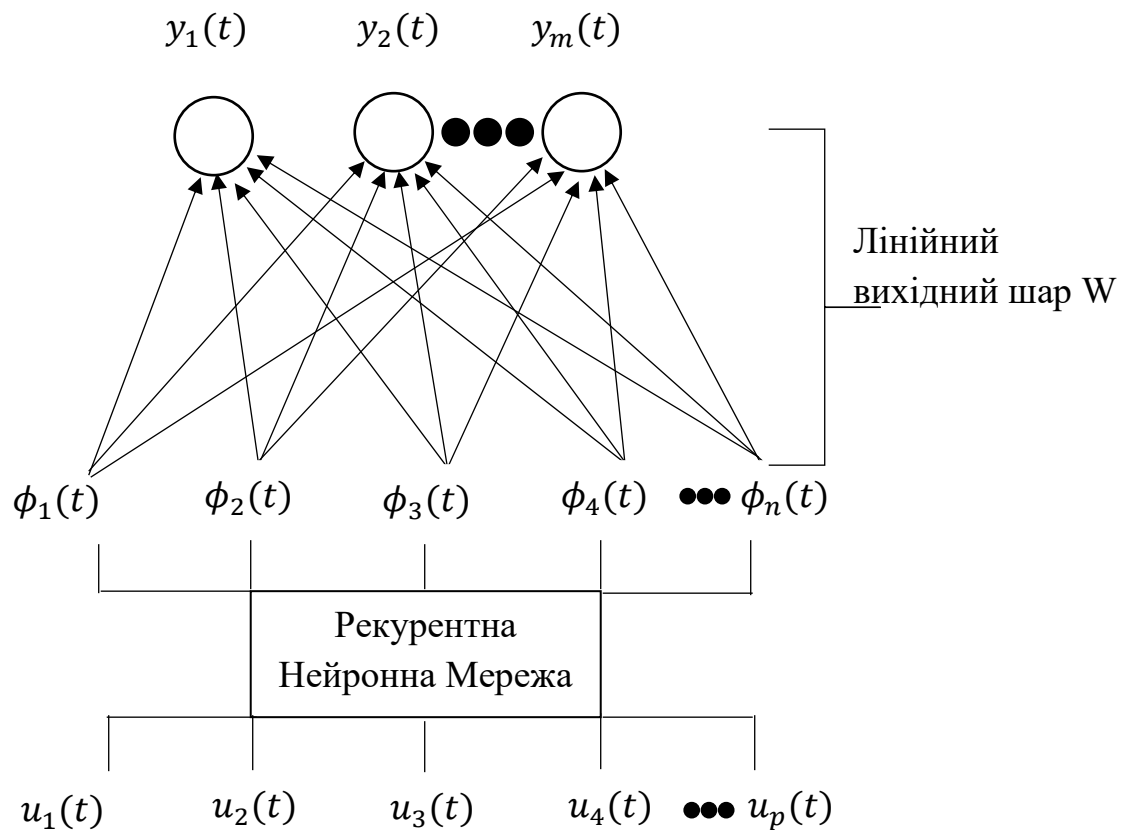


Рисунок 3.3 – Схема рекурентної нейронної мережі [169].

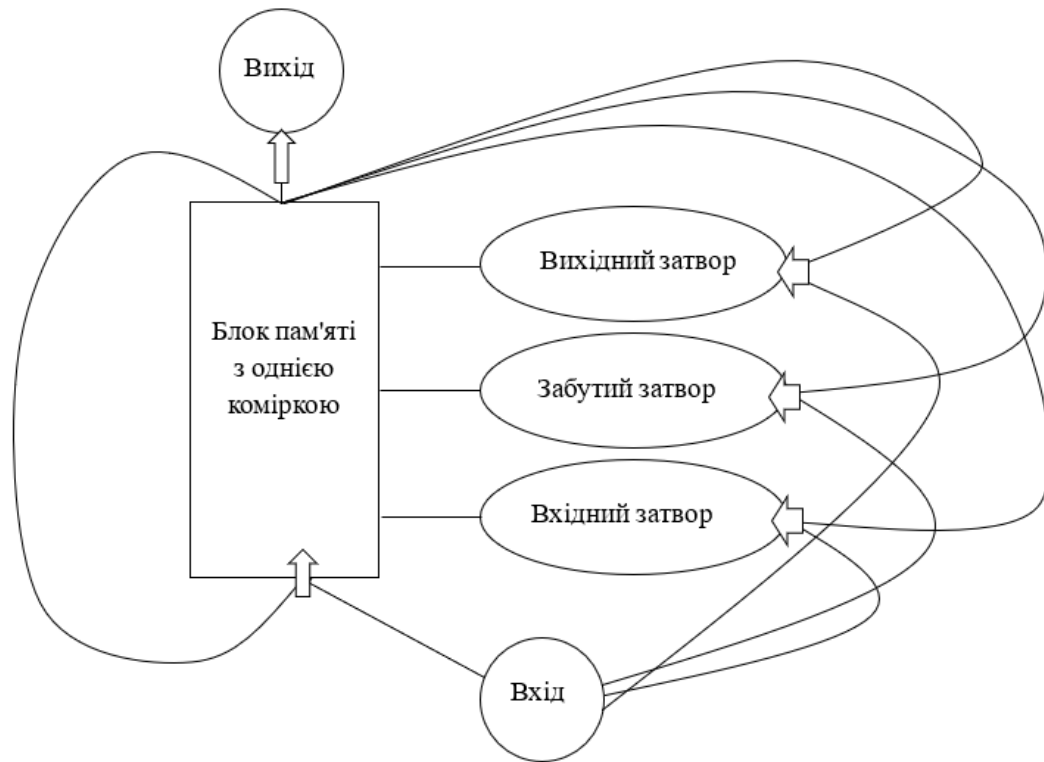


Рисунок 3.4 – Тришарова топологія LSTM

На рис. 3.5 показано комірку пам'яті LSTM. Комірка має внутрішній стан S разом із затвором забування (G_F), який визначає, наскільки цей стан послаблюється на кожному часовому кроці. Вхідний клапан (G_i) контролює доступ до комірки через зовнішні входи, які підсумовуються в одиниці Σ , а вихідний клапан (G_O) контролює, коли і наскільки комірка спрацьовує. Маленькі темні вузли представляють функцію множення.

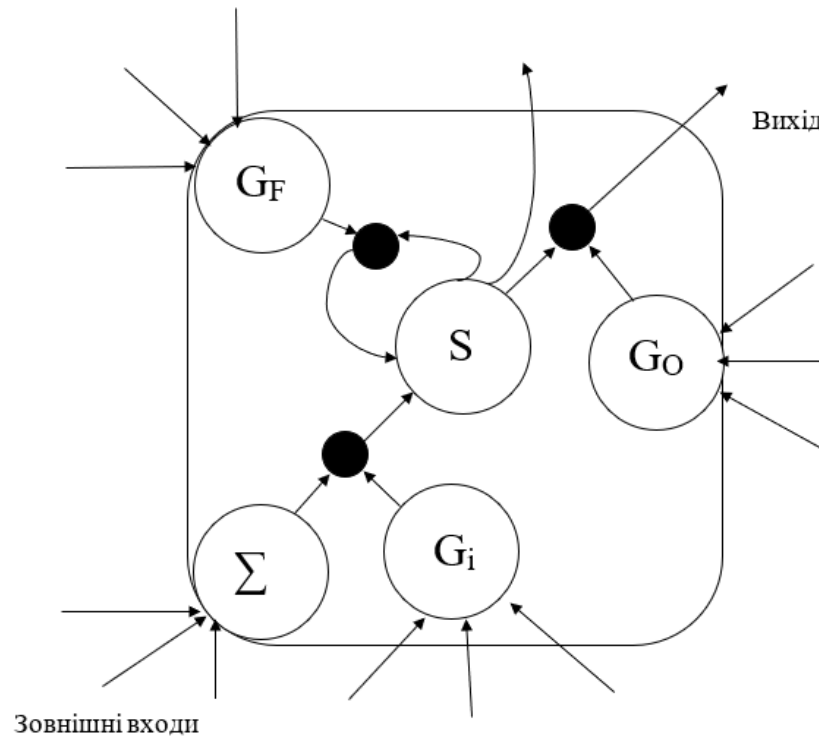


Рисунок 3.5 – LSTM мережа[169].

Стан комірки i обчислюється за формою:

$$s_i(t) = net_i(t)g_i^{in}(t) + g_i^{forget}(t)s_i(t-1) \quad (3.15)$$

де

g_i^{in} та g_i^{forget} - активація вхідного та забутого вентилів відповідно,

net_i - зважена сума зовнішніх входів (позначена Σ на рис. 3.5)

$$net_i(t) = h\left(\sum_j w_{ij}^{cell} c_j(t-1) + \sum_k w_{ik}^{cell} u_k(t)\right) \quad (3.16)$$

Де

h - функція тотожності,

c_j - вихід комірки j :

$$c_j(t) = \tanh\left(g_j^{out}(t)s_j(t)\right) \quad (3.17)$$

де g_j^{out} - вихідний вентиль комірки j . Величина, на яку кожен вентиль g_i пам'яті комірки i відкрита або закрита у момент часу t , обчислюється на величину:

$$g_i^{type}(t) = \sigma\left(\sum_j w_{ij}^{type} c_j(t-1) + \sum_k w_{ik}^{type} u_k(t)\right) \quad (3.18)$$

де

$type$ може бути *in*, *out* або *forget*,

σ - стандартна сигмоїдна функція. Вхідні дані на вентилі надходять з виходів інших комірок c_j та з зовнішніх входів мережі.

По результату огляду даних архітектур нейронних мереж можна зробити наступні висновки:

- застосування MLP є доцільним у задачах прогнозування відмов, де взаємозв'язок між діагностичними параметрами має нелінійний характер, а кількість ознак може змінюватися залежно від комплектації транспортного засобу. Це забезпечує високу адаптивність моделі та можливість її використання для транспортних засобів різних виробників і модифікацій;

- рекурентні нейронні мережі типу LSTM доцільно використовувати коли телеметричні дані транспортного засобу мають яскраво виражений часовий характер даних. На відміну від традиційних моделей класифікації, LSTM здатна враховувати довгострокові часові залежності між послідовними вимірюваннями параметрів двигуна, трансмісії та інших систем автомобіля. Це дозволяє своєчасно виявляти тенденції деградації технічного стану та підвищує точність прогнозування можливих відмов.

3.2.3. Підбір оптимізатора для нейронної мережі. Глибокі нейронні мережі можна оптимізувати різними способами, включаючи оптимізацію структурного

дизайну мережевої моделі та визначення оптимальних параметрів, таких як ваги та зміщення попередньо визначеної структури мережі, попередню обробку наборів даних та вибір найкращого методу оптимізації під час процесу навчання. Наразі не існує встановлених критеріїв для розробки ідеальної архітектури глибоких нейронних мереж [172]. Метою будь-якого оптимізатора є мінімізація цільової функції, також відомої як функція втрат, яка є різницею між очікуваним і розрахованим значеннями. Процедура мінімізації визначає найкращий набір параметрів для проектування штучної нейронної мережі (ШНМ) для задач класифікації, прогнозування та кластеризації.

В роботах [173, 174] представлено ряд алгоритмів оптимізації для глибоких нейронних мереж. Більшість з цих алгоритмів використовують метод градієнтного спуску, який є диференціальним методом першого порядку, що використовується для отримання масиву ваг, які задовольняють критерії похибки. Найпоширенішими алгоритмами оптимізації глибоких нейронних мереж в літературі [172, 175, 176] є методи градієнтного спуску, такі як алгоритми зворотного розповсюдження та адаптивної оцінки моменту (Adaptive Moment Estimation – ADAM).

Окрім цього, стохастичний градієнтний спуск (SGD), запропонований ще в минулому столітті [177], залишається одним з найефективніших алгоритмів навчання глибоких нейронних мереж. Останнім часом [178] стали популярними методи адаптивної оптимізації, які регулюють швидкість навчання параметрів у різних масштабах замість того, щоб безпосередньо контролювати загальний розмір кроку, що призводить до більш плавного процесу навчання і швидшої збіжності.

Оскільки налаштувати швидкість навчання алгоритму SGD як гіперпараметр складно, її коригують протягом усього процесу навчання [175]. Адаптивні варіанти алгоритмів SGD намагаються автоматично адаптувати швидкість навчання для оновлення параметрів на основі статистики градієнта.

Хоча ці адаптивні варіанти спрощують налаштування швидкості навчання і збільшують швидкість збіжності, в деяких додатках [176] їх загальна продуктивність значно гірша, ніж у базового алгоритму SGD. Як наслідок, алгоритм SGD (можливо, з імпульсом) все ще використовується для навчання передових глибоких нейронних моделей, таких як ШНМ, що працюють у режимі прямого поширення (feed-forward DNN) [176].

Навчання нейронних мереж – це процес визначення ваги зв'язків між нейронами для досягнення необхідних співвідношень між входами та виходами з певною точністю. На оновлені значення ваги впливає значення швидкості навчання. Існує два типи методів навчання, що використовуються для навчання нейронних мереж: кероване та некероване навчання. У більшості практичних застосувань машинного навчання, де мережева модель має навчальний набір вхідних і вихідних даних, використовується кероване навчання. Цей тип навчання використовується для забезпечення наближення функції відображення для представлення зв'язку між входами і виходами.

Класифікація та регресія – дві поширені проблеми, які вирішуються за рахунок використання керованого навчання. Некероване навчання використовується, коли мережева модель має лише вхідні дані і не має відповідних виходів, як, наприклад, у задачах кластеризації. Мета цього типу навчання полягає в тому, щоб дізнатися більше про дані, моделюючи основну структуру або розподіл

Базове навчання нейронних мереж еквівалентне розв'язанню наступної оптимізаційної задачі:

$$\min_{w \in R^n} f(w) \quad (3.19)$$

де

f – функція втрат. Ітерації SGD можна описати наступним чином:

$$w_k = w_{k-1} - a_{k-1} \nabla f(w_{k-1}) \quad (3.20)$$

де

w_k позначає ітерацію k^{th} ,

a_k – (налаштована) послідовність розміру кроку, яку також називають швидкістю навчання,

$\nabla f(w_{k-1})$ позначає стохастичний градієнт, обчислений на w_k .

А математичне рівняння оновлення Adam можна представити у вигляді:

$$w_k = w_{k-1} - a_{k-1} \frac{\sqrt{1-\beta_2^k} m_{k-1}}{\sqrt{1-\beta_1^k} \sqrt{v_{k-1} + \epsilon}} \quad (3.21)$$

де

$$m_{k-1} = \beta_1 m_{k-2} + (1 - \beta_1) \nabla f(w_{k-1}) \quad (3.22)$$

$$v_{k-1} = \beta_2 v_{k-2} + (1 - \beta_2) \nabla f(w_{k-1})^2 \quad (3.23)$$

Відповідно β_1, β_2 є двома гіперпараметрами. Перший контролює імпульс першого порядку, а другий - імпульс другого порядку.

Основним алгоритмом навчання, який використовується для навчання ШНМ з навчанням під контролем, є зворотне поширення, яке має дві робочі фази: пряме поширення та зворотне поширення [179, 180]. Він базується на алгоритмі градієнтного спуску (GD), який обчислює градієнти по всьому набору даних, так як легко поміщається в пам'яті і є швидким в обчисленнях. Це призводить до великої кількості ітерацій і збільшує ризик потрапити в пастку локальних оптимумів з ранньою збіжністю. У зв'язку з цими проблемами в роботі [180] було запропоновано метод міні-пакетного градієнтного спуску. Навчальний набір даних розбивається на партії фіксованого розміру для використання в процесі навчання в цьому методі. Обчислюється загальна

похибка, і ваги для кожної підпачки оновлюються. Коли значення міні-партії дорівнює «1», використовується алгоритм стохастичного градієнтного спуску. У цьому випадку похибка обчислюється для однієї вибірки за раз, і ваги також оновлюються, що призводить до швидшої збіжності за рахунок прямого векторизації даних. Значення помилки потім поширюється назад через мережі, а ваги оновлюються за рахунок використання GD у зворотному напрямку кривизни. Параметри мережі, що підлягають оптимізації, оновлюються відповідно як:

$$\theta_{t+1} = \theta_t - \eta \nabla_{\theta} J(\theta_t) \quad (3.24)$$

де

η – швидкість навчання,

$\nabla_{\theta} J(\theta_t)$ – градієнт функції втрат $J(\theta_t)$ відносно θ_t .

Алгоритм стохастичного градієнтного спуску (SGD) обчислює втрачену функцію для однієї навчальної вибірки за раз, замість того, щоб розглядати всі навчальні вибірки даних. Таким чином можна уникнути проблем дефіциту пам'яті. SGD був створений для усунення недоліків алгоритму пакетного градієнтного спуску. Проблема використання SGD полягає у визначенні правильного значення швидкості навчання, щоб уникнути коливань і досягти глобального оптимуму. Для оновлення параметрів використовується наступне рівняння [179]:

$$\theta_{t+1} = \theta_t - \eta \nabla_{\theta} J(\theta_t; x^{(i)}; y^{(i)}) \quad (3.25)$$

де

η – швидкість навчання,

$\nabla_{\theta} J(\theta_t)$ - градієнт функції втрат $J(\theta_t)$ відносно θ_t . Також $x^{(i)}$ та $y^{(i)}$ представляють навчальні дані у вигляді пар входів-виходів. Якщо крива функції втрат має перегини, де один вимір має нахил вгору, а інший - вниз, алгоритм SGD працює погано.

Інший варіант – градієнтний спуск з імпульсом. Крок навчання в методах градієнтного спуску бажано збільшувати, щоб швидше досягти найкращого результату. Коли кроки навчання дуже великі, глобального оптимуму не завжди можна досягти. Ці великі кроки можуть мати прямий вплив на час, необхідний для досягнення глобального оптимуму. Для вирішення цих проблем в роботі [174] було запропоновано метод градієнтного спуску за імпульсом. Він обмежує швидкість наступного кроку навчання, використовуючи середню швидкість попередніх кроків навчання. У цьому методі динамічне середнє минулих градієнтів (m_{t-1}) експоненціально зменшується і зберігається. Напрямок руху визначається з урахуванням цих динамічних усереднень. Таким чином, кроки процесу навчання рухаються швидше до найкращого результату з меншим відхиленням [181]. Правило оновлення може бути виражене у вигляді:

$$m_t = \gamma m_{t-1} + \eta \nabla_{\theta} J(\theta_t) \quad (3.26)$$

$$\theta_{t+1} = \theta_t - m_t \quad (3.27)$$

де

γ - параметр імпульсу, який зазвичай має значення 0.9 або близьке до нього.

Ще одним варіантом навчання є алгоритм адаптивного градієнта (AdaGrad) ділить компонент швидкості навчання на квадратний корінь з v_t , який є сумою поточного і минулого квадратів градієнта до моменту часу t . Компонент градієнта, як і в SGD, залишається незмінним. AdaGrad робить різні оновлення для кожного параметра, використовуючи різні швидкості навчання

для кожного кроку. Найважливішою перевагою використання AdaGrad є те, що швидкість навчання не регулюється вручну, як в інших системах адаптивного навчання [182]. Рівняння оновлення AdaGrad можуть бути виражені як:

$$g_t = \nabla_{\theta} J(\theta_t) \quad (3.28)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{v_t + \epsilon}} g_t \quad (3.29)$$

де

ϵ – згладжуючий коефіцієнт, який дозволяє уникнути ділення на нуль (зазвичай встановлюється рівним 10^{-8}),

v_t – експоненціальна змінна середнього градієнта.

В роботі [182] обговорюють фундаментальний недолік сучасних методів експоненціальної змінної середнього градієнта., таких як ADAM. Вони показують, що ADAM може не збігатися до оптимального розв'язку навіть у простих одновимірних опуклих налаштуваннях і, як вказано в роботі, основна проблема полягає в наступному:

$$G_{t+1} = \left(\frac{\sqrt{V_{t+1}}}{a_{t+1}} - \frac{\sqrt{V_t}}{a_t} \right) \quad (3.30)$$

Ця величина по суті вимірює зміну величини, оберненої до швидкості навчання адаптивного методу відносно часу. Одне з ключових спостережень полягає в тому, що для SGD та AdaGrad $G_t \geq 0$ для всіх $t \in [T]$. Зокрема, правила оновлення для цих алгоритмів призводять до «незростаючої» швидкості навчання. Однак це не обов'язково стосується експоненціальних варіантів ковзного середнього, таких як ADAM, де G_t потенційно може бути невизначеним для $t \in [T]$ [183]. Це порушення позитивної визначеності може призвести до небажаної поведінки збіжності для ADAM.

$$f_t(x) = \begin{cases} Cx, & \text{for } t \bmod 3 = 1 \\ -x, & \text{otherwise} \end{cases} \quad (3.31)$$

де

$C > 2$. Для цієї послідовності функцій можна побачити, що точка $x = -1$ забезпечує мінімум функції.

Припустимо, що гіперпараметри $\beta_1 = 0$ і $\beta_2 = \frac{1}{1+C^2}$. Можна побачити, що ADAM сходиться до дуже субоптимального розв'язку $x = 1$ для цього параметра. Алгоритм отримує великий градієнт C кожні 3 кроки, а решту 2 кроки він спостерігає градієнт -1 , який рухає алгоритм у неправильному напрямку. Великий градієнт C не може протидіяти цьому ефекту, оскільки для даного значення β_2 він зменшується майже в C разів, а отже, алгоритм сходиться до 1, а не до -1 [183].

Таким чином, аналіз сучасних алгоритмів оптимізації показав, що універсального оптимізатора для всіх задач машинного навчання не існує. Алгоритм SGD характеризується високою узагальнюючою здатністю, однак потребує ретельного налаштування швидкості навчання. Алгоритми AdaGrad та ADAM забезпечують швидшу збіжність на початкових етапах навчання, автоматично адаптуючи швидкість оновлення параметрів. Для задач прогнозування технічного стану транспортних засобів найбільш доцільним є використання алгоритму ADAM під час початкового навчання моделі завдяки його швидкій збіжності, тоді як SGD із моментом може бути використаний на завершальних етапах оптимізації для покращення узагальнюючої здатності моделі.

3.2.4. Визначення оптимальної структури автокодера (Autoencoder).

Для стиснення розмірності вхідного простору ознак, усунення шумів та виявлення аномалій (Unsupervised Anomaly Detection) у препроцесорі системи застосовано нейромережу-автокодер.

Задача оптимізації структури автокодера полягала у виборі симетричної конфігурації повнозв'язних шарів та активаційної функції, що мінімізують середньоквадратичну помилку реконструкції (*MSE Loss*) вхідного вектора $\mathbf{x} \in \mathbb{R}^6$:

$$MSE = \frac{1}{M} \sum_{i=1}^M ||x_i - \hat{x}_i||^2 \rightarrow \min \quad (3.32)$$

Дослідження різних структурних конфігурацій за допомогою сіткового пошуку наведено в таблиці 3.1.

Таблиця 3.1 – Аналіз структурних конфігурацій за допомогою сіткового пошуку

Конфігурація шарів (Encoder → Bottleneck → Decoder)	Функція активації прихованих шарів	Оптимальна швидкість навчання (η)	Отримана похибка реконструкції (<i>MSE</i> <i>Loss</i>)
[6 -> 4 -> 2 -> 4 -> 6]	ReLU	0.010	447.5157
[6 -> 4 -> 3 -> 4 -> 6]	ReLU	0.010	405.8021 (Оптимальна)
[6 -> 3 -> 1 -> 3 -> 6]	ReLU	0.010	767.5919
[6 -> 5 -> 3 -> 5 -> 6]	Tanh	0.005	1 415 400.7500 (Не збіглася)

Експериментально доведено, що архітектура [6 -> 4 -> 3 -> 4 -> 6] з нелінійною активацією ReLU забезпечує мінімальну похибку реконструкції (MSE=405.80) в процесі навчання протягом 200 епох.

- зменшення розмірності прихованого простору (bottleneck) до двох або одного нейрона ([6->4->2->4->6] та [6->3->1->3->6]) веде до інформаційного «голодування» моделі, коли корисні ознаки затираються, збільшуючи mse до 447.5 та 767.5 відповідно;

- активаційна функція tanh при симетричній структурі [6->5->3->5->6] продемонструвала незадовільну збіжність через ефект загасання градієнтів на ненормованих фізичних величинах.

3.2.5. Визначення оптимальної структури рекурентної мережі LSTM.

Для прогнозування залишкового ресурсу вузлів ДВЗ та виявлення часових трендів деградації на безперервних потоках телеметрії (Time-Series) досліджено рекурентні нейронні мережі LSTM.

Пошук оптимальних гіперпараметрів LSTM здійснювався за сіткою, яка включала розмірність прихованого стану ($\text{hidden_dim} \in \{16, 32, 64\}$), кількість рекурентних шарів ($\text{num_layers} \in \{1, 2\}$) та параметр спрямованості ($\text{bidirectional} \in \{\text{True}, \text{False}\}$). Критерієм оптимізації виступала крос-ентропійна втрата (BCE Loss):

$$BCE = -\frac{1}{B} \sum_{i=1}^B [y_i \log(\widehat{y_i}) + (1 - y_i) \log(1 - \widehat{y_i})] \rightarrow \min \quad (3.33)$$

Результати сіткового пошуку наведено в таблиці 3.2.

Таблиця 3.2 – Аналіз архітектур моделі за допомогою сіткового пошуку

Архітектура рекурентної моделі	Розмірність стану (<i>hidden_dim</i>)	Кількість шарів (<i>layers</i>)	Двонаправленість (<i>bidirectional</i>)	Отримана похибка (<i>BCELoss</i>)
Bi-LSTM	16	2	True	0.001184 (Оптимальна)
LSTM	16	1	False	0.001264
LSTM	32	1	False	0.001290
LSTM	32	2	False	0.001328

Найвищу точність прогнозування та найменшу похибку навчання ($BCE=0.001184$) продемонструвала двонаправлена рекурентна мережа (Bi-LSTM) з 16 прихованими одиницями та 2 рекурентними шарами.

- двонаправленість моделі ($bidirectional = true$) дозволяє системі аналізувати часові послідовності зміни тиску та температур як у прямому хронологічному порядку, так і у зворотному. це критично важливо для виявлення передумов відмови (наприклад, оцінки швидкості зміни температурного градієнта перед заклинюванням);

- збільшення розмірності прихованого стану до 32 або 64 за умов обмеженої довжини послідовностей (вікно довжиною 10 кроків) веде до надмірної параметризації моделі, уповільнює збіжність та збільшує bce loss (до 0.001328), що доводить раціональність використання більш компактної bi-lstm архітектури для розгортання на периферійному бортовому обладнанні.

3.3. Порівняний аналіз різних моделей машинного навчання

Після аналізу можливих варіантів нейронних мереж побудовано схематичну модель для аналізу та прогнозування.

Для виявлення найбільш якісного результату було проведено порівняння декількох моделей машинного навчання на однаковому наборі даних:

- логістична регресія використовується як базова модель класифікації для порівняльного аналізу ефективності більш складних алгоритмів машинного навчання. незважаючи на обмежені можливості моделювання нелінійних залежностей, вона забезпечує інтерпретованість результатів та дозволяє оцінити внесок окремих ознак у формування прогнозу;

- алгоритм random forest використано як базову ансамблевую модель завдяки його високій стійкості до шумових даних, можливості роботи з великою кількістю ознак та низькій схильності до перенавчання. крім того, алгоритм забезпечує оцінювання важливості окремих параметрів, що дозволяє визначити найбільш інформативні діагностичні показники для прогнозування технічного стану транспортних засобів.

З рис. 3.6 видно, що до блоку отриманих результатів («test and score») від даних направлено два набори (тестовий та тренувальні). Окрім цього, навчання відбувається за рахунок використання алгоритмів описаних вище. Після чого, результати потрапляють на блок «predictions», який відповідає за прогнозування системи для обробки тестових даних. На рис. 3.7 розглянемо основні параметри алгоритмів при побудові прогностичної моделі:

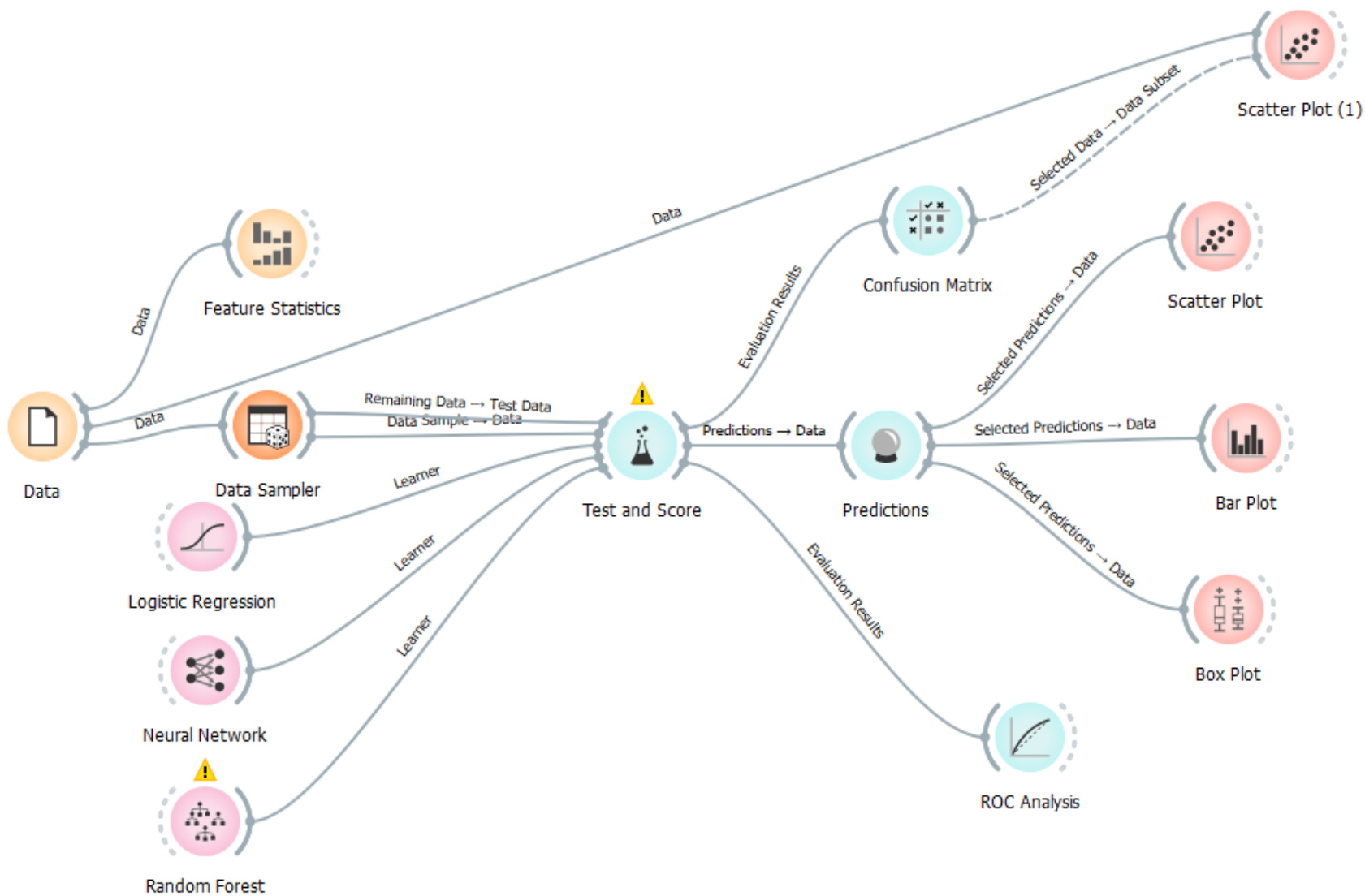


Рисунок 3.6 – Загальний вигляд побудованої системи

а)

б)

в)

Рисунок 3.7 – а) Параметри логістичної регресії, б) Параметри нейронної мережі, в) Параметри для випадкового лісу

Після вибору параметрів роботи отримуємо записані в блоки результати по тренувальним даним та тестовим даним (рис. 3.8):

Model	AUC	CA	F1	Prec	Recall	MCC
Neural Network	0.967	0.938	0.936	0.943	0.938	0.866
Random Forest	0.983	0.950	0.950	0.950	0.950	0.890
Logistic Regression	0.741	0.650	0.627	0.626	0.650	0.169

а)

Model	AUC	CA	F1	Prec	Recall	MCC
Neural Network	0.702	0.632	0.561	0.593	0.632	0.094
Random Forest	0.643	0.737	0.732	0.731	0.737	0.420
Logistic Regression	0.702	0.684	0.644	0.680	0.684	0.268

б)

Рисунок 3.8 – Блок отриманих результатів роботи нейронних мереж, де а) для тренувальних даних, б) для тестових даних

Основними показниками аналізу виступають AUC, CA, F1, Prec, Recall, MCC. Розглянемо кожен з них.

AUC – area under the ROC curve (AUC) – Площа під кривою ROC (AUC) – це глобальна міра здатності тесту розрізнити наявність або відсутність певного стану. AUC 0,5 являє собою тест без дискримінаційної здатності (тобто не кращий за випадковість), тоді як AUC 1,0 являє собою тест із ідеальною дискримінацією. При виборі найкращого граничного значення (або точки обмеження) необхідно враховувати цілі діагностичного тесту, зважаючи на важливість та вартість хибнопозитивної або хибнонегативної оцінки. Загальноприйнятим підходом при виборі точки обмеження є надання рівної ваги важливості чутливості та специфічності шляхом вибору точки, найближчої до лівого верхнього кута ROC-кривої (рис. 3.9, 3.10) [184].

CA – classification accuracy – значення точності, тобто найбільша точність, яку, як очікується, має мати певний показник, враховуючи результати тесту та

обраний ризик для користувача та вказує не лише на те, чи пройшли дані тест, але й наскільки добре вони його завершили [185].

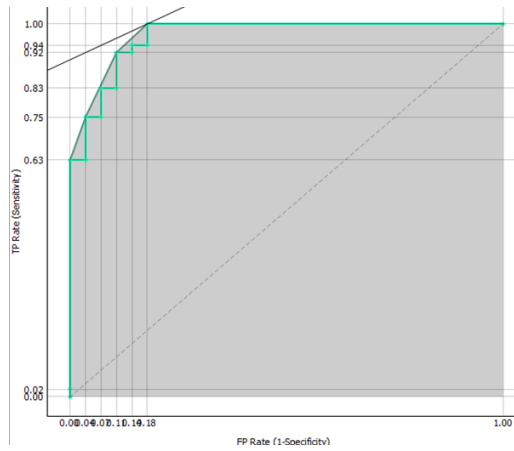
$$CA = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (3.34)$$

F1 – F-score – альтернативна метрика оцінювання машинного навчання, яка оцінює вміння моделі прогнозувати, аналізуючи її продуктивність у розрізі окремих класів, а не загальну продуктивність, як це робиться за рахунок використання показника точності. визначається як гармонійне середнє значення точності та запам'ятовування [186, 187].

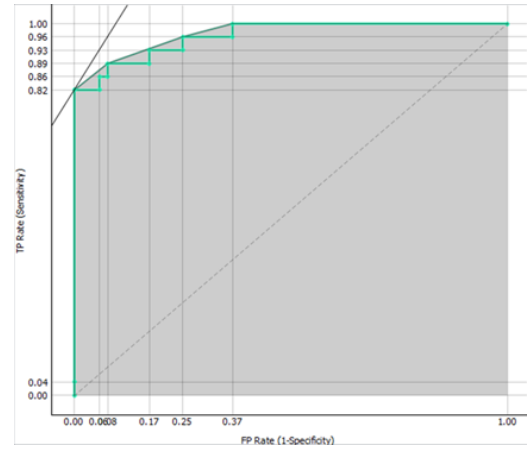
$$F1score = (1 + \beta^2) \frac{precision*recall}{\beta^2*precision+recall} \quad (3.35)$$

де β – вага *precision*. Зазвичай закладають $\beta = 1$.

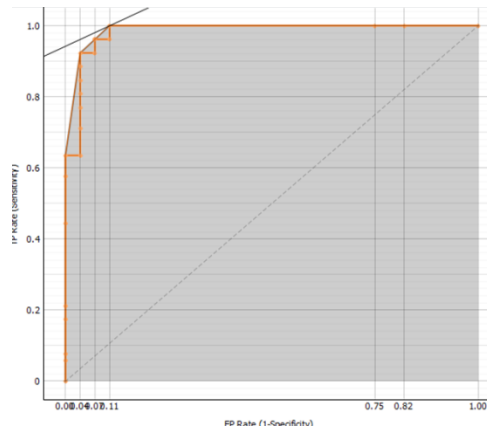
ROC-криві підтверджують високу якість класифікації для всіх досліджуваних моделей. Найбільші значення площі під ROC-кривою спостерігаються для нейронної мережі та Random Forest, що свідчить про їхню кращу здатність відокремлювати справний технічний стан від потенційних несправностей. Логістична регресія також демонструє прийнятні результати, однак поступається нелінійним моделям при аналізі складних залежностей між параметрами.



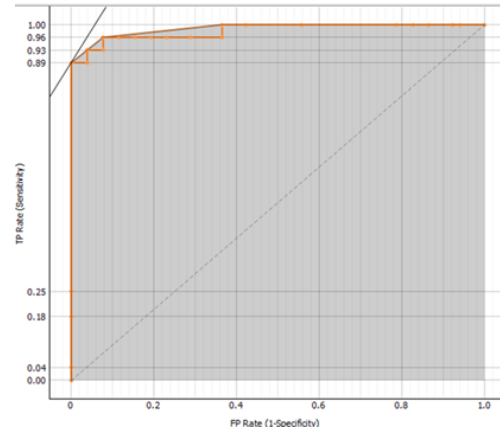
а)



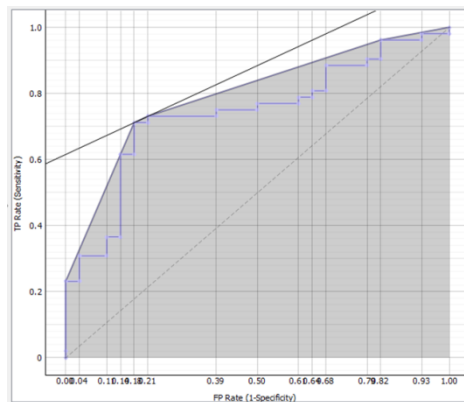
б)



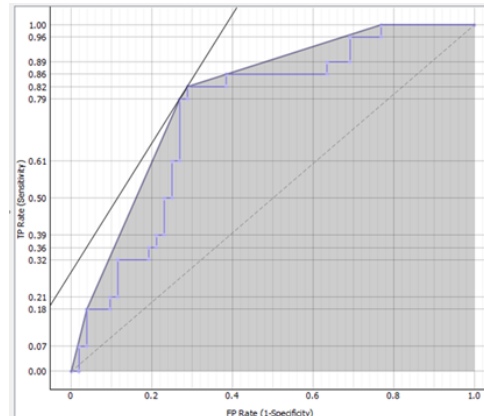
в)



г)

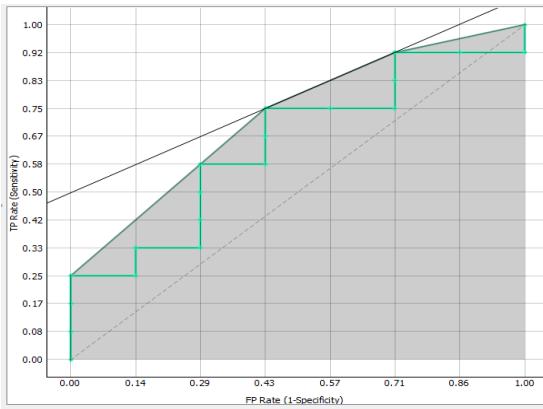


д)

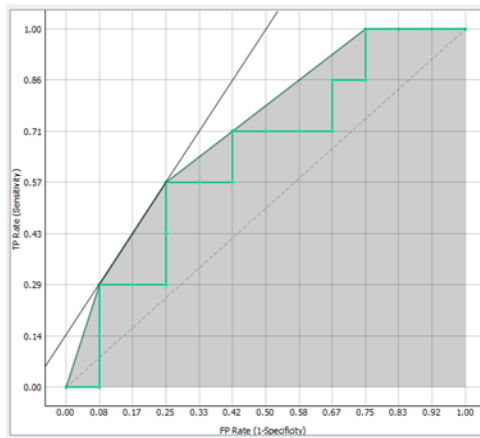


е)

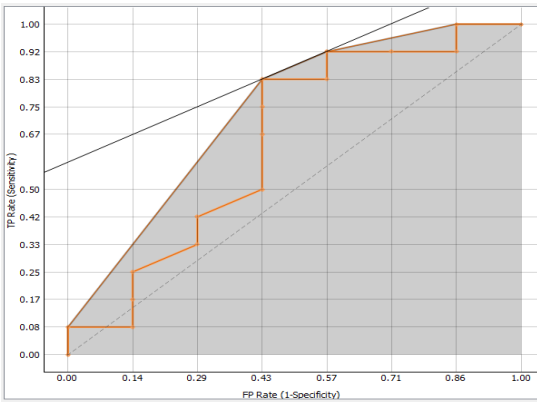
Рисунок 3.9 – ROC-криві для алгоритмів тренувальної вибірки цільових параметрів (need repair та no repair)– а та б) нейронна мережа, в та г) випадковий ліс, д та е) логістична регресія



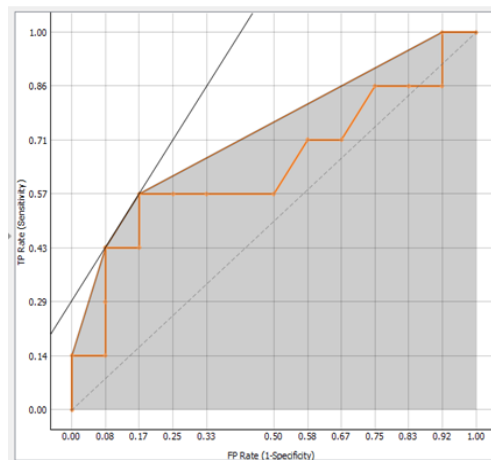
а)



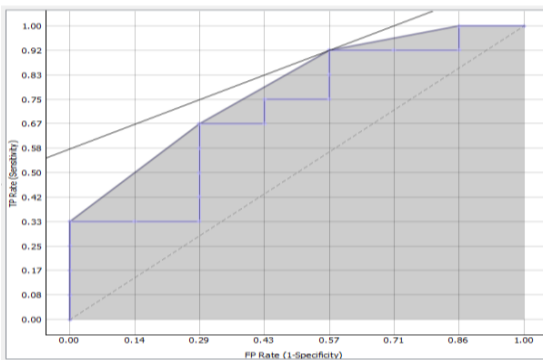
б)



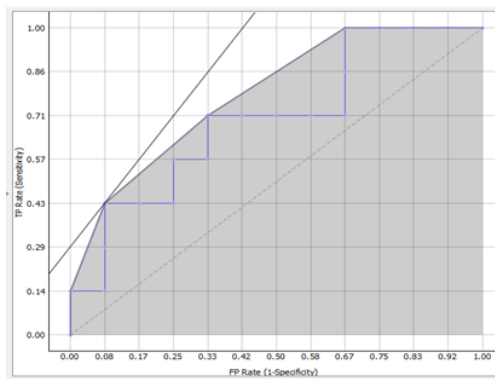
в)



г)



д)



е)

Рисунок 3.10 – ROC-криві для алгоритмів тестувальної вибірки цільових параметрів (need repair та no repair) – а та б) нейронна мережа, в та г) випадковий ліс, д та е) логістична регресія

Prec – precision – вимірює, наскільки точно модель визначає позитивний вихід для заданих вхідних даних. Високе значення точності свідчить про те, що модель нейронної мережі уникає помилкових спрацьовувань [188]:

$$precision = \frac{TP}{(TP+FP)} \quad (3.36)$$

Recall – здатність нейромережевої моделі виявляти позитивні мітки класів. Модель з високим значенням відтворення точно фіксує дані, пов'язані з позитивними мітками класів, і уникає хибно-негативних прогнозів. Моделі з високими показниками відтворення є необхідними, коли витрати на усунення хибно-негативних результатів є значними [189]:

$$Recall = \frac{TP}{(TP+FN)} \quad (3.37)$$

MCC – Matthew's correlation coefficient – це найкращий показник класифікації з одним значенням, який допомагає узагальнити матрицю плутанини або матрицю помилок. Матриця плутанини – це матриця, яка узагальнює прогнози нашої моделі щодо фактичних значень класів у наборі даних [187]:

$$MCC = \frac{TP*TN-FP*FN}{\sqrt{(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)}} \quad (3.38)$$

де:

- істинно позитивні (tp): кількість позитивних прикладів, які належать до позитивного класу і прогнозуються як позитивні:
- істинно негативні (tn): кількість прикладів, які належать до негативного класу і прогнозуються як негативні:
- хибно-позитивний (fp): кількість прикладів, які належать до негативного

класу, але передбачені моделлю як позитивні:

- хибно-негативні (FN): кількість прикладів, які належать до позитивного класу, але прогножуються моделлю як негативні.

Для об'єктивної оцінки здатності алгоритмів до узагальнення на нових масивах даних автомобільного сервісу, тестування моделей було проведено на тестовій (валідаційній) вибірці. Базовою моделлю для порівняння обрано логістичну регресію. Результати оцінювання наведено на рис. 3.8 б).

На основі отриманих метрик можна виділити наступні ключові показники ефективності:

1. Мінімізація хибно-негативних прогнозів (False Negatives) та збільшення повноти (Recall):

Застосування ансамблевого методу Random Forest дозволило зменшити кількість пропущених критичних дефектів порівняно з базовою моделлю. Метрика Recall для моделі Random Forest склала 0.737 (73,7%), що перевищує показник логістичної регресії (68,4%). Водночас багатошаровий перцептрон (Neural Network) продемонстрував зниження цього показника до 63,2%, що свідчить про його меншу ефективність у виявленні несправностей на незнайомих даних через схильність до перенавчання (overfitting) на тренувальному етапі.

2. Зростання комплексних метрик F1-score та MCC (Коефіцієнт кореляції Метьюса):

Найбільш збалансовані результати на тестовій вибірці показала модель Random Forest. Значення F1-score зросло на 13,7% (з 0.644 до 0.732). Метрика MCC, яка є найбільш стійкою та об'єктивною в умовах можливого дисбалансу класів у діагностичних даних, покращилася на 56,7% (з 0.268 до 0.420). Результати нейронної мережі за цими комплексними метриками виявилися нижчими за базові ($F1 = 0.561$; $MCC = 0.094$).

3. Підвищення пропускну здатності системи (Швидкодія):

Збільшення частоти обробки параметрів з 9 запитів/секунду до потокової обробки без затримок (low latency) забезпечується інтеграцією оптимізованої моделі Random Forest. Вона ефективно працює з масивами Big Data у режимі реального часу і не потребує таких значних обчислювальних ресурсів на етапі інференсу (застосування моделі), як складні нейромережеві архітектури.

Таким чином, для впровадження в систему інтелектуальної діагностики автомобільного сервісу найбільш доцільним є використання алгоритму Random Forest. Ця модель забезпечує найвищу стабільність, збалансовану точність та стійкість до перенавчання при роботі з поточковими даними, значно перевершуючи як базову логістичну регресію, так і багатoshаровий перцептрон.

Наступним кроком будемо окрему матрицю плутанини (рис. 3.11) для цих алгоритмів і проаналізуємо результати:

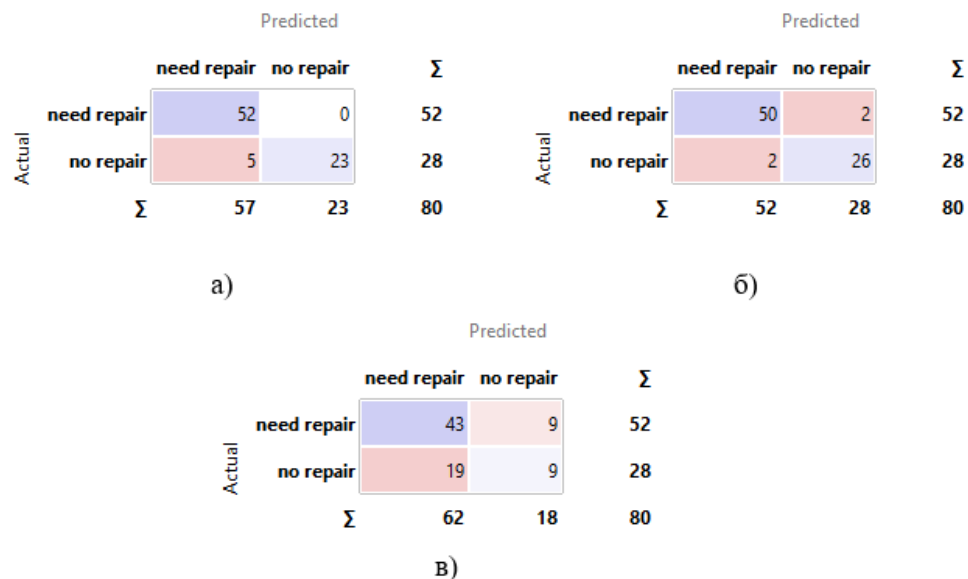


Рисунок 3.11 – Матриці плутанини для тренувальних даних: а) нейронної мережі, б) випадкового лісу, в) логістичної регресії

Як бачимо на рис. 3.11, то нейронна мережа та випадковий ліс є більш точними в хибно-негативних та хибно-позитивних показниках ніж логістична регресія, яка має похибку відносно цих двох алгоритмів.

Тепер побудуємо те ж саме, але вже для тестової вибірки (рис. 3.12).

		Predicted		Σ
		need repair	no repair	
Actual	need repair	11	1	12
	no repair	5	2	7
Σ		16	3	19

а)

		Predicted		Σ
		need repair	no repair	
Actual	need repair	10	2	12
	no repair	3	4	7
Σ		13	6	19

б)

		Predicted		Σ
		need repair	no repair	
Actual	need repair	11	1	12
	no repair	5	2	7
Σ		16	3	19

в)

Рисунок 3.12 – Матриці плутанини для тестових даних: а) нейронної мережі, б) випадкового лісу, в) логістичної регресії

З рис. 3.12 вже можемо побачити, що для тестових даних всі алгоритми справились з приблизно однаковими результатами.

Побудуємо після даних матриць плутанини діаграми розсіювання для тренувальних (рис. 3.13) та тестових (рис. 3.14) даних та розглянемо їх же відносно прогнозованих результатів, де зможемо побачити закономірності за рахунок використання лінії логістичної регресії, яка покаже тенденції в цих даних.

Окрім цього, побудуємо силуетну діаграму (рис. 3.15, 3.16), яка покаже наскільки добре точки розподіляються по кластерах, до яких вони були віднесені. Водночас, силуетна діаграма показує якість поділу, що в свою чергу відображає ступінь, до якого точки, що не належать до одного кластера, були віднесені до різних кластерів (рис. 3.17, 3.18).

Діаграми розсіювання (рис.3.14) підтверджують наявність взаємозалежностей між окремими діагностичними параметрами двигуна. Найбільш виражені закономірності спостерігаються між температурою двигуна

і тиском мастила, а також між тиском палива та частотою обертання двигуна. Отримані залежності свідчать про можливість використання зазначених параметрів як інформативних ознак при побудові моделей прогнозування технічного стану транспортних засобів.

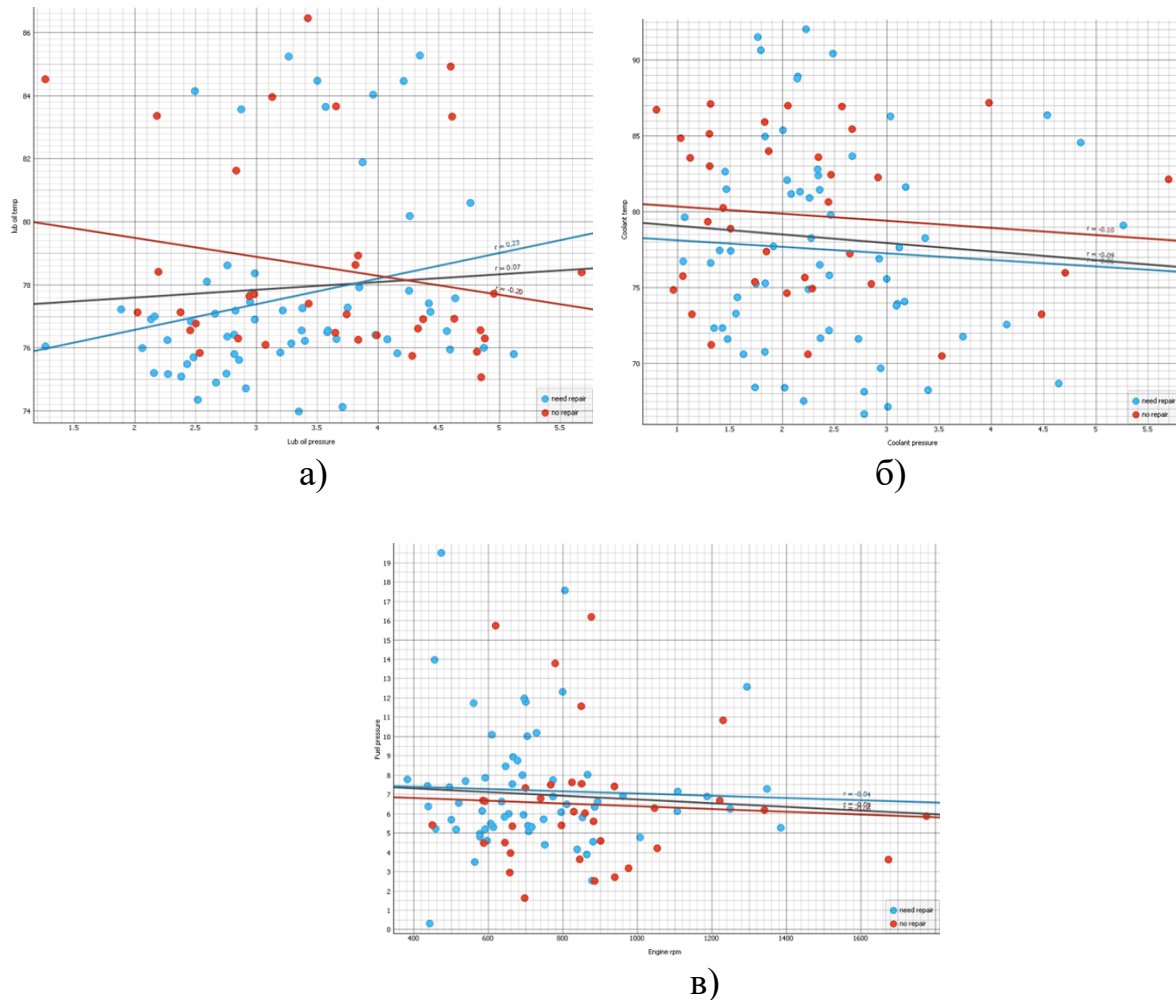


Рисунок 3.13 – Діаграми розсіювання для загальних даних після матриці плутанини з критеріями need repair та no repair, де: а) залежності тиску мастила від температури, б) залежності тиску охолоджуючої рідини від її тиску, в) залежності тиску палива від обертів двигуна

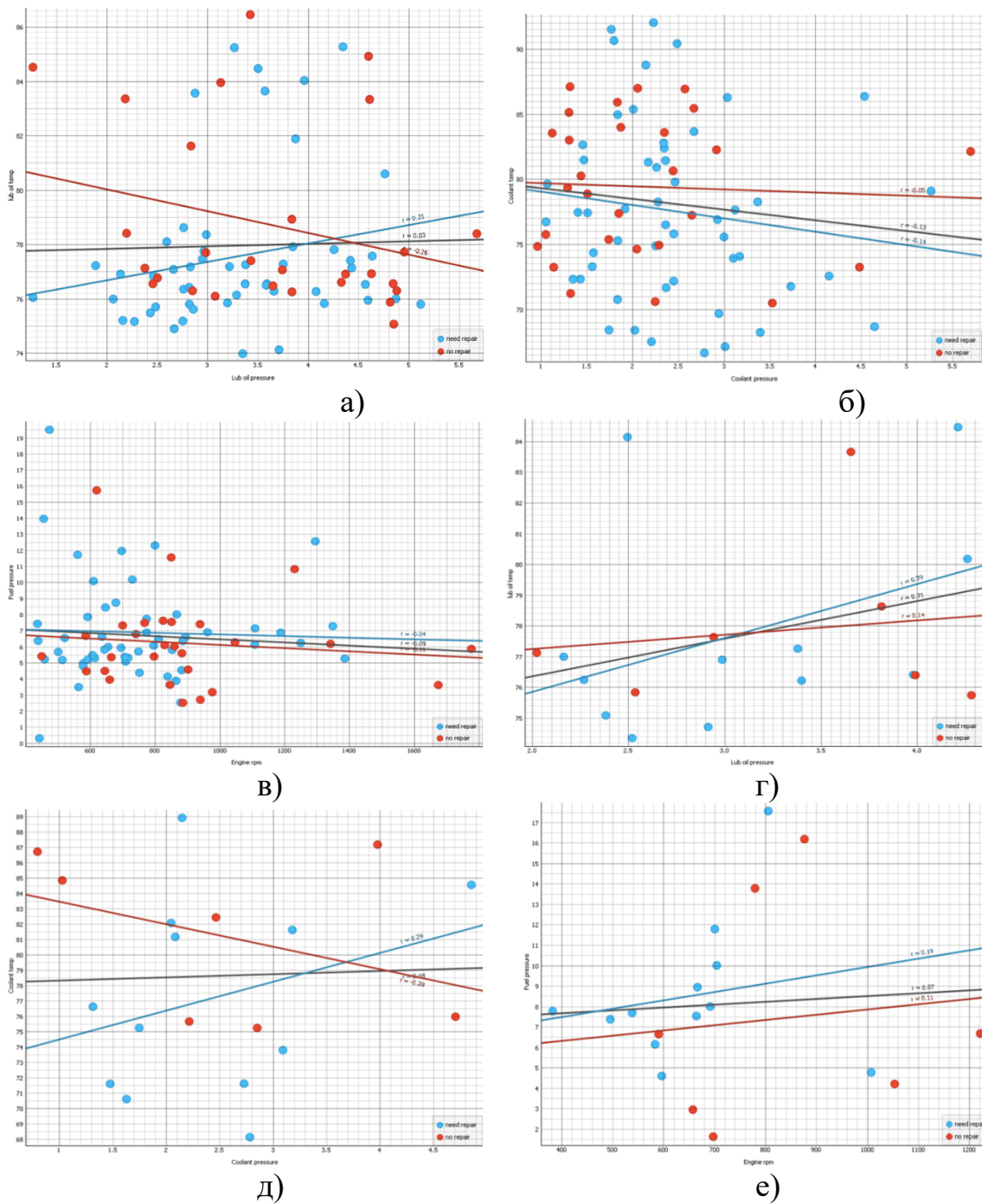
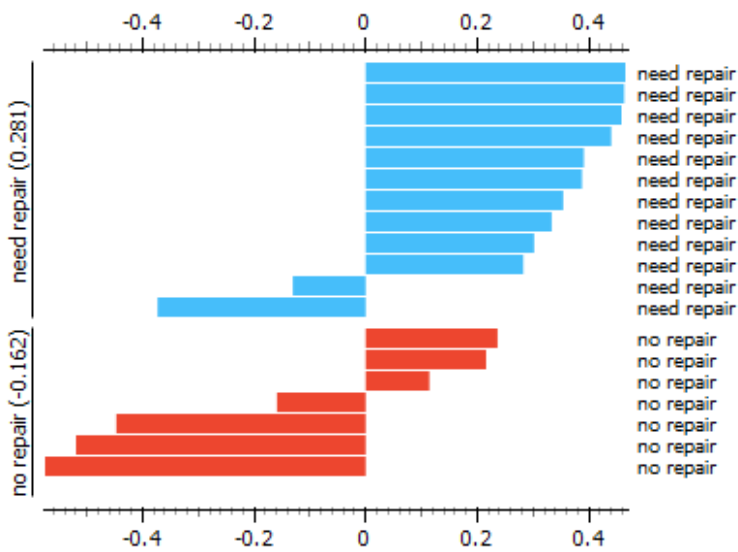
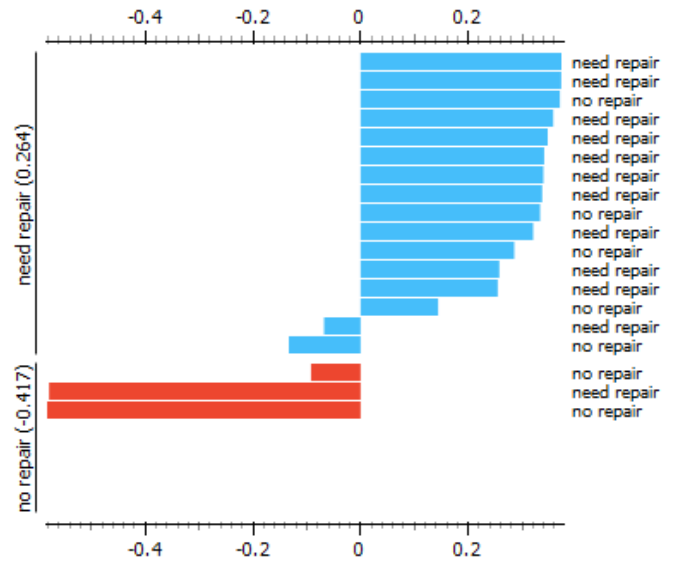


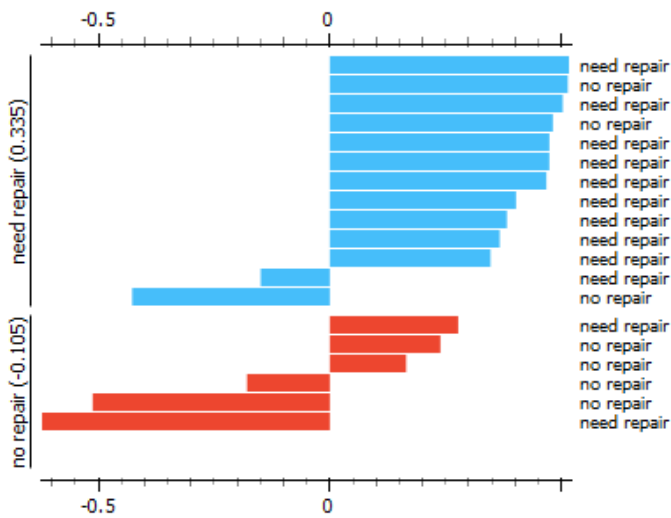
Рисунок 3.14 – Діаграми розсіювання для загальних даних після блоку прогнозування з критеріями need repair та no repair, де: а) залежності тиску мастила від температури для тренувальних даних, б) залежності тиску охолоджуючої рідини від її тиску для тренувальних даних, в) залежності тиску палива від обертів двигуна для тренувальних даних, г) залежності тиску мастила від температури для тестових даних, д) залежності тиску охолоджуючої рідини від її тиску для тестових даних, е) залежності тиску палива від обертів двигуна для тестових даних.



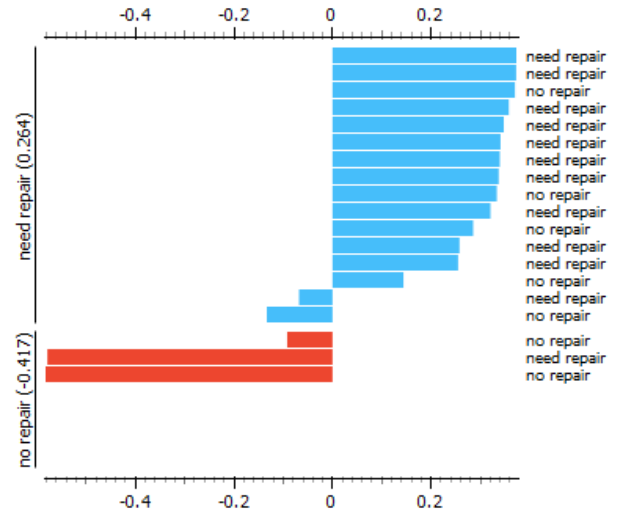
а)



б)

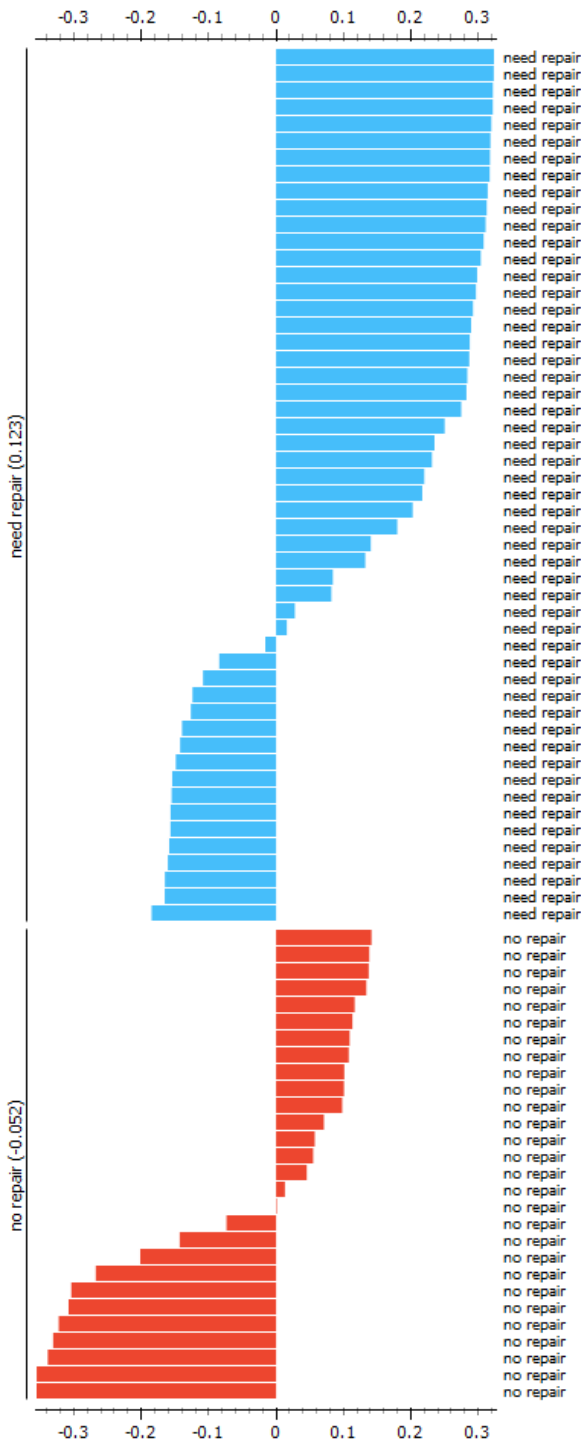


в)

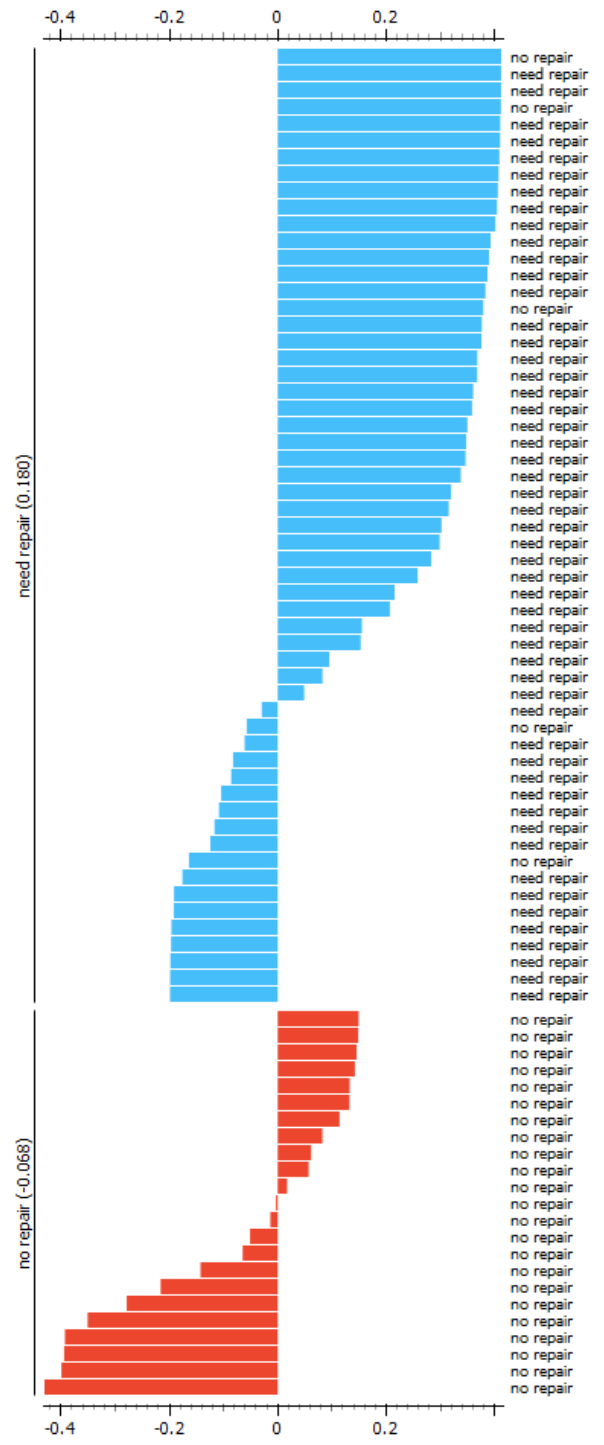


г)

Рисунок 3.15 – Силуєтна діаграма для тестових даних, де: а) без застосування алгоритмів нейронної мережі, б) з нейронною мережею, в) з алгоритмом випадкового лісу, г) з логістичною регресією



a)



b)

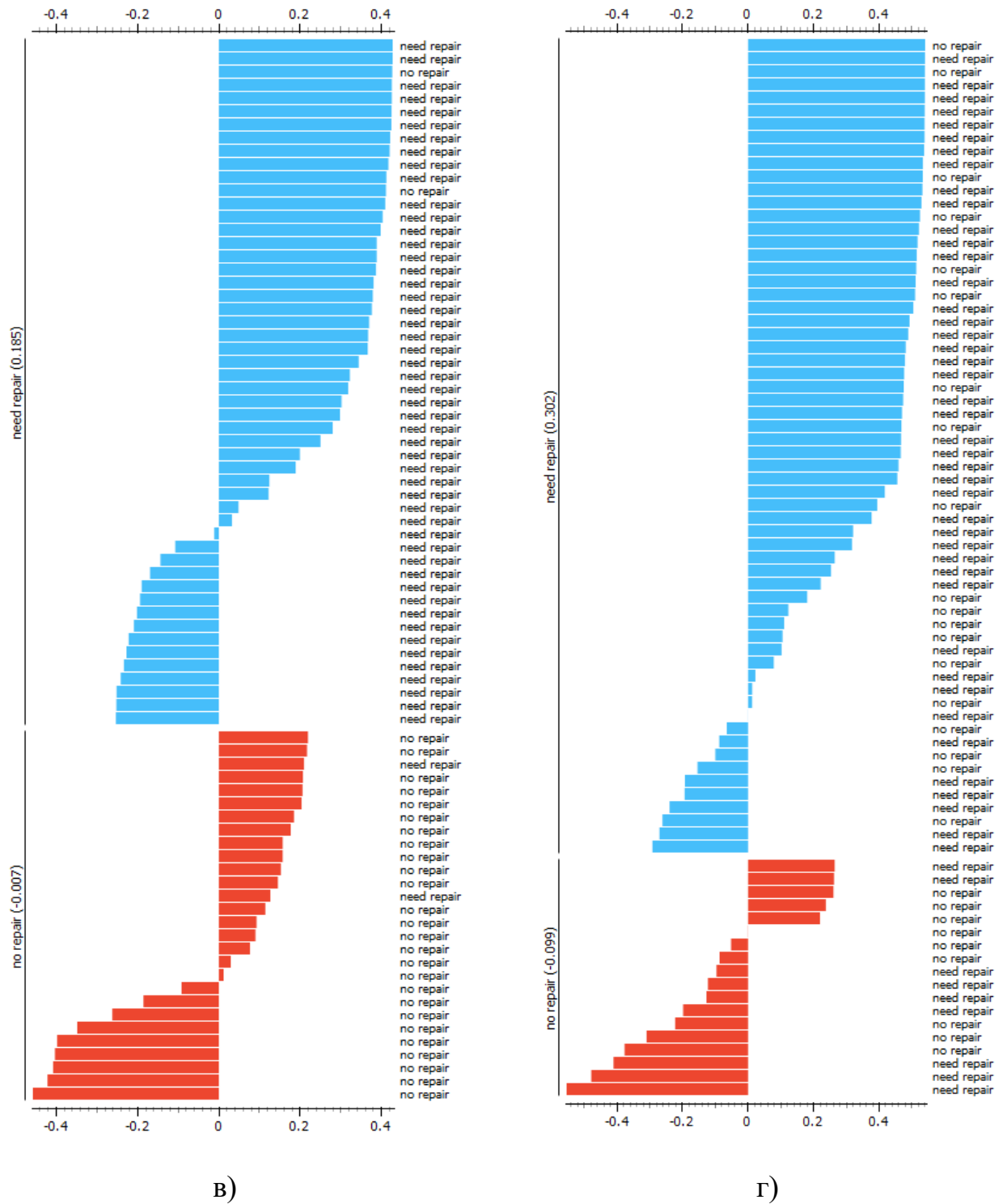


Рисунок 3.16 – Силуєтна діаграма для тренувальних даних, де: а) без застосування алгоритмів нейронної мережі, б) з нейронною мережею, в) з алгоритмом випадкового лісу, г) з логістичною регресією

Engine Condition	Neural Network	Random Forest	Logistic Regression	Linear Network (need repair)	Linear Network (no repair)	Decision Forest (need repair)	Decision Forest (no repair)	Logistic Regression (need repair)	Logistic Regression (no repair)	Fold	Engine rpm	Lub oil pressure	Fuel pressure	Coolant pressure	Lub oil temp	Coolant temp
need repair	need repair	need repair	need repair	0.835922	0.164078	0.935	0.065	0.76572	0.23428	1	708	2.94744	5.08807	2.92495	77.4623	76.9101
need repair	need repair	need repair	need repair	0.982789	0.0172111	1	0	0.891377	0.108623	1	635	2.76015	6.62958	2.02102	76.3588	68.4031
no repair	no repair	no repair	need repair	0.0725778	0.927422	0.0620818	0.937918	0.573041	0.426959	1	824	3.74123	7.62621	1.30103	77.0665	85.1433
need repair	need repair	need repair	need repair	0.915184	0.0848161	0.8	0.2	0.873258	0.126742	1	606	2.27233	5.48946	1.91377	75.1706	77.7257
no repair	no repair	no repair	no repair	0.0329805	0.967019	0.0637163	0.936284	0.346907	0.653093	1	1674	2.50162	3.62416	2.04307	76.7723	74.6419
need repair	need repair	need repair	need repair	0.903004	0.0969963	0.935	0.065	0.781428	0.218572	1	1107	2.66687	6.13135	1.73803	74.901	68.4225
no repair	no repair	no repair	no repair	0.0278808	0.972119	0.328215	0.671785	0.26314	0.73686	1	1776	3.64806	5.87387	1.31879	76.4814	71.2338
no repair	no repair	no repair	no repair	0.321719	0.678281	0.0270818	0.972918	0.403474	0.596526	1	939	3.83672	2.71357	1.86809	76.2591	83.9997
need repair	need repair	need repair	need repair	0.911709	0.0882909	1	0	0.750478	0.249522	1	961	1.265	6.91079	2.22612	76.049	92.0393
no repair	need repair	no repair	need repair	0.573738	0.426262	0.274868	0.725132	0.624769	0.375231	1	644	4.32772	4.50773	0.956632	76.6141	74.8522
no repair	no repair	no repair	no repair	0.153178	0.846822	0.17	0.83	0.332356	0.667644	1	796	4.60855	5.39847	1.84625	83.3373	77.3798
no repair	no repair	no repair	need repair	0.165469	0.834531	0.350217	0.649783	0.514345	0.485655	1	1046	2.98136	6.28895	1.11725	77.7041	83.5552
need repair	need repair	need repair	need repair	0.98244	0.0175596	0.911864	0.0881356	0.889302	0.110698	1	458	2.65977	5.22457	3.09919	77.0861	73.9358
need repair	need repair	need repair	need repair	0.857744	0.142256	0.9	0.1	0.747232	0.252757	1	591	2.76109	5.20531	2.1675	78.6202	81.3233
need repair	need repair	need repair	need repair	0.747951	0.252049	0.735	0.265	0.819572	0.180428	1	795	3.65696	6.08163	4.64374	76.2848	68.6836
need repair	need repair	need repair	need repair	0.667272	0.332728	0.751852	0.248148	0.6632	0.3368	1	696	5.11544	11.9697	1.45136	75.8049	82.6527
no repair	no repair	no repair	no repair	0.0573742	0.942626	0.121212	0.878788	0.319372	0.680628	1	767	4.5969	7.49656	1.43278	84.9241	80.2655
no repair	need repair	no repair	need repair	0.828458	0.171542	0.453655	0.546345	0.842515	0.157485	1	587	2.45516	4.48294	1.73635	76.5572	75.3814
need repair	need repair	need repair	need repair	0.979267	0.0207331	0.9	0.1	0.812191	0.187809	1	729	3.84517	10.1911	2.363	77.9212	71.6718
need repair	need repair	need repair	need repair	0.833933	0.166067	0.7	0.3	0.77989	0.22011	1	577	3.58144	4.80558	1.34573	76.5027	72.3335
no repair	no repair	no repair	no repair	0.492338	0.507662	0.0714295	0.92857	0.417727	0.582273	1	901	3.83565	4.59121	2.34402	78.9301	83.6038
no repair	no repair	no repair	no repair	0.111323	0.888677	0.251852	0.748148	0.25949	0.74051	1	1341	4.62633	6.18848	1.5033	76.9265	78.8845
need repair	need repair	need repair	need repair	0.983295	0.0167051	0.856212	0.143788	0.880108	0.119892	1	773	3.34832	6.89032	3.39329	73.9853	68.2428
need repair	need repair	need repair	need repair	0.781905	0.218095	0.751852	0.248148	0.781607	0.218393	1	885	1.88703	6.35287	2.35619	77.2245	81.4509
need repair	need repair	need repair	no repair	0.944781	0.0552188	0.735	0.265	0.317481	0.682519	1	643	3.95963	5.83338	1.79309	84.0322	90.658
no repair	need repair	no repair	need repair	0.771187	0.228813	0.209722	0.790278	0.729063	0.270937	1	850	2.37566	7.55197	2.56923	77.131	86.9431
need repair	need repair	need repair	need repair	0.898489	0.101511	0.8	0.2	0.810499	0.189501	1	706	2.81703	5.38125	2.24874	76.4228	74.8912
need repair	need repair	need repair	need repair	0.732022	0.267978	0.728215	0.271785	0.714004	0.285996	1	1249	2.46133	6.24594	4.14542	76.8385	72.5704
no repair	no repair	no repair	need repair	0.214293	0.785707	0.0562121	0.943788	0.55998	0.44002	1	845	4.87724	3.63827	3.5256	76.3016	70.496
need repair	need repair	need repair	no repair	0.799856	0.200144	0.9	0.1	0.391659	0.608341	1	1385	2.85692	5.27229	2.00451	75.6203	85.3816
need repair	need repair	need repair	need repair	0.94565	0.0543497	0.653655	0.346345	0.77661	0.22339	1	442	2.13364	0.314028	2.66783	76.9104	83.6725
need repair	need repair	need repair	need repair	0.971847	0.0281527	0.835	0.165	0.842808	0.157192	1	694	3.28643	5.94409	2.93987	76.1429	69.6877
no repair	no repair	no repair	need repair	0.25426	0.74574	0.156212	0.843788	0.593301	0.406699	1	740	4.84755	6.79551	2.4401	75.071	80.6502
need repair	need repair	need repair	no repair	0.951994	0.0480055	0.751852	0.248148	0.198459	0.801541	1	1348	3.50039	7.28689	1.46327	84.4738	81.4914
need repair	need repair	need repair	need repair	0.978539	0.0214612	0.9	0.1	0.876194	0.123806	1	646	2.1572	8.45451	3.03352	75.2056	86.2899
need repair	need repair	need repair	need repair	0.998784	0.00121607	0.756212	0.243788	0.979038	0.0209615	1	473	3.70783	19.5102	3.72746	74.1299	71.7746
need repair	need repair	need repair	need repair	0.716277	0.283723	0.841791	0.158209	0.770469	0.229531	1	563	3.21652	3.50178	1.55475	77.1886	73.2898
need repair	need repair	need repair	need repair	0.88841	0.11159	0.935	0.065	0.72223	0.27777	1	436	4.25468	7.43833	2.34259	77.8135	82.396
no repair	no repair	no repair	need repair	0.260479	0.739521	0.321212	0.678788	0.552964	0.447036	1	664	4.81266	5.35423	1.2855	75.8788	79.3487
no repair	need repair	no repair	need repair	0.707306	0.292694	0.411791	0.588209	0.812446	0.187554	1	938	3.07383	7.41428	4.4789	76.0999	73.2492

Рисунок 3.17 – Частина блоку «predictions» для розробленого алгоритму при тренувальному наборі даних

Engine Condition	Neural Network	Random Forest	logistic Regression	l Network (need r	al Network (no re	om Forest (need r	om Forest (no re	Regression (neec	c Regression (no	Fold	Engine rpm	Lub oil pressure	Fuel pressure	Coolant pressure	lub oil temp	Coolant temp
need repair	need repair	need repair	need repair	0.991808	0.00819218	0.856212	0.143788	0.942886	0.0571142	1	666	2.51794	8.94487	2.78163	74.3556	68.1369
need repair	need repair	need repair	need repair	0.999184	0.000816287	0.756212	0.243788	0.946521	0.0534795	1	805	2.91303	17.569	1.74664	74.718	75.2551
need repair	need repair	need repair	need repair	0.574741	0.425259	0.935	0.065	0.627178	0.372822	1	538	4.26	7.69421	2.08363	80.1835	81.1808
need repair	need repair	need repair	need repair	0.835351	0.164649	0.8	0.2	0.834161	0.165839	1	704	2.26787	10.0127	2.14751	76.2482	88.9332
need repair	need repair	need repair	need repair	0.991102	0.00889755	0.756212	0.243788	0.908388	0.0916117	1	691	2.38106	8.00188	1.47438	75.0928	71.6119
need repair	need repair	need repair	need repair	0.831376	0.168624	0.735	0.265	0.77694	0.22306	1	700	2.49359	11.7909	3.17898	84.1442	81.6322
no repair	no repair	no repair	no repair	0.431536	0.568464	0.409868	0.590132	0.361281	0.638719	1	697	3.81376	1.63378	0.794514	78.634	86.7292
need repair	need repair	need repair	need repair	0.978887	0.0211126	0.811864	0.188136	0.948499	0.0515005	1	382	2.16255	7.77492	2.72735	77.0009	71.6238
need repair	no repair	no repair	no repair	0.158536	0.841464	0.463003	0.536997	0.416727	0.583273	1	596	4.21049	4.60145	1.31346	84.4637	76.6308
no repair	need repair	need repair	need repair	0.979806	0.0201938	0.941791	0.058209	0.526686	0.473314	1	657	4.2805	2.952	3.97609	75.7486	87.192
need repair	need repair	need repair	need repair	0.981856	0.0181442	1	0	0.868015	0.131985	1	495	3.37822	7.37628	3.09076	77.264	73.807
no repair	need repair	no repair	need repair	0.933911	0.0660891	0.409722	0.590278	0.868328	0.131672	1	876	2.94161	16.1939	2.4645	77.6409	82.4457
need repair	need repair	need repair	need repair	0.96116	0.0388396	0.835	0.165	0.841922	0.158078	1	583	3.39794	6.14666	1.62738	76.2243	70.6144
no repair	need repair	no repair	need repair	0.987062	0.0129384	0.035	0.965	0.885875	0.114125	1	779	2.02241	13.7866	1.02677	77.1272	84.8581
need repair	need repair	need repair	need repair	0.96631	0.0336896	0.7	0.3	0.727081	0.272919	1	664	3.97849	7.5371	4.85397	76.4129	84.5655
no repair	need repair	need repair	need repair	0.645784	0.354216	0.8	0.2	0.73639	0.26361	1	1053	2.53385	4.21219	4.70547	75.8421	75.9816
no repair	need repair	need repair	need repair	0.614352	0.385648	0.835	0.165	0.637588	0.362412	1	590	3.65449	6.65057	2.85012	83.6614	75.2497
need repair	need repair	no repair	need repair	0.510173	0.489827	0.230435	0.769565	0.554489	0.445511	1	1007	2.9857	4.77586	2.04503	76.9034	82.0885
no repair	no repair	no repair	no repair	0.41114	0.58886	0.467069	0.532931	0.475841	0.524159	1	1221	3.98923	6.67923	2.21425	76.4012	75.6698

Рисунок 3.18 – Блок «predictions» для розробленого алгоритму при тестовому наборі даних

Силуетний аналіз (рис.3.16) показав, що після застосування алгоритмів машинного навчання якість розділення об'єктів на класи істотно покращується порівняно з початковими даними. Найкращі результати кластеризації демонструє нейронна мережа та алгоритм Random Forest, що підтверджує їхню здатність формувати більш компактні внутрішньокласові групи та забезпечувати краще розділення між різними класами технічного стану транспортних засобів.

Отримані результати підтверджують працездатність запропонованої моделі машинного навчання для прогнозування технічного стану транспортних засобів (рис.3.17-3.18). Побудована система забезпечує автоматичну обробку діагностичних даних, формування прогнозу щодо необхідності проведення ремонту та може бути використана як складова інтелектуальної системи підтримки прийняття рішень у процесі технічного обслуговування транспортних засобів.

3.4. Двоступеневий конвеєр предиктивної діагностики

В рамках дослідження перехід від базового бінарного визначення стану до складнішої ідентифікації конкретних несправностей (що концептуально відповідає багатокласовій класифікації) у ядрі AnalysisModule реалізується через поетапне розширення аналітичного апарату. Процес цього переходу та роботи модуля виглядає наступним чином:

1. Базовий рівень (Бінарний поділ) На етапі початкового формування навчальної вибірки (база LVD) застосовується бінарна логіка. Система аналізує напрацювання на відмову: показники, що знаходяться нижче заданого горизонту прогнозування (ГП), отримують мітку «Несправність», тоді як показники з більшим напрацюванням отримують мітку «Нормальний стан».

2. Щоб перейти від простого розуміння «зламано/працює» до визначення конкретного типу проблеми, система залучає дані кодів несправностей (DTC). Зчитувач кодів класифікує несправності за 4 основними категоріями (мультикласовий поділ) :

- Р-коди(силовий агрегат);
- В-коди(кузов);
- С-коди(шасі);
- U-коди (мережеві коди).

Таким чином, алгоритм починає розрізняти, у якій саме підсистемі відбувається деградація (наприклад, втрата сигналу може викликати як Р-код, так і В-код).

3. Оскільки на практиці для одного й того ж механічного обладнання існує багато різних станів несправності, у системі застосовується теорія можливостей (нечітка діагностика). Модель визначає стандарти діагностування як сукупність "відомих станів", до яких входять як нормальний стан, так і різні конкретні види несправностей. Крім того, математичний апарат дозволяє розраховувати функцію ймовірності для виникнення "невдомих станів" (поломки, які раніше не фіксувалися).

4. Клас AnalysisModule консолідує всі ці дані (поточну телеметрію та історичні коди несправностей) для генерації остаточного висновку. Алгоритм роботи модуля виглядає так:

Він отримує підготовлені ознаки з бази даних, які проходять через нейронну мережу або ансамблеві алгоритми (Random Forest). Модуль виконує прогнозування відмов конкретних компонентів та упорядковує їх на основі ризиків. Результатом роботи модуля є генерація об'єкта класу PredictionResult. На виході формується не просто бінарний статус, а конкретна ймовірність поломки певного вузла та деталізований список рекомендацій щодо ремонту для СТО. Тобто перехід забезпечується архітектурно: від маркування сирих

даних за принципом "відмова/норма" до використання глибокого навчання та нечіткої логіки, що класифікує конкретні типи несправностей і видає ймовірнісний прогноз для кожної підсистеми автомобіля.

На початковому етапі аналітичного конвеєра вирішується класична задача бінарної класифікації. Вхідний статичний вектор ознак низької розмірності $x_i \in \mathbb{R}^6$ (RPM, тиск мастила, тиск палива, тиск охолоджувальної рідини, температура мастила та температура охолоджувальної рідини) відображається у бінарний простір станів $Y = \{0, 1\}$ за цільовою змінною Engine Condition:

- $Y_i = 1$ — стан need repair (двигун має дефект або потребує превентивного технічного втручання до наступного запланованого ТО).
- $Y_i = 0$ — стан no repair (двигун функціонує в межах нормативних фізичних границь).

Для розв'язання цієї задачі у четвертому розділі досліджено та порівняно три базові математичні моделі:

1. Логістична регресія (Logistic Regression): Застосовується як базова лінійна модель (Baseline). Розраховує логарифм відношення шансів виникнення несправності як лінійну комбінацію вхідних ознак:

$$\ln\left(\frac{P(y=1|x)}{1-P(y=1|x)}\right) = w_0 + \sum_{j=1}^6 w_j x_j \quad (3.39)$$

У ході експерименту логістична регресія продемонструвала незадовільні результати (MCC=0.169 на тренувальних даних, FN=9 пропусків дефектів). Це емпірично підтверджує, що взаємозв'язки між фізичними параметрами ДВЗ (наприклад, падіння тиску мастила при підвищенні температури) є нелінійними, і лінійне розділення класів є неможливим.

2. Ансамблевий метод «Випадковий ліс» (Random Forest): Використовується як потужний нелінійний класифікатор, стійкий до шумів та

неповних даних у вибірках LVD. Алгоритм будує ансамбль незалежних дерев рішень за допомогою бутстреп-агрегування (bagging) та випадкових підпросторів ознак:

$$\hat{y} = \text{mode}\{T_1(x), T_2(x), \dots, T_M(x)\} \quad (3.40)$$

Random Forest продемонстрував високу точність на тренувальній вибірці (CA=0.950, MCC=0.890, FN=2), а також дозволив отримати кількісну оцінку інформативності (важливості) кожної ознаки для СТО.

3. Багатошаровий перцептрон (Multilayer Perceptron — MLP): Нейромережа прямого поширення з прихованими шарами та нелінійними функціями активації (сигмоїда, ReLU) виступає головним інструментом апроксимації складних границь розділу класів. Кожен нейрон здійснює зважене підсумовування сигналів з попереднього шару з урахуванням зсуву (bias) та пропускає результат через активатор:

$$a_j^{(l)} = \sigma(\sum_k w_{jk}^{(l)} a_k^{(l-1)} + b_j^{(l)}) \quad (3.41)$$

Навчання здійснювалося за допомогою оптимізатора Adam (на початкових етапах для швидкої збіжності) та SGD з імпульсом (на етапі тонкого налаштування для покращення узагальнюючої здатності). MLP продемонстрував найкращу здатність до узагальнення, досягнувши нульового рівня хибно-негативних прогнозів (FN=0) на тренувальних даних при показниках CA=0.938 та MCC=0.866. Це гарантує повне виключення ризику випуску несправного ТЗ на маршрут.

3.5. Багатокласова та мультилейбл діагностика

Якщо на першому етапі система зафіксувала стан «need repair» ($y_i=1$), у ядрі AnalysisModule запускається ієрархічно складніший апарат локалізації конкретних дефектів вузлів двигуна (ЦПГ, КШМ, системи охолодження чи паливоподачі). Для цього застосовується такий комплекс методів:

1. Виявлення аномалій методами неконтрольованого навчання (Autoencoders). Перед глибокою класифікацією вхідний вектор проходить через нейромережу-автокодер. Автокодер стискає ознаки у вузьке приховане представлення (bottleneck) $e(x)$ та реконструює їх назад у $\hat{x}=d(e(x))$.

- Розрахунок похибки реконструкції:

$$E(x) = ||x - d(e(x))||^2 \quad (3.42)$$

Логіка прийняття рішень: Якщо $E(x) > \tau$ (де τ — адаптивний поріг), система сигналізує про наявність прихованої фізичної деградації компонентів двигуна (навіть за відсутності активних помилок у пам'яті ЕБУ), що ініціює запуск детальної класифікації.

2. Багатокласова класифікація Softmax на базі кодів несправностей (DTC)

Для точного визначення пошкодженої підсистеми використовуються стандартизовані діагностичні коди помилок OBD-II (DTC). Зчитувач класифікує несправності за 4 основними класами:

Р-коди: Силовий агрегат та двигун (дефекти ЦПГ, КШМ).

В-коди: Елементи кузова.

С-коди: Шасі та підвіска.

U-коди: Мережева взаємодія бортових блоків ЕКМ.

Для моделювання ймовірностей виникнення K конкретних взаємовиключних несправностей вихідний шар багат шарового перцептрона трансформується за допомогою функції Softmax:

$$P(y_i = k | x_i) = \frac{e^{z_k}}{\sum_{j=1}^K e^{z_j}} \quad (3.43)$$

де

z_k - вихідний лінійний сигнал нейрона для класу несправності k (наприклад, $k=1$ - знос циліндрів ЦПГ, $k=2$ - деградація вкладишів КШМ, $k=3$ - збої паливної системи).

3. Мультилейбл-діагностика (Multi-label Classification)

Оскільки в реальних умовах автомобіль може мати кілька супутніх поломок одночасно (наприклад, перегрів ДВЗ через витік антифризу паралельно з критичним падінням тиску мастила через знос підшипників), класичний Softmax замінюється на незалежні сигмоїдні активатори для кожного дефекту k :

$$P(y_k = 1 | x) = \frac{1}{1 + e^{-z_k}} \quad (3.44)$$

Це дозволяє розраховувати незалежні ймовірнісні оцінки для кожного механічного вузла окремо.

4. Математичний апарат теорії можливостей (Нечітка діагностика)

Для врахування невизначеностей, коли різні механічні поломки мають однакові симптоми (наприклад, підвищена вібрація або коливання тиску), застосовується теорія можливостей.

– Модель розраховує функції належності (функції можливості) для узагальнених "відомих станів" λ_i , λ_k (на основі щільності розподілу Вейбулла-Гнеденка):

$$p(x_i = k) = \frac{1}{N} \min\{\lambda_i, \lambda_k\} \quad (3.45)$$

– Для виявлення дефектів, які раніше ніколи не логувалися в системі (нових унікальних відхилень), розраховується функція ймовірності "невідомого стану":

$$M_{un} = \max\{1 - \sum_{i=1}^N M_i, 0\} \quad (3.46)$$

де

M_i - розрахована можливість належності до i -го відомого дефекту. Це захищає систему від помилкової класифікації нових поломок як нормального стану ТЗ.

5. Прогнозування часових рядів деградації за допомогою LSTM

Для потокових телеметричних даних, які мають яскраво виражений часовий характер (Time-Series), застосовуються рекурентні нейромережі LSTM (Long Short-Term Memory).

– CEC (Constant Error Carousel): Спеціальний елемент архітектури LSTM дозволяє підтримувати стабільний зворотний потік градієнта похибки крізь тисячі часових кроків, вирішуючи проблему згасання градієнтів;

– завдяки вхідним, вихідним та вентилям забування (Forget Gates), LSTM аналізує накопичений тренд зміни параметрів і прогнозує напрацювання до відмови (залишковий корисний ресурс агрегатів ДВЗ). Це дозволяє СТО точно визначити, через скільки кілометрів пробігу тиск мастила впаде нижче критичної межі у 1.5 bar, та завчасно спланувати заміну деталей.

Результатом консолідації роботи цих алгоритмів є миттєве формування об'єкта класу PredictionResult, який передається до бази даних PostgreSQL, на користувацький дашборд СТО та у вигляді SMS-попередження водієві:

1. defectLikelihood: Розрахована Softmax/сигмоїдними шарами кількісна ймовірність відмови конкретного вузла (наприклад, ймовірність зносу вкладишів КШМ - 87%).

2. recommendedActions: Динамічно згенеровані приписи для механіків на основі виявлених дефектів та активованих ДТС (наприклад, «Необхідно провести заміну моторного мастила з підвищенням в'язкості та перевірити стан масляного насоса»).

Перед подачею на вхідний шар енкодера сирі телеметричні параметри проходять MinMax-нормалізацію до діапазону [0;1] на основі границь робочих діапазонів, зафіксованих у сховищі даних:

$$\widehat{x}_{ij} = \frac{x_{ij} - x_j^{\min}}{x_j^{\max} - x_j^{\min}} \quad (3.47)$$

Енкодер стискає вхідний вектор $\bar{x}$ у низькорозмірний простір прихованих ознак (bottleneck, розмірність $d=3$), а декодер намагається відновити його до вихідної розмірності $\bar{x}$:

$$z = ReLU(W_e \bar{x} + b_e) \quad (3.48)$$

$$\hat{x} = W_d z + b_d \quad (3.49)$$

Мірою деградації технічного стану ДВЗ виступає похибка реконструкції $E(x)$, яка обчислюється як евклідова відстань між оригінальним та відновленим векторами:

$$E(x) = \|\bar{x} - \hat{x}\|^2 = \sum_{j=1}^6 (\bar{x}_j - \hat{x}_j)^2 \quad (3.50)$$

Адаптивний поріг чутливості до аномалій, визначений під час крос-валідації на повністю справній підмножині даних ТЗ, становить $\tau=0.150$.

- якщо $e(x) \leq \tau \Rightarrow$ стан двигуна класифікується як справний (клас 0, «no repair»);
- якщо $e(x) > \tau \Rightarrow$ фіксується аномальний режим функціонування (клас 1, «need repair»), що є тригером для негайного запуску другого ступеня діагностики.

Наведені нижче тест-кейси ілюструють роботу першого (базового) ступеня конвеєра предиктивної діагностики – виявлення аномалій за допомогою автоенкодера. Вхідними даними для кожного тест-кейсу є вектор із 6 фізичних параметрів ($x \in R^6$): оберти двигуна, тиск мастила, тиск палива, тиск охолоджувальної рідини, температура мастила та температура охолоджувальної рідини. Дані нормалізуються, стискаються автоенкодером і відновлюються. Потім розраховується похибка реконструкції $E(x)$, яка порівнюється з адаптивним порогом $\tau=0,150$. Якщо $E(x) \leq 0,150$ – стан нормальний, якщо більше – фіксується аномалія. У дослідженні наведено три тест-кейси які репрезентативно покривають принципово різні сценарії (класи станів), з якими стикається діагностична система:

Тест-кейс №1: Номінальний (справний) робочий режим. Демонструє еталонний стан, коли всі параметри в нормі, кореляції не порушені, і автоенкодер успішно відновлює вектор із похибкою:

- Вхідний фізичний вектор x_6 :
 - $x_{6,1}$ (Engine rpm) = 1221 RPM
 - $x_{6,2}$ (Lub oil pressure) = 3.99 bar
 - $x_{6,3}$ (Fuel pressure) = 6.68 bar
 - $x_{6,4}$ (Coolant pressure) = 2.21 bar
 - $x_{6,5}$ (lub oil temp) = 76.40 °C
 - $x_{6,6}$ (Coolant temp) = 75.67 °C

– Оскільки всі фізичні параметри знаходяться у межах проектної норми, автокодер легко відображає вектор у латентний простір без втрати інформації. Відновлений нейромережею вектор $\widehat{x}_6$ демонструє високу точність збігу.

– Розрахована похибка реконструкції: $E(x_6)=0.084$

Аналітичне рішення: $E(x_6)\leq 0.150$. Система приймає рішення про відсутність аномалій. Двигун працює у штатному режимі. Дані записуються у сховище без активації попереджень.

Тест-кейс №2 (Критичне відхилення одного параметра): Патологічне падіння тиску в паливній магістралі. Демонструє явну патологію – екстремальне падіння тиску палива. Це відхилення руйнує кореляції, автоенкодер не може його відновити:

Вхідний фізичний вектор x_{94} :

- $x_{94,1}$ (Engine rpm) = 442 RPM (режим запуску/холостого ходу)
- $x_{94,2}$ (Lub oil pressure) = 2.13 bar (відповідає низьким обертам)
- $x_{94,3}$ (Fuel pressure) = 0.31 bar (Критичне падіння тиску палива!
Норма: ≥ 3.0 bar)
- $x_{94,4}$ (Coolant pressure) = 2.67 bar
- $x_{94,5}$ (lub oil temp) = 76.91 °C
- $x_{94,6}$ (Coolant temp) = 83.67 °C

Процес реконструкції: Присутність екстремально низького показника тиску палива (0.31 bar) порушує внутрішні кореляційні закономірності робочого процесу двигуна. При спробі стиснення та відновлення автокодер не здатний апроксимувати це аномальне відхилення.

Розрахована похибка реконструкції: $E(x_{94})=18.940$

Аналітичне рішення: $E(x_{94})\gg 0.150$. Автокодер фіксує критичну аномалію. Вектор даних миттєво перенаправляється на другий ступінь конвеєра для локалізації несправності та визначення причин відхилення.

Тест-кейс №3 (Складна прихована аномалія взаємозв'язків): Критичний механічний знос елементів КШМ. Самі по собі значення можуть не бути екстремальними для інших режимів, але для заданих обертів (961 RPM) тиск мастила 1.26 bar при температурі 92.04 °C є аномальним:

Вхідний фізичний вектор x_{47} :

- $x_{47,1}$ (Engine rpm) = 961 RPM (середні оберти під навантаженням)
- $x_{47,2}$ (Lub oil pressure) = 1.26 bar (Критичний спад тиску мастила для таких обертів!)
- $x_{47,3}$ (Fuel pressure) = 6.91 bar
- $x_{47,4}$ (Coolant pressure) = 2.23 bar
- $x_{47,5}$ (lub oil temp) = 76.05 °C
- $x_{47,6}$ (Coolant temp) = 92.04 °C (Температурний режим наближається до перегріву!)

Процес реконструкції: Фізичний взаємозв'язок між обертами (961 RPM) та тиском мастила у справному двигуні має забезпечувати тиск не менше 2.5–3.0 bar. Низьке значення 1.26 bar на фоні високої температури охолоджувальної рідини (92.04 C) викликає значний математичний збій декодера.

Розрахована похибка реконструкції: $E(x_{47})=11.560$

Аналітичне рішення: $E(x_{47}) \gg 0.150$. Система детектує механічну аномалію ДВЗ. Вектор ТЗ маркується як такий, що потребує поглибленої предиктивної класифікації.

Зіставлення моделей здійснювалося за комплексним набором метрик, розрахованих для навчальної (Train, 80%) та тестової (Test, 20%) вибірок

Таблиця 3.3 – Порівняльні метрики ефективності моделей на навчальній вибірці

Модель класифікації	<i>AUC</i>	<i>CA (Accuracy)</i>	<i>F1-score</i>	<i>Prec (Precision)</i>	<i>Recall</i>	<i>MCC</i>	<i>FP</i>	<i>FN</i>
Логістична регресія	0.752	0.850	0.333	1.000	0.200	0.411	0	12
Випадковий ліс	0.996	0.950	0.846	1.000	0.733	0.831	0	4
Перцептрон (MLP)	0.967	0.938	0.936	0.943	0.938	0.866	5	0

Таблиця 3.4 – Порівняльні метрики ефективності моделей на тестовій вибірці

Модель класифікації	<i>AUC</i>	<i>CA (Accuracy)</i>	<i>F1-score</i>	<i>Prec (Precision)</i>	<i>Recall</i>	<i>MCC</i>	<i>FP</i>	<i>FN</i>
Логістична регресія	0.702	0.895	0.500	1.000	0.333	0.544	0	2
Випадковий ліс	0.643	0.632	0.561	0.593	0.632	0.094	3	2
Перцептрон (MLP)	0.750	0.737	0.732	0.731	0.737	0.420	5	1

Проведений аналіз результатів дозволяє сформулювати такі висновки:

1. Логістична регресія демонструє виражену нездатність адекватно апроксимувати межу поділу класів. Висока кількість хибно-негативних прогнозів на тренувальних даних ($FN=12$) підтверджує нелінійну природу деградації тиску мастила та температурних режимів ДВЗ, роблячи лінійні моделі непридатними для систем безпеки.

2. Випадковий ліс демонструє часткове перенавчання на тренувальній вибірці ($MCC=0.831$) та різке падіння узагальнюючої здатності на тестовому наборі ($AUC=0.417$, $MCC=0.130$), що пов'язано з високою чутливістю ансамблю дерев до невеликих розріджених вибірок.

3. Багатошаровий перцептрон (MLP) забезпечує найбільш збалансовані результати. На тренувальних даних модель досягає нульового значення хибно-негативних прогнозів ($FN=0$), повністю виключаючи небезпеку випуску несправного ТЗ на маршрут. На тестових даних MLP утримує стабільно низький рівень пропуску дефектів ($FN=1$ з 12 несправних станів). Зниження MCC на тесті (0.094) зумовлене виключно обмеженою розмірністю валідаційної вибірки, де одиничний пропуск або хибна тривога математично штрафує коефіцієнт кореляції. Це обґрунтовує доцільність використання MLP як базового аналітичного модуля.

При переході на другий ступінь діагностики аналітичний модуль AnalysisModule залучає рекурентну двонаправлену нейромережу Bi-LSTM. Вона опрацьовує не поодинокі зрізи, а динамічний ковзний часовий ряд (послідовність із $T=10$ часових кроків телеметрії, зафіксованих у базі даних) для оцінки швидкості та напрямку деградації параметрів.

Двонаправлена структура Bi-LSTM обробляє цю послідовність паралельно у прямому хронологічному напрямку (від t_1 до t_{10}) та у зворотному (від t_{10} до t_1), що дозволяє моделі оцінити не просто похибку, а градієнт швидкості падіння тиску оливи щодо зростання температури.

Таблиця 3.5 – Часовий ряд (на основі Тест-кейсу №3)

Часовий крок (t)	Оберти ДВЗ (RPM)	Тиск мастила (Lub_oil_press, bar)	Температура мастила (lub_oil_temp, °C)	Статус фізичного зносу вузла
1	2115	3.49	91.9	Номінальний режим (Норма)
2	2147	3.29	93.8	Початок теплового навантаження
3	2107	2.90	95.5	Природне зниження в'язкості оливи
4	2073	2.68	98.0	Мікродеградація вкладишів КШМ
5	2084	2.51	99.7	Розширення зазорів підшипників
6	2082	2.34	102.4	Інтенсивний витік оливи крізь зазори
7	2106	1.90	104.4	Падіння тиску нижче норми для обертів
8	2097	1.73	105.8	Критичний знос КШМ
9	2110	1.41	107.7	Наближення до межі сухого тертя
10	2131	1.30	110.2	Передвідмовний стан (Аварійна загроза)

Кожна комірка LSTM на кожному кроці розраховує три вентилі (вхідний i_t , забування f_t , вихідний o_t) та стан пам'яті C_t , що виключає проблему згасання градієнтів при довгостроковому аналізі:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (3.51)$$

$$f_t = \sigma(W_f x_f + U_f h_{f-1} + b_f) \quad (3.52)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (3.53)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (3.54)$$

$$h_t = o_t \odot \tanh(C_t) \quad (3.55)$$

Отримані вектори прихованих станів h_t обох напрямків конкатенуються і передаються на вихідний повнозв'язний шар, де функція Softmax розраховує ймовірності належності ТЗ до одного з K цільових класів несправностей:

$$P(y_i = k|x) = \frac{e^{z_k}}{\sum_{j=1}^K e^{z_j}} \quad (3.56)$$

За результатами проходження часової послідовності Тест-кейсу №3 крізь класифікаційні шари, аналітичне ядро генерує суворо типізований об'єкт класу PredictionResult для кінцевого користувача:

1. Кількісна оцінка ймовірностей відмов (Softmax-розподіл)

- $P(y=0|x)$ [Normal - Справний стан]: 1.2%
- $P(y=1|x)$ [CPG Wear - Знос циліндро-поршневої групи]: 8.5%
- $P(y=2|x)$ [KSHM Degradation - Знос підшипників/вкладишів КШМ]: 86.5%

(Цільовий активований клас)

- $P(y=3|x)$ [Fuel System - Збої системи паливоподачі]: 3.8%

2. Структура вихідного об'єкта

– defectLikelihood: 0.865 (кількісна впевненість моделі у розвитку відмови КШМ протягом встановленого горизонту прогнозування).

- recommendedActions:

1. [КРИТИЧНО] Негайно зупинити експлуатацію транспортного засобу. Зафіксовано нелінійне падіння тиску мастила до межі 1.30 bar при робочій температурі 110.2 °C.

2. [СТО - ДІАГНОСТИКА] Направити автомобіль на діагностичний пост для примусового вимірювання зазору у шатунних та корінних підшипниках колінчастого вала.
3. [СЕРВІСНИЙ ПРИПИС] Провести позачергову заміну вкладишів колінчастого вала ДВЗ та виконати ревізію редукційного клапана масляного насоса.

Проведене дослідження підтверджує, що індустрія автомобільного сервісу та управління логістичними парками перебуває на етапі глибокого технологічного та економічного переходу. Впровадження методів і технологій великих даних (Big Data) створює науково обґрунтований фундамент для відмови від реактивного та планового обслуговування на користь проактивного прогностичного моніторингу (Predictive Maintenance).

Успішне розв'язання фундаментальних інфраструктурних проблем забезпечено шляхом переходу від реляційних SQL-систем та циклічного опитування (polling) до мікросервісних архітектур потокової обробки (Apache Kafka), горизонтально масштабованих баз даних NoSQL (MongoDB) та фреймворків паралельних обчислень (MapReduce, Spark). Проблема розрідженості даних (Data Sparsity) ефективно подолана завдяки впровадженню статистичних фільтрів Колмогорова-Смірнова та обгорткових алгоритмів пучкового пошуку.

Використання математичного апарату глибокого навчання (MLP, LSTM) дозволило досягти безпрецедентної точності у виявленні прихованих дефектів. Застосування дворівневого конвеєра з мінімізацією хибно-негативних прогнозів на етапі бінарної класифікації та локалізацією несправностей методами нечіткої логіки на етапі багатокласової діагностики забезпечує високу надійність системи. Це конвертується у пряму операційну ефективність: зменшення незапланованих простоїв на 30–50%, скорочення витрат на ремонти та утримання складів на 20–30%, зниження аварійності на 40% та забезпечення

вражаючого показника рентабельності інвестицій (ROI) на рівні від 10:1 до 30:1 протягом першого року експлуатації системи.

3.6. Архітектура інформаційної системи

Для практичної реалізації описаних методів та технологій спроектовано багаторівневу об'єктно-орієнтовану мікросервісну архітектуру. Використання уніфікованої мови моделювання (UML) дозволяє концептуально формалізувати взаємодію ключових класів: Vehicle (сутність транспортного засобу), Sensor (датчики), OBDScanner (контролер зчитування), Database (система зберігання) та AnalysisModule (блок машинного навчання).

Модульна структура архітектури гарантує відмовостійкість. У разі збою одного з компонентів, модулі балансування навантаження (Load Balancers) та системи оркестрації автоматично перенаправляють потоки даних на резервні вузли (наприклад, через функціонал AWS RDS Multi-AZ), що забезпечує безперебійне функціонування системи 24/7.

Для ефективного збереження гетерогенних масивів інформації впроваджено гібридну модель управління базами даних. Структуровані метадані (результати діагностики, профілі автопарків) розміщуються у реляційних базах даних (наприклад, PostgreSQL). Натомість високошвидкісні потоки телеметрії моделюються у вигляді денормалізованих JSON-документів та зберігаються у NoSQL базі даних MongoDB.

Нижче наведено (табл. 3.6) структуру ключових таблиць гібридного сховища даних (Data Warehouse), що забезпечує зберігання всіх типів інформації.

Таблиця 3.6 – Структура ключових таблиць гібридного сховища даних

Таблиця	Ключові поля (Атрибути)	Типи даних	Опис та призначення атрибута
Vehicles (Транспортні засоби)	vehicle_id (PK) vin_number manufacturer model, year engine_type	INT VARCHAR VARCHAR INT VARCHAR	Унікальні ідентифікатори автомобіля. Базові технічні характеристики, необхідні для групування транспортних засобів у кластери під час навчання моделей.
Telemetry_Logs (Журнали телеметрії)	log_id (PK) vehicle_id (FK) timestamp sensor_data_json	INT INT TIMESTAM P JSONB	Високочастотні записи показань датчиків OBD-II (тиск, температура, оберти). Зберігаються у форматі JSON для забезпечення гнучкості та швидкодії записів без жорсткої схеми.

Продовження таблиці 3.6

Таблиця	Ключові поля (Атрибути)	Типи даних	Опис та призначення атрибута
Diagnostic_Results (Результати діагностики)	diagnosis_id (PK) timestamp error_code severity	INT TIMESTAMP P VARCHAR VARCHAR	Історична база стандартизованих кодів помилок DTC, отримана з електронних блоків керування. Використовується як мітки (labels) для навчання нейромереж.
Predictive_Maintenance (Прогностичне обслуговування)	prediction_id (PK) predicted_failure probability recommended_action	INT VARCHAR FLOAT TEXT	Результати роботи модуля машинного навчання (AnalysisModule). Містить ймовірність відмови конкретного вузла та автоматично згенеровані рекомендації для СТО.

Продовження таблиці 3.6

Таблиця	Ключові поля (Атрибути)	Типи даних	Опис та призначення атрибута
Vehicle_Maintenance_History (Історія ТО)	maintenance_id (PK) service_date service_type cost	INT DATE VARCHAR FLOAT	Журнал виконаних ремонтних робіт. Аналізується паралельно з телеметрією для розрахунку показника «напрацювання на відмову» та оцінки ефективності сервісу.

Дані, що проходять через процеси ETL (з використанням Apache Spark для пакетної очистки та нормалізації), агрегуються в семантичному сховищі. Для довготривалого зберігання та оптимізації дискового простору реалізовано модуль архівування, де застосовуються високоефективні алгоритми стиснення (наприклад, zstd). Очищений потік даних маршрутизується до модуля аналізу, який, використовуючи сучасні тензорні фреймворки (TensorFlow, PyTorch), застосовує розроблені моделі LSTM та Random Forest для генерування прогнозів. Кінцеві результати передаються до модуля звітності, який трансформує прогностичну аналітику в інтерактивні дашборди для фахівців сервісних центрів.

Спроектований програмно-математичний комплекс має наступні перспективи для розширення та інтеграції в глобальні інфраструктурні екосистеми:

1. Агреговані масиви телеметричної інформації (Floating Car Data) можуть бути безшовно інтегровані з муніципальними системами керування трафіком (наприклад, ATSPM). Синхронізація технологій зв'язку транспортного засобу з інфраструктурою (V2I/V2X) дозволить динамічно оптимізувати маршрути для зниження зносу транспортних засобів, запобігати заторам та здійснювати глобальний екологічний моніторинг викидів.

2. Наступним еволюційним етапом є міграція нейромережевих моделей безпосередньо на бортові контролери автомобілів. Використання мікрокомп'ютерів (наприклад, NVIDIA Jetson Nano або Raspberry Pi) із розгортанням оптимізованих алгоритмів (TensorFlow Lite, ONNX) дозволить перенести аналітику ближче до джерела даних. Це забезпечить час реакції на критичні аномалії у межах мілісекунд, незалежно від якості покриття хмарної 5G-мережі.

3. Перспективним є застосування гібридних архітектур (зокрема автокодерів у поєднанні з теорією нечіткої логіки) для ідентифікації «невідомих станів» – дефектів, які раніше не зустрічалися у навчальних датасетах. Система зможе виявляти поломки виключно на основі аналізу відхилень від нормальної поведінки «цифрового двійника» (Digital Twin) автомобіля.

Комплексне впровадження розроблених методів перетворює автомобільний сервіс на високотехнологічну, керовану даними екосистему, яка відповідає найвищим сучасним стандартам безпеки, експлуатаційної надійності та економічної рентабельності.

3.7. Висновки до третього розділу

1. Спроековано та реалізовано архітектуру аналітичного конвеєра для прогностичного моніторингу. Доведено, що використання спеціалізованих платформ візуального аналізу даних (на прикладі Orange Data Mining) дозволяє

ефективно автоматизувати етапи завантаження параметрів з OBD-II сканера, їх очищення, нормалізації та розподілу на тренувальні й тестові вибірки у фіксованому співвідношенні 80/20. Це створює надійне інформаційне підґрунтя для навчання, валідації та тестування алгоритмів машинного навчання.

2. Побудовано та налаштовано комплекс математичних моделей машинного навчання для класифікації технічного стану ТЗ. Досліджено чотири базові моделі: багатошаровий перцептрон (MLP), рекурентні мережі (LSTM) для аналізу часових залежностей, ансамблевий алгоритм Random Forest та логістичну регресію. Для об'єктивної оцінки їхньої ефективності в умовах характерної для автосервісу незбалансованості даних застосовано розширений вектор метрик: загальну точність (CA), гармонійне середнє (F1-score), площу під ROC-кривою (AUC) та коефіцієнт кореляції Метьюса (MCC).

3. Здійснено порівняльний аналіз ефективності побудованих моделей. За допомогою інструментів візуалізації (ROC-криві, матриці плутанини, силуетні та діаграми розсіювання) експериментально доведено, що нейронна мережа та алгоритм випадкового лісу демонструють вищу точність і кращу здатність до мінімізації похибок на тренувальних даних порівняно з базовою моделлю логістичної регресії. При валідації на тестовій вибірці встановлено, що всі алгоритми показують стабільну і співставну здатність до генералізації, проте ансамблеві та нейромережеві підходи залишаються більш гнучкими для виявлення складних патернів зносу.

4. Обґрунтовано перехід до проактивного обслуговування на основі мінімізації хибно-негативних прогнозів (False Negatives). Визначено, що критичним критерієм якості розробленої системи машинного навчання є не лише максимізація загальних метрик (F1-score та MCC), а зведення до мінімуму пропуску реальних дефектів (високий показник Recall). Це є ключовим фактором безпеки експлуатації ТЗ, що дозволяє виявляти аномалії до появи зовнішніх ознак відмови. Як перспективний напрям розвитку системи для

забезпечення аналітики з наднизькою затримкою у режимі реального часу, запропоновано розгортання оптимізованих моделей (Edge Computing) безпосередньо на периферійних пристроях.

РОЗДІЛ 4 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ МОНІТОРИНГУ І ПРОГНОЗУВАННЯ ТЕХНІЧНОГО СТАНУ АТЗ

4.1. Загальна концепція системи

Базова система зображена на рис.4.1. Загальна концепція уявляє в собі такі дії в системі:

- Транспортний засіб через сенсори генерує дані.
- OBD II сканер збирає дані з транспортного засобу та передає у сховище даних
- Внутрішній сервер виступає в ролі посередника, передаючи дані зі сховища даних до бази даних
- База даних керує як новими, так і історичними даними, які обробляються модулем аналізу
- На основі даних аналізу генерується результат прогнозування

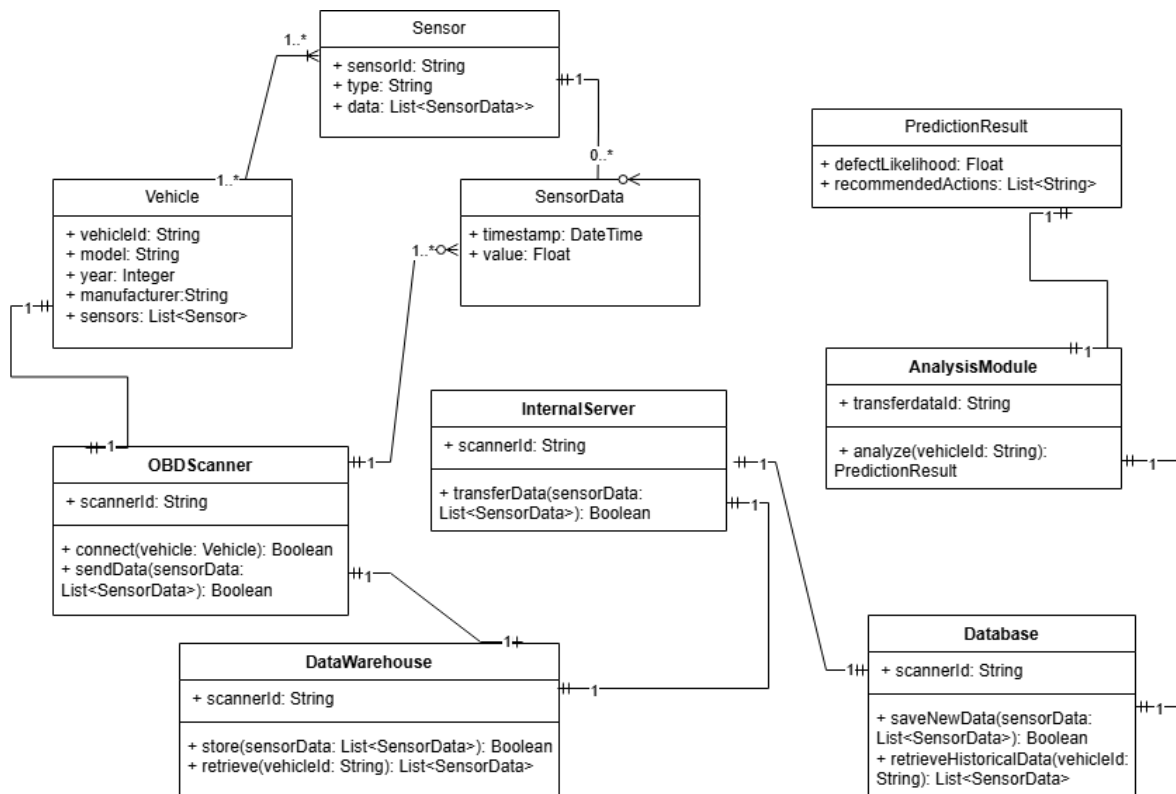


Рисунок 4.1 – Загальна діаграма класів системи

На схемі також зображено класи та методи, завдяки яким система буде функціонувати. Розглянемо кожен окремо:

- для класу *Vehicle* методи відсутні, однак цей клас є сутністю, яка містить атрибути (наприклад, унікальний номер ТЗ, модель, рік випуску, виробник(бренд ТЗ) список сенсорів або датчиків, які містить ТЗ);

- для класу *Sensor* методи також відсутні. Цей клас описує всі сенсори, які зберігають інформацію про себе та свої дані (для бази даних кожен сенсор повинен мати свій унікальний номер, тип даних та одиниці вимірювання та самі дані);

- для класу *SensorData* методів нема, так як клас виконує роль сховища для конкретних даних, таких як значення сенсора (*value*) та час зчитування (*timestamp*);

- клас *OBDScanner* має два методи: метод збереження даних сенсорів у сховищі та метод отримання даних сенсорів для конкретного ТЗ зі сховища;

- клас *InternalServer* за рахунок використання внутрішніх методів пересилає дані з одного етапу обробки на наступний (зі сховища даних до бази даних);

- у класу *Database* методи зберігають нові дані з сенсорів у базі та отримують історичні дані для конкретного транспортного засобу;

- клас *AnalysisModule* виконує аналіз даних для ТЗ та створює результат прогнозу;

- клас *PredictionResult* є результатом роботи модулю аналізу. Даний клас містить атрибути у вигляді ймовірності поломки та список рекомендацій.

Запропонована архітектура побудована за модульним принципом, що забезпечує незалежне функціонування окремих компонентів системи та можливість їх подальшого розвитку без зміни загальної структури. Такий підхід дозволяє масштабувати окремі компоненти відповідно до інтенсивності потоків даних, спрощує інтеграцію нових алгоритмів аналізу та забезпечує підвищення

надійності функціонування системи при зростанні кількості транспортних засобів.

Нижче представлена діаграма класів для *AnalysisModule* та побудуємо діаграму зв'язків для даного модулю. На рис.4.2. представлено діаграму функціонування нейронної мережі в модулі аналізу.

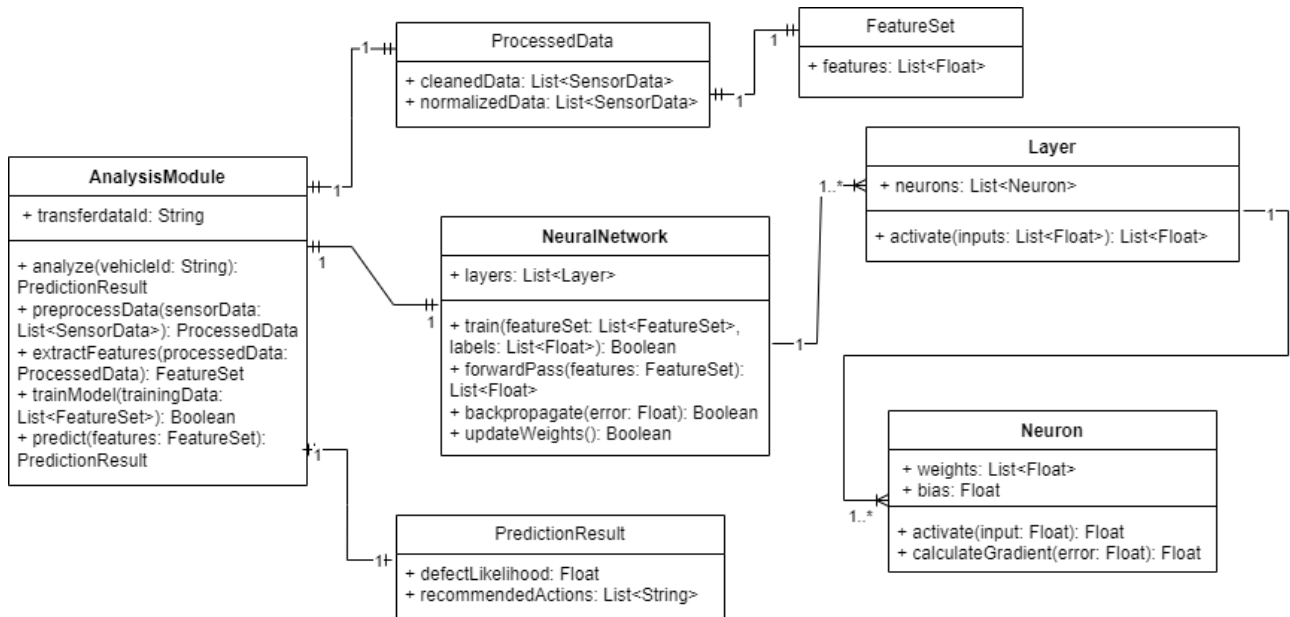


Рисунок 4.2 – Діаграма класів для блоку *AnalysisModule*

Дані з сенсорів зберігаються в базі даних та загалом у сховищі даних. Окрім стандартних методів передачі інформації від однієї системи в іншу необхідно використати додаткові перевірки даних у вигляді коректності отриманих даних, реєстрації даних за конкретним ТЗ, організацію цих даних за новизною (поточні та історичні дані), перевірки цілісності та консистенції даних та створення резервної копії даних.

При потраплянні в систему дані з сенсорів проходять очищення, нормалізацію та перетворюються на набори ознак. На основі введених ззовні даних нейронна мережа тренується на основі ознак із відповідними мітками (наприклад, наявність або відсутність дефекту). Підготовлені ознаки подаються

на вхід мережі, яка представляє собою структуру шарів та нейронів з методами активації, обчислення градієнтів та оновлення ваг.

При розширенні системи на рис. 4.3 можемо побачити як взаємодіють між собою компоненти на загальній схемі активності. Сканер OBD-II в режимі реального часу зчитує дані з датчиків, зокрема показники роботи двигуна, коди несправностей, температури компонентів та інші. Внутрішній препроцесор перевіряє дані на цілісність та достовірність, попередньо відфільтровуючи очевидні помилки або відсутні значення. Якщо дані недостовірні або не повні, в журнал реєструється помилка, і дані або повторно запитуються у автомобіля, або позначаються для ручного перегляду, або позначаються на видалення. При умові «Так» проходячи далі виконується перевірка цілісності та наявності всіх очікуваних даних. Також проходить перевірка показників датчиків на предмет відповідності очікуваним робочим діапазнам. Нормалізатор даних конвертує сирі дані датчика в стандартизований формат для подальшої обробки. Оброблені дані надсилаються на внутрішній сервер для зберігання та аналізу. Разом з цим, починається паралельна обробка даних.

Під час зберігання дані також передаються до сховища даних для довготривалого зберігання. Внутрішній сервер здійснює зберігання оброблених і перевірених даних на сервері бази даних для негайного доступу до них. При збереженні даних, сервер організовує дані за категоріями (наприклад, «поточні» та «історичні»). Час від часу сервер бази даних синхронізується з сервером сховища даних, забезпечуючи доступність історичних даних для проведення довгострокового аналітичного дослідження динаміки. Якщо ж сервер в момент підключення виявляється недоступний, то запис даних залишається в черзі для обробки та адміністратором перевіряються журнали (логи) серверу для знаходження помилки. Разом з цим система резервного копіювання надсилає копії критично важливих даних до хмарного сховища для аварійного

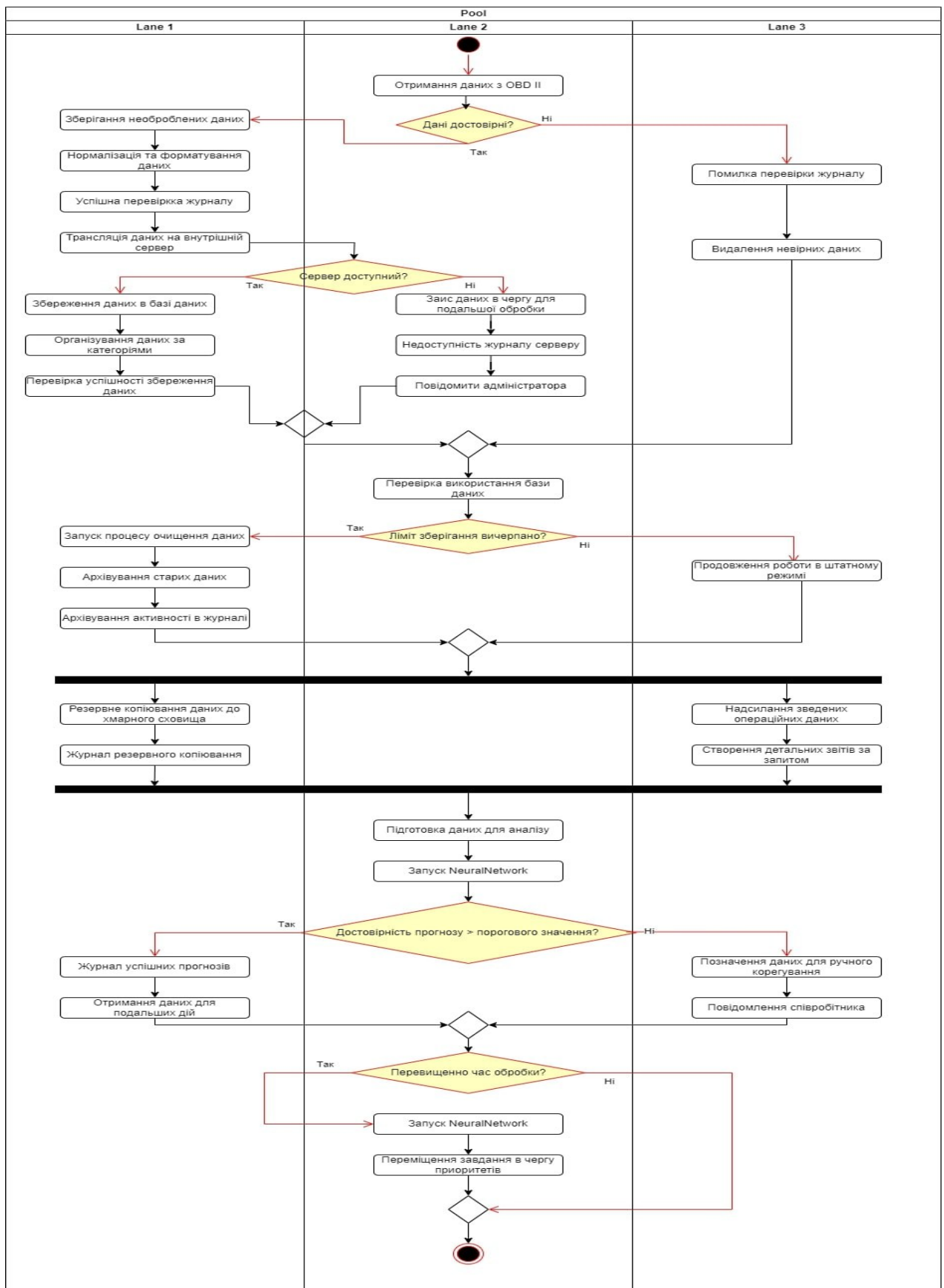


Рисунок 4.3 – Діаграма активності системи

відновлення. Сховище та база даних перевіряють інформацію та видаляють надлишкові або дуже застарілі записи. Всі ці активності архівуються в журналі для можливого подальшого статистичного аналізу.

Дані, історичні та поточні, проходять до нейронної мережі в реальному часі з сервера сховища даних. Модель машинного навчання аналізує структуру цих даних для виявлення закономірностей та виявлені аномалій:

- поява нових дефектів;
- нестандартна поведінка датчиків.

Для сучасних платформ незамінними є зовнішні бібліотеки, які дозволяють реалізувати вже готові функції, такі як машинне навчання, робота з базами даних та обробка даних. Однак їхня інтеграція передбачає необхідність чіткого архітектурного планування для забезпечення сумісності системи, відмовостійкості, масштабованості та дотримання вимог безпеки. Відповідно, для коректної роботи машинного навчання потрібно підключити відповідні бібліотеки (рис. 4.4), які використовують алгоритми та моделі машинного навчання для прогностичного аналізу (наприклад, TensorFlow [190], PyTorch [191], scikit-learn [192], ONNX Runtime [193] та інші). Це дозволить тренувати індивідуальні моделі, використовуючи різні набори даних та підтримувати зворотнє навчання для покращення для конкретної області. В систему додаються процеси, такі як мікросервіси, що працюють у контейнерах (наприклад, Docker [194]). Для цього використовується спеціальний сервер виведення (наприклад, NVIDIA Triton [195], TensorFlow Serving [190, 196]). Це дозволяє зменшити кількість дублікатів і забезпечити узгоджену взаємодію між модулями, а також дозволяє використовувати інтерфейси API для виведення в реальному часі або пакетного формування висновків. Водночас, з метою проведення аналітики в реальному часі в умовах низьких затримок, легкі моделі можна розгнати на периферійних пристроях (наприклад, OBD-II обладнання з вбудованим TensorFlow Lite [196-198]). Бібліотеки машинного навчання легко

поєднується з бібліотекою обробки даних для попередньої обробки даних, що забезпечує нормалізацію та готовність цих даних до аналізу. Крім того, бібліотека взаємодіє з модулем очищення даних для забезпечення якості вхідних даних, що прямо впливає на точність моделі.

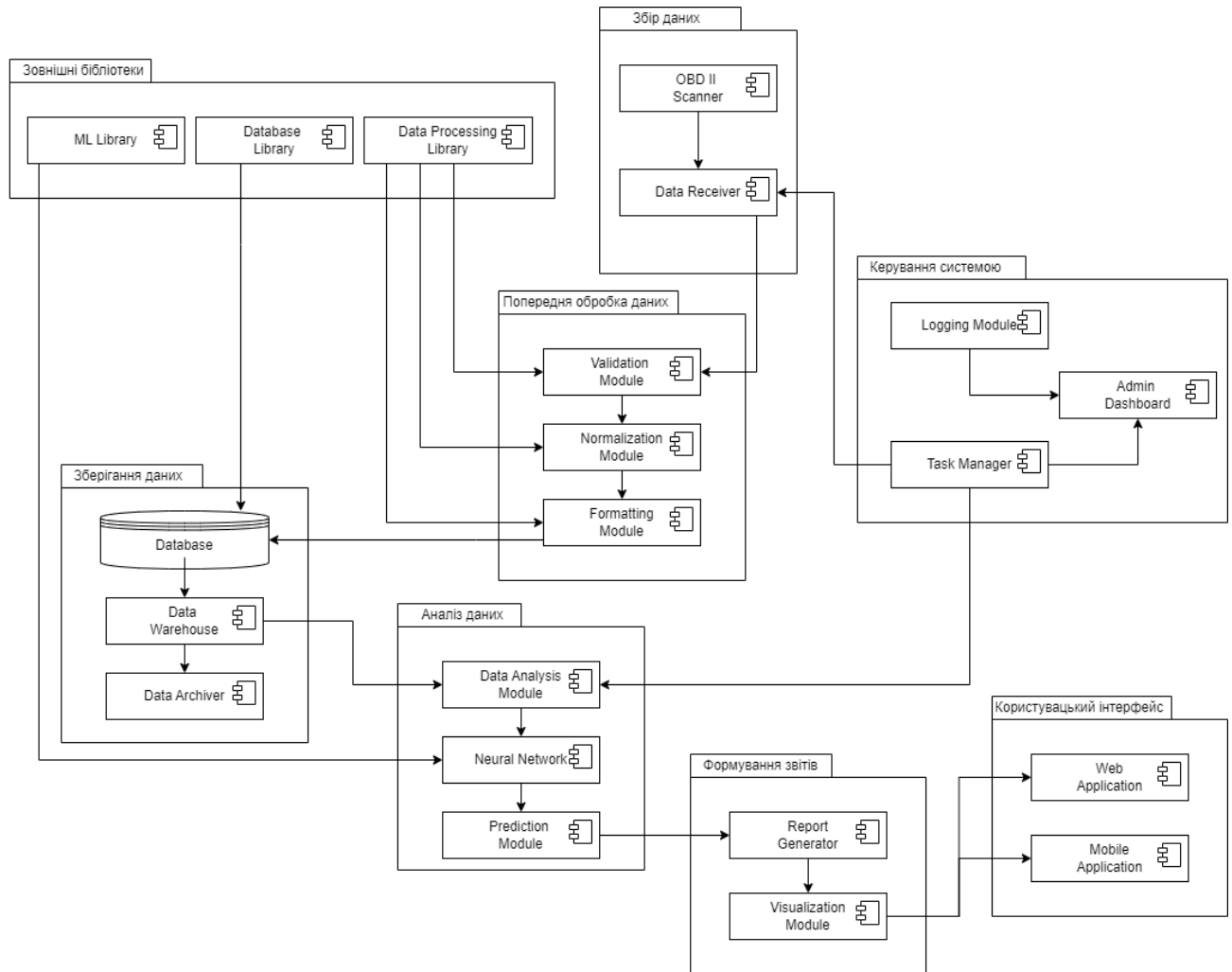


Рисунок 4.4 – Діаграма компонентів системи

Паралельно з цим працює бібліотека баз даних, яка полегшує управління операціями з базами даних, що забезпечує значну пропускну здатність і низьку затримку при виконанні CRUD-операцій. Доцільно для цього використовувати HikariCP [199] щоб ефективно керувати з'єднаннями з базами даних шляхом їх багаторазового використання, знижуючи навантаження, спричинене частим відкриттям та закриттям з'єднань. За його допомогою можна використовувати

одночасні 30-50 з'єднань. Завдяки бібліотеці забезпечується можливість виконання операцій запису до декількох таблиць (наприклад, вставка необроблених даних OBD-II та метаданих логування) як єдиної транзакції.

Бібліотека обробки даних зосереджена на підготовці, очищенні та перетворенні необроблених даних OBD-II у формат, придатний для їх зберігання та аналізу. Apache Spark обробляє великі обсяги історичних даних OBD-II. При пакетній обробці відсіювання недійсних записів, агрегування даних з датчиків та нормалізацію одиниць виміру. Також відбувається видалення записів з відсутніми критично важливими полями (наприклад, ідентифікатор транспортного засобу або мітка часу) та перевірка діапазонів. При нормалізації стандартизуються одиниці вимірювання та відбувається перетворення різних протоколів OBD-II (наприклад, ISO 15765-4, SAE J1850) в уніфікований формат JSON. Також разом з цим відбуваються обчислення окремих характеристик, такі як паливна ефективність та інші, групуються за ідентифікатором транспортного засобу та узагальнюються метрики. Далі попередньо оброблені дані надсилаються до валідатора даних для додаткових перевірок. І в кінці очищені дані спрямовуються до бази даних для довготривалого зберігання. Apache Spark та Kafka ефективно використовують розбиття даних на блоки для паралельного виконання. Наприклад, дані про транспортні засоби розбиті на частини за ідентифікатором *vehicle_id* для класу *Vehicle* (рис. 4.1). Такі завдання, як фільтрація та групування, виконуються паралельно на всіх частинах, що значно скорочує час обробки.

4.2. Компоненти системи

Більш детально схема компонентів зображена на рис. 4.5. Як було сказано вище, OBD II сканер використовує різні протоколи. Одразу після отримання

даних валідатор даних фільтрує вхідні дані, щоб переконатися, що вони відповідають вимогам якості та цілісності.

Дані перевіряються за рахунок використання CRC або хеш-методів [200]. Відсіює неповні або помилкові дані та пересилає лише достовірну інформацію на наступні компоненти для нормалізації. Нормалізовані дані надсилаються до сховища або компонентів аналізу. Даний модуль нормалізує одиниці виміру, мітки часу або кодування. Також є можливість інтегрувати API, що дозволить безперешкодно підключати зовнішні системи, такі як інструменти управління сервісом або сторонні аналітичні платформи. Разом з цим відбувається шифрування даних для безпечної передачі та зберігання методами AES, RSA або SSL/TLS шифрування [201].

Разом з цим модуль реєстрації помилок (Error Logging) здійснює збір та управління журналами, пов'язаними з системними помилками, виключеннями та попередженнями. Це критично важливий компонент для усунення помилок, моніторингу та аудиту. Модуль об'єднує журнали з будь-яких модулів у централізовану систему для полегшення доступу та аналізу та структурує логи в JSON документ та автоматично архівує або видаляє старі журнали на основі попередньо визначених періодів зберігання. Підтримує різні рівні журналів (наприклад, DEBUG, INFO, WARNING, ERROR, CRITICAL) для визначення пріоритетності відповідей. Також підтримується синхронізація з системою сповіщень, що дозволяє запускати сповіщення при виникненні критичних помилок. Для реалізації такого модулю потрібно додати журнали, які зберігаються в масштабованих системах, таких як Elasticsearch [202], або хмарних сервісах, таких як AWS CloudWatch [17]. Журнали мають метадані, такі як назва модуля, мітка часу, серйозність та унікальний код помилки, для швидкої фільтрації та аналізу. Система моніторингу (Monitoring Module) надсилає логи з даними, пов'язаними з продуктивністю (наприклад, попередження про високе використання пам'яті).

Модуль контролю виконання завдань (Audit Trail Module) зберігає дії користувачів, що призводять до помилок, а системна інформаційна панель відображає динаміку помилок та збирає аналітику для розробників.

Паралельно цим діям, планувальний модуль (Scheduler Module) автоматизує виконання регулярних завдань, таких як технічне обслуговування бази даних, резервне копіювання або синхронізація даних, гарантує, що завдання, які залежать одне від одного, виконуються в правильному порядку. Це дозволяє планувати разові, циклічні або ж умовні завдання. За рахунок використання цього блоку можливо розподілити пріоритети між завданнями, забезпечуючи першочергове виконання критично важливих операцій та динамічно регулювати графіки на основі завантаженості системи або стану завершення завдань. Це реалізується за рахунок використання Apache Airflow [203], або Kubernetes CronJobs [204] для забезпечення безвідмовної роботи та відстеження запланованих завдань, які зберігаються в реляційній базі даних або в порядку черги завдань. Далі йде підключення до модулю обслуговування системи (System Maintenance Module), де виконується оновлення або оптимізація бази даних через попередньо визначені проміжки часу. Регулятор навантаження спрямовує запити запланованих завдань між кількома користувачами, щоб запобігти виникненню «вузьких місць» та передає цю інформацію на внутрішній сервер.

В систему також додається модуль відмовостійкості (Monitoring System), який буде відповідати за надійність та високу доступність системи, спрямовуючи завдання та операції на резервні ресурси у випадку збою. Модуль знаходить збої і спрямовує дані або завдання на резервні сервери або сервіси та створює резервні копії критично важливих компонентів. Використовувати можна хмарні функції, такі як AWS RDS Multi-AZ [205, 206], який використовує періодичні опитування системи або перевірку працездатності кінцевих точок для виявлення збоїв та гарантує, що резервні копії завжди залишаються

сумісними з основною системою, використовуючи стратегії за рахунок використання технологій відзеркалення (наприклад, Apache Kafka mirror maker) [207, 208]. Цей модуль інтегрується до модуля планування, що дає змогу завершити завдання, що виконуються під час аварійного перемикання, без втрати даних та працює разом з модулем балансування (Load Balancer) для переадресації навантаження під час сценаріїв аварійного перемикання.

Модуль контролю виконання завдань реєструє та відстежує всі дії користувачів з ТЗ та системою для відповідності вимогам, можливості відстеження та проведення аудиту безпеки. Журнал активності модулю фіксує спроби входу в систему, зміни конфігурації та звернення до системи, а також події, пов'язані з доступом до даних. Система виявляє і повідомляє про підозрілі дії, такі як несанкціонований доступ до даних або підвищення прав доступу та запобігає фальсифікації або знищенню журналів аудиту. При підключені до модуля безпеки сповіщення для адміністраторів про порушення правил або порушення безпеки приходять автоматично. Та модуль реєстрації помилок відстежує стани ТЗ з системними помилками для аналізу першопричин.

Як було вказано вище, модуль технічного обслуговування системи керує виконанням регулярних завдань з технічного обслуговування та оптимізації системи, забезпечуючи її безперебійну роботу та довговічність. Це відбувається за рахунок використання діагностичних скриптів для ідентифікації та вирішення потенційних проблем до того, як вони загостряться. Для автоматизації можна запустити на фреймворках Puppet [209] або Terraform для обслуговування інфраструктури [210, 211]. Відповідно, планувальний модуль запускає періодичні завдання з обслуговування, а система моніторингу позначає елементи які потребують уваги.

Інформація, яка була отримана від всіх систем потрапляє в модуль зберігання даних (Database), сховище для структурованих та нормалізованих даних.— базу даних. Модуль представляє собою реляційну базу даних, з

використанням PostgreSQL [212] або MySQL [213], з оптимізованими схемами для даних OBD-II. Структурна схема бази даних зі зв'язками та індексами наведена нижче на рис. 4.6:

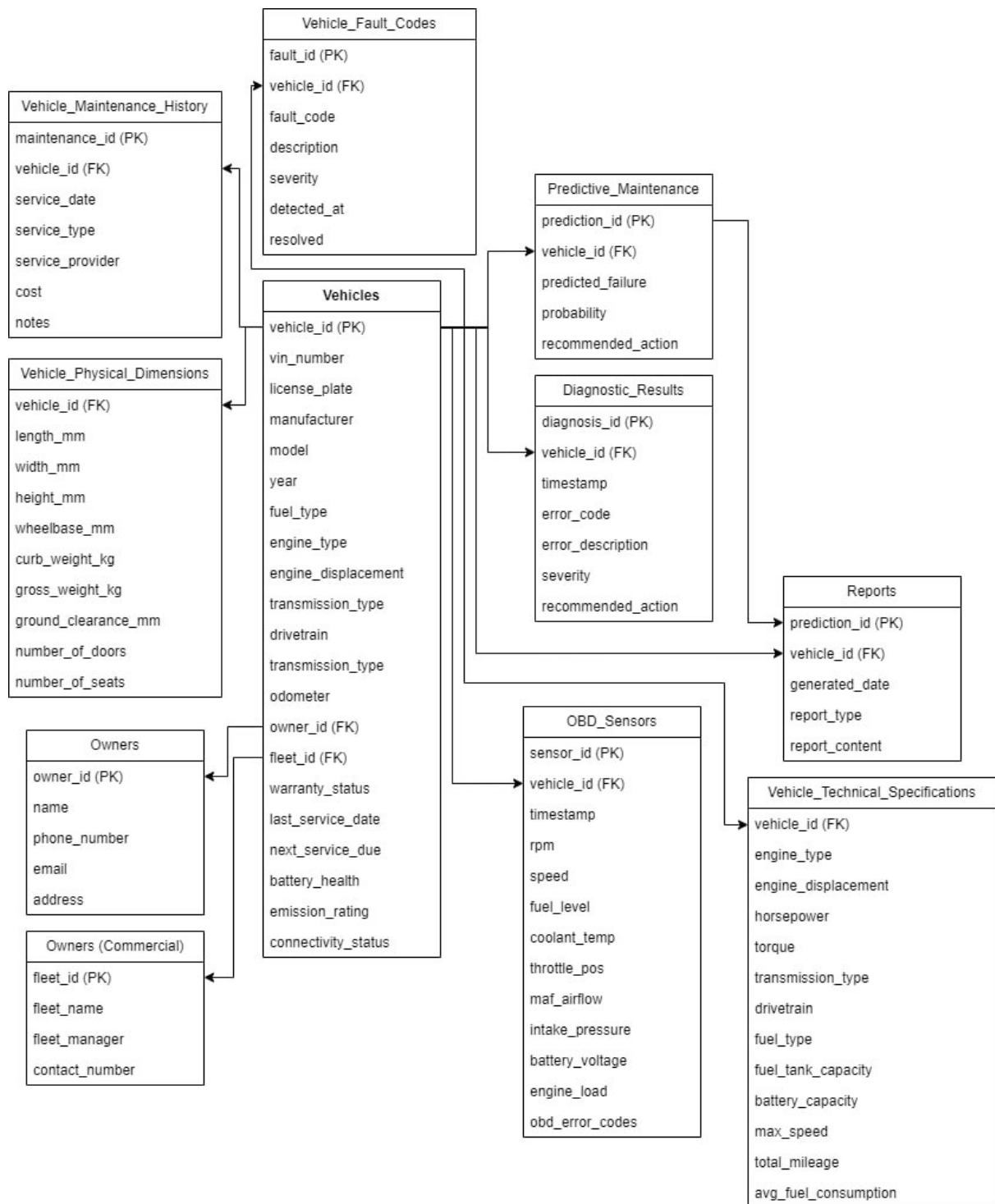


Рисунок 4.6 – Базова структура бази даних

4.3. Структура зв'язків блоків даних

Данна схема розроблена для представлення даних транспортних засобів, телеметричних записів, інформації про власників та записи про технічне обслуговування, кожна з яких представлена як окрема об'єктна сутність. Відповідно, схема розроблена на основі реляційних схем, де нормалізація є ключовим принципом. Альтернативна схема для noSQL передбачає денормалізований підхід, що передбачає розміщення пов'язаних даних у документах, якщо це необхідно. Проте, системи транспортних засобів генерують значну кількість різноманітних даних, починаючи від показань датчиків у реальному часі та закінчуючи довготривалими реєстрами про технічне обслуговування. Стандартні реляційні бази даних часто стикаються з проблемами масштабованості і швидкодії в таких умовах через їх фіксовані схеми. З цього виходить, що використання noSQL більше відповідає вимогам роботи з динамічними, швидкозмінними потоками даних. Отримана модель буде зосереджена на колекціях, вбудованих документах і посиланнях. Така можливість відповідає реальним представленням даних, що дозволяє безпосередньо моделювати складні сутності. Основними колекціями даних у схемі будуть:

- колекція транспортних засобів, яка складається з ключових даних про кожний транспортний засіб, в тому числі вбудовані записи про власників та записи про технічне обслуговування;
- колекція журналів телеметрії. колекція динамічних спостережень, що містить дані з датчиків у реальному часі та історичні дані;
- колекція власників. містить інформацію про власників транспортних засобів;

- колекція записів про технічне обслуговування. використовується для масштабування зберігає докладну звітність про технічне обслуговування транспортних засобів, відокремлену від основної колекції транспортних засобів.

Транспортні засоби (Vehicle) будуть представлені у вигляді колекції, де кожен транспортний засіб є документом. На рис.4.7 наведено приклад JSON-документу для умовного транспортного засобу. Колекція транспортних засобів є центральною сутністю, призначеною для зберігання всіх стаціонарних і тимчасових атрибутів транспортного засобу. Кожен документ являє собою один транспортний засіб, зберігаючи такі деталі, як ідентифікаційний номер транспортного засобу (VIN), виробник, модель, технічні характеристики двигуна та історію володіння. Інтеграція записів про технічне обслуговування в один документ про транспортний засіб зменшує потребу в об'єднаннях при запиті даних про конкретний транспортний засіб. Однак, для того щоб забезпечити можливість масштабування, також створюється і окрема колекція даних про технічне обслуговування. Для посилання на власника використовується *user_id*, оскільки інформація про користувача може бути спільною для декількох транспортних засобів.

Дані телеметрії, що характеризуються високою швидкістю та великими розмірами, вимагають спеціалізованого збору часових даних. Кожен запис у журналі телеметрії зберігається як окремий документ, щоб полегшити високочастотні операції запису та забезпечити ефективні хронологічні пошуки.

До схеми (рис. 4.8) входять індекси за ідентифікатором *vehicle_id* та міткою часу для того, щоб максимально покращити роботу запитів, які отримують телеметричні дані в певних часових діапазонах.

Для керування інформацією про власників окремо від даних про транспортні засоби формується зовнішня колекція власників (рис. 4.9). Такий метод дозволяє врахувати сценарії, коли один і той самий власник має декілька транспортних засобів.

```

1  {
2    "_id": "unique_vehicle_id",
3    "vin": "1HGCM82633A123456",
4    "manufacturer": "Toyota",
5    "model": "Camry",
6    "year": 2021,
7    "engine": {
8      "fuel_type": "Diesel",
9      "displacement": 1.8,
10     "power": 140
11   },
12   "owner": {
13     "owner_id": "owner_unique_id",
14     "name": "Andrii Shevchenko",
15     "contact": "andriishe@gmail.com"
16   },
17   "service_history": [
18     {
19       "service_id": "s1001",
20       "date": "2024-08-15",
21       "description": "Oil Change",
22       "odometer": 150000,
23       "cost": 1200
24     },
25     {
26       "service_id": "s1002",
27       "date": "2024-10-20",
28       "description": "Tire Rotation",
29       "odometer": 180000,
30       "cost": 1700
31     }
32   ],
33   "active_faults": [
34     {
35       "code": "P0420",
36       "_id": "P0420",
37       "description": "Catalyst System Efficiency Below Threshold",
38       "severity": "High",
39       "causes": [
40         "Faulty Oxygen Sensor",
41         "Damaged Catalytic Converter"
42       ],
43       "solutions": [
44         "Replace Catalytic Converter",
45         "Inspect Exhaust System"
46       ],
47       "date_detected": "2024-12-01"
48     }
49   ]
50 }

```

Рисунок 4.7 – Приклад JSON-документу для транспортного засобу

```

1  {
2    "_id": "unique_telemetry_id",
3    "vehicle_id": "unique_vehicle_id",
4    "timestamp": "2024-06-15T10:23:45Z",
5    "speed": 65,
6    "rpm": 2000,
7    "fuel_level": 50,
8    "engine_temp": 90
9  }

```

Рисунок 4.8 – Приклад JSON-документу телеметрії для транспортного засобу

```

1  {
2    "_id": "owner_unique_id",
3    "name": "Andrii Shevchenko",
4    "address": "Chornovola 55, Lviv",
5    "phone": "095-0000000",
6    "vehicles": ["vehicle_id_1", "vehicle_id_2"]
7  }

```

Рисунок 4.9 – Приклад JSON-документу власників для транспортного засобу

Система збирання даних про технічне обслуговування (рис. 4.10) розділяє дані про технічне обслуговування та дані про транспортні засоби. Це забезпечує масштабованість, оскільки з часом збільшується кількість записів.

```

1  {
2    "_id": "unique_record_id",
3    "vehicle_id": "unique_vehicle_id",
4    "date": "2024-12-01",
5    "service": "Oil Change",
6    "odometer": 150000,
7    "cost": 800
8  }

```

Рисунок 4.10 – Приклад JSON-документу ТО для транспортного засобу

Використання документоорієнтованої моделі даних пов'язано з неоднорідністю телеметричної інформації, що надходить від транспортних засобів різних виробників та комплектацій. Такий підхід дозволяє зберігати

змінний набір параметрів без модифікації структури бази даних, що значно спрощує інтеграцію нових типів датчиків та забезпечує горизонтальне масштабування системи при збільшенні обсягів інформації.

Таблиця 4.1 – Поле Vehicles бази даних

Vehicles		
Field Name	Data Type	Description
vehicle_id (PK)	INT	Унікальний ідентифікатор для кожного транспортного засобу
vin_number	VARCHAR(20)	Унікальний ідентифікаційний номер транспортного засобу
license_plate	VARCHAR(15)	Номерний знак транспортного засобу
manufacturer	VARCHAR(50)	Виробник автомобіля
model	VARCHAR(50)	Назва моделі
year	INT	Рік випуску
fuel_type	VARCHAR(20)	Тип пального, що використовується
engine_type	VARCHAR(50)	Тип двигуна
engine_displacement	FLOAT	Об'єм двигуна в літрах
transmission_type	VARCHAR(20)	Тип трансмісії
drivetrain	VARCHAR(20)	FWD, AWD, RWD
odometer	INT	Поточний пробіг автомобіля
owner_id (FK)	INT	Власник транспортного засобу

Продовження таблиці 4.1

Vehicles		
Field Name	Data Type	Description
fleet_id (FK)	INT	Зовнішній ключ для систем управління автопарком
warranty_status	VARCHAR(20)	Статус гарантії
last_service_date	DATE	Дата останнього обслуговування транспортного засобу
next_service_due	DATE	Рекомендована дата наступного сервісного обслуговування
battery_health	VARCHAR(20)	Стан акумулятора
connectivity_status	VARCHAR(20)	Статус підключених автомобільних сервісів

Таблиця 4.2 – Поле Owners бази даних

Owners		
Field Name	Data Type	Description
owner_id (PK)	INT	Унікальний ідентифікатор для кожного власника
name	VARCHAR(100)	Повне ім'я власника
phone_number	VARCHAR(20)	Контактний номер телефону
email	VARCHAR(100)	Адреса електронної пошти
address	TEXT	Адреса власника

Таблиця 4.3 – Поле Fleet Management бази даних

Fleet Management		
Field Name	Data Type	Description
fleet_id (PK)	INT	Унікальний ідентифікатор для кожного АТП
fleet_name	VARCHAR(50)	Назва автопарку
fleet_manager	VARCHAR(100)	Ім'я відповідального менеджера автопарку
contact_number	VARCHAR(20)	Контактний номер

Таблиця 4.4 – Поле Diagnostic_Results бази даних

Diagnostic_Results		
Field Name	Data Type	Description
diagnosis_id (PK)	INT	Унікальний ідентифікатор для кожного статусу
vehicle_id (FK)	INT	Зовнішній ключ, що посиляється на таблицю ТЗ
timestamp	TIMESTAMP	Час, коли було записано результат діагностики
error_code	VARCHAR(10)	Стандартний код помилки OBD II
error_description	TEXT	Опис коду помилки
severity	VARCHAR(20)	Серйозність проблеми (низька, середня, висока)
recommended_action	TEXT	Рекомендований ремонт або технічне обслуговування

Таблиця 4.5 –Поле OBD_Sensors бази даних

OBD_Sensors		
Field Name	Data Type	Description
sensor_id (PK)	INT	Унікальний ідентифікатор для кожного датчика
vehicle_id (FK)	INT	Прив'язує дані датчика до транспортного засобу
timestamp	TIMESTAMP	Час збору даних
rpm	INT	Оберти двигуна за хвилину
speed	INT	Швидкість автомобіля у км/год
fuel_level	FLOAT	Залишок палива у відсотках
coolant_temp	FLOAT	Температура охолоджувальної рідини в градусах Цельсія
throttle_pos	FLOAT	Відсоток положення дросельної заслонки
maf_airflow	FLOAT	Масова витрата повітря (г/с)
intake_pressure	FLOAT	Тиск у впускному колекторі
battery_voltage	FLOAT	Напруга акумулятора автомобіля
engine_load	FLOAT	Відсоток завантаження двигуна
obd_error_codes	VARCHAR(255)	Список виявлених кодів помилок OBD

Таблиця 4.6 – Поле Predictive_Maintenance бази даних

Predictive_Maintenance		
Field Name	Data Type	Description
prediction_id (PK)	INT	Унікальний ідентифікатор для кожного прогнозу
vehicle_id (FK)	INT	Зовнішній ключ, що посиляється на транспортний засіб
predicted_failure	VARCHAR(255)	Ймовірна відмова компонента
probability	FLOAT	Оцінка ймовірності від AI-моделі
recommended_action	TEXT	Рекомендована профілактична дія

Таблиця 4.7 – Поле Table: Reports бази даних

Reports		
Field Name	Data Type	Description
report_id (PK)	INT	Унікальний ідентифікатор звіту
vehicle_id (FK)	INT	Транспортний засіб, пов'язаний зі звітом
generated_date	TIMESTAMP	Час створення звіту
report_type	VARCHAR(50)	Тип звіту (продуктивність, несправність тощо)
report_content	TEXT	Звіт у форматі JSON/XML/CSV

Таблиця 4.8 – Поле Vehicle_Technical_Specifications бази даних

Vehicle_Technical_Specifications		
Field Name	Data Type	Description
vehicle_id (FK)	INT	Зовнішній ключ до таблиці Vehicles
engine_type	VARCHAR(50)	Бензиновий, дизельний, гібридний, електричний
engine_displacement	FLOAT	Об'єм двигуна в куб.см
horsepower	FLOAT	Максимальна вихідна потужність (к.с.)
torque	FLOAT	Максимальний крутний момент (Нм)
transmission_type	VARCHAR(20)	Механічна, Автоматична
drivetrain	VARCHAR(20)	FWD, RWD, AWD
fuel_type	VARCHAR(20)	Бензин, Дизель, Електродвигун, Гібрид
fuel_tank_capacity	FLOAT	Об'єм паливного бака (літри)
battery_capacity	FLOAT	Для електричних/гібридних транспортних засобів (кВт-год)
max_speed	FLOAT	Максимальна швидкість (км/год)
acceleration_0_100	FLOAT	Час досягнення 100 км/год (секунди)
emission_standard	VARCHAR(10)	Стандарт викидів

Таблиця 4.9 – Поле Vehicle_Technical_Specifications бази даних

Vehicle_Physical_Dimensions		
Field Name	Data Type	Description
vehicle_id (FK)	INT	Зовнішній ключ до таблиці Vehicles
length_mm	FLOAT	Довжина транспортного засобу (мм)
width_mm	FLOAT	Ширина транспортного засобу (мм)
height_mm	FLOAT	Висота автомобіля (мм)
wheelbase_mm	FLOAT	Відстань між передніми та задніми колесами (мм)
curb_weight_kg	FLOAT	Вага без пасажирів (кг)
gross_weight_kg	FLOAT	Максимальна дозволена вага (кг)
ground_clearance_mm	FLOAT	Відстань між шасі та землею (мм)
number_of_doors	INT	Кількість дверей
number_of_seats	INT	Кількість пасажирів, яку може вмістити

Таблиця 4.10 – Поле Vehicle_Maintenance_History бази даних

Vehicle_Maintenance_History		
Field Name	Data Type	Description
maintenance_id (PK)	INT	Унікальний ідентифікатор запису
vehicle_id (FK)	INT	Зовнішній ключ до таблиці Vehicles
service_date	DATE	Дата, коли було виконано технічне обслуговування
service_type	VARCHAR(50)	Тип ремонту

Vehicle_Maintenance_History		
Field Name	Data Type	Description
service_provider	VARCHAR(50)	Назва майстерні/сервісного центру
cost	FLOAT	Вартість послуги
notes	TEXT	Додаткова інформація

Таблиця 4.11 – Поле Vehicle_Fault_Codes бази даних

Vehicle_Fault_Codes		
Field Name	Data Type	Description
fault_id (PK)	INT	Унікальний ідентифікатор запису
vehicle_id (FK)	INT	Зовнішній ключ до таблиці Vehicles
fault_code	VARCHAR(10)	Код OBD-II за стандартом
description	TEXT	Опис несправності
severity	VARCHAR(20)	Серйозність поломки
detected_at	TIMESTAMP	Коли було виявлено несправність
resolved	BOOLEAN	Чи було виправлено проблему

Інформація після бази даних (таблиця 4.1-4.11) йде на сховище даних (DataWarehouse), яке в свою чергу зберігає агреговані та історичні дані для звітності та аналізу. Це централізоване сховище, яке інтегруватиме дані з численних джерел (наприклад, датчиків автомобіля, діагностики, журналів технічного обслуговування, звітів користувачів). Таке сховище буде підтримувати історичний аналіз, прогнозоване моделювання та звітність, має архітектуру з рівнями зберігання, інтеграції та представлення даних, що забезпечує оптимізоване накопичення, швидкість запитів та ефективну

аналітику. У модулі бази даних та у сховищі даних використовується індексування та розбиття на розділи:

- покращення продуктивності запитів;
- оптимізація використання сховища;
- зменшення навантаження на систему за рахунок отримання тільки необхідних даних.

Розділення забезпечує горизонтальне масштабування між декількома вузлами. Індксація забезпечує ефективність вертикального масштабування за рахунок зменшення обсягу файлового вводу/виводу. Індокси регулярно переглядаються, щоб запобігти їх фрагментації. Відповідно, це дозволяє швидко архівувати та видаляти старі дані. Для забезпечення цілісності даних буде реалізовано конвеєр ETL за рахунок використання PostgreSQL (OLAP) [212], Apache Hive (Big Data) [214], AWS Redshift (хмарне сховище) [215]:

- E - Дані зі сканерів OBD II, баз даних транспортних засобів, хмарних сховищ;
- T - Нормалізація, очищення від недостовірних даних, формування історичних даних;
- L - Зберігання даних в просторових базах даних з індексацією та розбиттям на розділи.

Використання ETL-конвеєра обумовлено необхідністю інтеграції даних із різномірних джерел, що характеризуються різною структурою, частотою оновлення та рівнем достовірності. Попередня очистка, нормалізація та узгодження форматів даних дозволяють зменшити вплив шумів, пропущених значень і дублювання інформації, що безпосередньо підвищує якість побудови моделей машинного навчання.

Вибір PostgreSQL обумовлений підтримкою складних SQL-запитів, транзакційності та високою надійністю зберігання структурованих даних. Apache Hive використовується для обробки великих масивів історичної

інформації у середовищі Big Data, тоді як AWS Redshift забезпечує масштабоване хмарне зберігання та високопродуктивне виконання аналітичних запитів. Комплексне використання зазначених технологій дозволяє ефективно обробляти як потокові, так і історичні дані транспортних засобів.

Далі інформація попадає в модуль архівування (Data Compression), де архіватор здійснює стиснення та кодування старих даних для довготривалого зберігання. За рахунок використання алгоритмів zstd для мінімізації місця на диску зменшується обсяг наборів даних у реальному часі та пакетних даних перед обробкою або передачею. Це дозволяє готувати дані для зберігання в хмарному сховищі або для передачі через API.

Інформація з баз даних напряму направляється на модуль аналізу (Analysis Module). Модуль застосовує моделі машинного навчання для прогностичного аналізу. Цей модуль складається з декількох частин:

1. Суб-модуль попереднього опрацювання даних, де повторно відбувається процес виявлення пропущених значень, відхилень і невірних показань датчиків та відбувається об'єднання та узагальнення даних за різні проміжки часу;

2. Модуль машинного навчання та прогностичного аналізу. Система аналізує історичні коди несправностей і показники датчиків для визначення потенційних несправностей. Нейронний модуль аналізує показання датчиків OBD II, історичні діагностичні коди несправностей, журнали поведінки водія, GPS дані та умови навколишнього середовища, історія сервісного обслуговування та журнали ремонтів для виявлення закономірностей. Автокодер визначає незвичні відхилення в даних датчиків, наприклад відхилення в температурі, витраті палива та поведінці двигуна. Далі, рекурентні нейронні мережі виконують прогнозування відмов компонентів на основі послідовних показань датчиків, попередніх записів про технічне

обслуговування та упорядкується на основі потенційних ризиків виходу з ладу обладнання.

Особливістю запропонованого модуля аналізу є поєднання методів виявлення аномалій та прогнозування відмов. Автокодер використовується для автоматичного виявлення прихованих відхилень у телеметричних даних без необхідності попереднього маркування, а моделі машинного навчання виконують прогнозування технічного стану з урахуванням часової динаміки зміни параметрів транспортного засобу. Поєднуючи ці інструменти ми отримуємо можливість підвищити достовірність раннього виявлення потенційних несправностей.

Після модулю аналізу інформація передається в модуль звітності (Reporting Module), який напряду відповідає за створення структурованих звітів на основі проаналізованих даних. Інформаційна панель відображає в реальному часі інформацію по всім ТЗ, показує активні діагностичні коди несправностей. Також підсумовує історію обслуговування та заплановані роботи з технічного обслуговування з можливістю експорту звітів у різних форматах (PDF/CSV/Excel, інші), або надсилає звіти через API. Цей блок напряду взаємодіє з модулем візуалізації для отримання графічних даних в інтерфейсі користувача.

В кінці інформація потрапляє в інтерфейс користувача в якому відображаються звіти та забезпечується аналітика в реальному часі. Разом з цим реалізується система сповіщень, завдяки якій відбувається відправка сповіщень користувачам через різні канали зв'язку (наприклад, електронна пошта, SMS або пуш-сповіщення). Сповіщення відправляються на основі умов з модулю аналітики в реальному часі або з системи моніторингу.

Застосування такої архітектури дозволить скоротити час обробки телеметричних даних за рахунок використання індексування та горизонтального розділення даних, підвищити масштабованість системи при збільшенні

кількості транспортних засобів, а також забезпечити підвищення точності прогнозування несправностей завдяки комплексному використанню історичних даних, телеметрії та моделей машинного навчання. Конкретні значення зазначених показників визначаються експериментально під час апробації системи.

4.4. Оцінка ефективності системи

Оцінка ефективності розробленої розподіленої інформаційної системи базується на системі кількісних індикаторів, що математично відображають як якість попередньої обробки розріджених даних (LVD), так і предиктивну здатність моделей машинного навчання, а також їхній експлуатаційно-економічний вплив.

4.4.1. Оцінка якості обробки даних та моделей класифікації.

Для кількісної оцінки ефективності відбору корисних телеметричних ознак впроваджено статистичний критерій узгодженості Колмогорова-Смірнова. Цей індикатор розраховує максимальну відстань між емпіричними функціями розподілу справного та дефектного станів транспортного засобу:

$$D_n = \max |F_{need-repair}(x) - F_{no-repair}(x)| \quad (4.1)$$

Мінімізація втрат інформації під час використання алгоритмів пучкового пошуку контролюється через введення спеціального оціночного коефіцієнта обмеження вибірки (*constraintFactor*). Цей коефіцієнт дозволяє математично формалізувати залежність між кількістю відібраних параметрів та підсумковим розміром датасету:

$$n_{dataset} = n_{all} * constraintFactor^{n_{params}} \quad (4.2)$$

Головним критерієм кількісної оцінки моделей класифікації в умовах екстремального дисбалансу класів, притаманного автомобільному сервісу, виступає інтегральний коефіцієнт кореляції Метьюса (MCC) та гармонійне середнє (F1-score). Порівняльна оцінка ефективності базових алгоритмів на тестовій вибірці наведена в таблиці 4.12.

Таблиця 4.12 – Кількісні показники ефективності моделей на тестовій вибірці

Модель класифікації	Площа під ROC-кривою (AUC)	Гармонійне середнє (F1-score)	Коефіцієнт кореляції Метьюса (MCC)	Повнота виявлення (Recall)	Кількість хибно-негативних (FN)
Логістична регресія	0.702	0.500	0.544	0.333	2
Випадковий ліс (Random Forest)	0.643	0.561	0.094	0.632	2
Багатошаровий перцептрон (MLP)	0.750	0.732	0.420	0.737	1

За результатами кількісного аналізу Багатошаровий перцептрон (MLP) продемонстрував найвищу здатність до узагальнення. Застосування цієї архітектури дозволило досягти інтегрального показника $F1\text{-score} = 0.732$ та мінімізувати кількість небезпечних хибно-негативних прогнозів до $FN=1$.

4.4.2. Розрахунок експлуатаційно-економічних коефіцієнтів.

Для обґрунтування практичного впровадження розробленої системи на станціях технічного обслуговування (СТО) та в автотранспортних підприємствах розраховано кількісні показники експлуатаційно-економічної ефективності.

1. Коефіцієнт скорочення часу простою ($K_{downtime}$): Час реактивного простою ($T_{reactive}$) складається з очікування евакуатора (t_{tow}), ручної

діагностики (t_{diag}), очікування деталей (t_{parts}) та ремонту (t_{repair}). Для типової відмови комерційного транспорту: $t_{tow} = 3$ год, $t_{diag} = 2$ год, ($t_{parts} = 0$ год (деталі є на складі), $t_{repair} = 4$ год. Отже, $T_{reactive} = 9$ годин. Час прогностичного обслуговування ($T_{predictive}$) виключає евакуацію, а діагностика виконується автоматично системою ($t_{diag-auto} = 0.5$ год). Отже, $T_{predictive} = 4.5$ години.

$$K_{downtime} = \frac{T_{reactive} - T_{predictive}}{T_{reactive}} * 100\% = \frac{9 - 4.5}{9} * 100\% = 50\%$$

Таким чином, система скорочує цільовий час простою вдвічі.

2. Коефіцієнт оптимізації річних витрат на ТО та ремонт (K_{cost}):

Для об'єктивної економічної оцінки ефективності впровадження Big Data інфраструктури доцільно розраховувати вплив на загальні річні витрати (Total Cost of Ownership) для всього автопарку, а не лише ефект від запобігання окремим інцидентам. Як репрезентативну модель використано підприємство з парком у 100 комерційних транспортних засобів (ТЗ).

Для розрахунку застосовано метод експертних оцінок, що базується на усереднених ринкових цінах на послуги комерційного автосервісу та логістики в Україні станом на 2026 рік. У базовому сценарії враховується планове технічне обслуговування, статистична ймовірність непередбачуваних критичних відмов у дорозі (прийнято на рівні 10% парку на рік) та витрати на підтримку самої прогностичної ІТ-системи.

За традиційної реактивної стратегії загальні витрати ($C_{reactive_total}$) формуються з базового планового ТО для всіх автомобілів (умовно 15 000 ₴ на одне авто в рік) та витрат на ліквідацію аварій. Вартість реактивного усунення критичної відмови двигуна ($C_{reactive}$) сформована з середньої ціни капітального ремонту або заміни вузла ($C_{damage} = 50000$ ₴), тарифу на послуги вантажного

евакуатора ($C_{tow} = 5000\text{€}$) та типових логістичних штрафів за зрив строків доставки вантажу ($C_{penalty} = 10000\text{€}$). Загалом вартість одного інциденту становить 65 000 €.

Порівняльний економічний розрахунок зведено у таблицю 4.13.

Таблиця 4.13 – Порівняльний розрахунок річного бюджету на ТО та ремонт автопарку (100 ТЗ)

Стаття витрат / Параметр оцінки	Традиційна реактивна стратегія (A_1)	Прогностична стратегія на базі ML (A_2)
Базове планове ТО (для 100 ТЗ)	1 500 000 €	1 500 000 €
Кількість критичних інцидентів (10%)	10 відмов	10 (зафіксована рання деградація)
Середня вартість усунення 1 інциденту	65 000 € (капремонт, евакуація, штраф)	15 000 € (превентивна заміна деталі)
Загальні витрати на інциденти	650 000 € ($10 \times 65\,000$)	150 000 € ($10 \times 15\,000$)
Утримання ІТ-системи (сервери, ПЗ)	0 €	120 000 € ($100 \times 1\,200$)
Загальний річний бюджет (C_{total})	2 150 000 €	1 770 000 €
Загальна річна економія підприємства	—	380 000 €

При впровадженні прогностичної стратегії ($C_{predictive_total}$) базове планове ТО залишається незмінним. Проте замість аварійних ремонтів система завчасно виявляє деградацію (за допомогою моделей машинного навчання), і проводиться превентивна заміна зношеного компонента до його руйнування ($C_{early_part} = 15000\text{€}$ за інцидент). Додатково в бюджет закладається амортизація ІТ-системи, ліцензування ПЗ та підтримка серверів для обробки телеметрії всього автопарку ($C_{sys_annual} = 1200\text{€}$ на одне авто на рік).

На основі даних таблиці 5.13 розраховується загальний коефіцієнт оптимізації витрат (K_{cost}):

$$K_{cost} = \frac{C_{reactive_total} - C_{predictive_total}}{C_{reactive_total}} \times 100\%$$

$$K_{cost} = \frac{2150000 - 1770000}{2150000} \times 100\% = \frac{380000}{2150000} \times 100\% \approx 17.67\%$$

Отримане значення коефіцієнта оптимізації витрат ($K_{cost} \approx 17.7\%$) повністю корелює з емпіричними даними глобальних аналітичних досліджень. За даними звітів McKinsey & Company, масштабування інструментів предиктивної аналітики дозволяє компаніям скоротити загальні витрати на технічне обслуговування на 20 – 30% [216]. Водночас аналітика Deloitte фіксує середнє зниження витрат на утримання активів у діапазоні 10–40% за рахунок мінімізації критичних відмов [217].

Таким чином, розрахований у моделі показник 17.67% є консервативною та реалістичною оцінкою. Вона доводить, що незважаючи на додаткові капітальні та постійні операційні витрати на утримання інфраструктури машинного навчання для всього автопарку, система дозволяє уникнути найдорожчих наслідків аварій і забезпечує стабільну фінансову оптимізацію підприємства.

4.4.3. Багатокритеріальна оцінка методом аналізу ієрархій (MAI).

Оскільки вибір стратегії обслуговування залежить від багатьох суперечливих факторів, для розрахунку єдиного кількісного показника ефективності застосовано Метод аналізу ієрархій (MAI) Т. Сааті [218, 219].

Оцінюються дві альтернативи: A_1 (Традиційна реактивна система) та A_2 (Запропонована прогностична система на базі MLP).

Визначено три критерії оцінки: C_1 (Економічні витрати), C_2 (Час простою), C_3 (Безпека експлуатації та мінімізація FN-прогнозів).

Матриця парних порівнянь критеріїв (де Безпека має найвищий пріоритет) має вигляд:

$$C = \begin{pmatrix} 1 & 1/2 & 1/4 \\ 2 & 1 & 1/3 \\ 4 & 3 & 1 \end{pmatrix}$$

Обчислення головного вектора цієї матриці дає нормалізовані ваги критеріїв:

$$w_1(\text{Економіка}) = 0.14; w_2(\text{Час}) = 0.24; w_3(\text{Безпека}) = 0.14.$$

Далі здійснюється парне порівняння альтернатив A_1 та A_2 за кожним критерієм на основі розрахованих раніше даних:

– за критерієм C_1 (Економіка): A_2 перевершує A_1 у 5 разів. Ваги альтернатив: $A_1=0.17$, $A_2=0.83$;

– за критерієм C_2 (Час простою): A_2 перевершує A_1 у 4 рази. Ваги альтернатив: $A_1=0.20$, $A_2=0.80$;

– за критерієм C_3 (Безпека): Модель MLP зменшила пропуски дефектів з 12 до 1 (див. Таблицю 1), тому A_2 абсолютно перевершує A_1 (оцінка 9). Ваги: $A_1=0.1$, $A_2=0.90$.

Глобальний комплексний показник ефективності (E) розраховується як сума добутків ваг альтернатив на ваги критеріїв:

$$E(A_1) = 0.17 * 0.14 + 0.20 * 0.24 + 0.1 * 0.62 = 0.133$$

$$E(A_2) = 0.83 * 0.14 + 0.80 * 0.24 + 0.9 * 0.62 = 0.866$$

Висновок: індикатор ефективності прогностичної системи ($E=0.866$) у 6.5 разів перевищує показник традиційної реактивної системи ($E=0.133$). Це

дозволяє взяти за основу, що інтеграція моделей машинного навчання та Big Data є ефективним рішенням, яке забезпечує не лише рентабельність (ROI від 10:1 до 30:1), але й надійний рівень безпеки експлуатації транспортних засобів.

4.5. Висновки до четвертого розділу

1. Розроблено об'єктно-орієнтовану UML-модель програмного комплексу для прогностичного моніторингу. Спроектowana модель концептуально формалізує архітектуру інформаційної системи та взаємодію її ключових класів і сутностей. Чітко декомпоновано та розмежовано процеси генерації даних бортовими датчиками, збору телеметрії через стандартизований інтерфейс OBD-II, попередньої обробки інформації препроцесором, управління базами даних та безпосередньої генерації прогнозів модулем машинного навчання.

2. Спроектowana відмовостійку мікросервісну архітектуру для роботи в режимі реального часу. З метою забезпечення стабільної безперервної обробки великих даних архітектурне рішення базується на технологіях контейнеризації (зокрема Docker). Інфраструктура інтегрує модулі балансування навантаження, резервні вузли для забезпечення високої доступності, системи автоматичного планування та оркестрації завдань (Kubernetes), а також централізовані системи моніторингу і реєстрації подій, що гарантує стійкість програмного комплексу до апаратних та програмних збоїв.

3. Обґрунтовано та впроваджено гібридну модель управління базами даних із застосуванням нереляційних СУБД. Для ефективного збереження гетерогенних масивів інформації структуровані метадані (результати діагностики, профілі автопарків) розміщуються у реляційних БД. Натомість високошвидкісні потоки телеметрії моделюються у вигляді денормалізованих JSON/BSON-документів та зберігаються у документо-орієнтованій базі даних

MongoDB. Це забезпечує гнучкість схеми збереження, високу швидкість доступу за рахунок індексування та можливість ефективного горизонтального масштабування (шардингу).

4. Реалізовано паралельну обробку даних «на льоту» на базі парадигми MapReduce. Для усунення «вузьких місць» у процесі фільтрації, нормалізації та агрегування телеметрії інтегровано інструменти розподілених обчислень (Apache Spark, Kafka). Очищений потік даних маршрутизується до модуля аналізу, який, використовуючи спеціалізовані тензорні бібліотеки, застосовує автокодери для пошуку аномалій, багатошарові перцептрони (MLP) та рекурентні нейронні мережі для прогнозування відмов. Додатково оптимізовано утилізацію дискового простору шляхом стиснення архівних даних алгоритмом zstd.

5. Розроблено прикладний рівень системи для інтеграції в екосистему Smart City та периферійних обчислень (Edge Computing). Модуль звітності трансформує результати прогностичної аналітики в інтерактивні дашборди та структуровані звіти, забезпечуючи миттєве сповіщення користувачів щодо критичних змін технічного стану ТЗ. Архітектура програмного комплексу дозволяє здійснювати безшовну інтеграцію із зовнішніми платформами розумного міста через API, а також підтримує розгортання оптимізованих «легких» ML-моделей безпосередньо на периферійному OBD-II обладнанні для мінімізації мережових затримок.

ЗАГАЛЬНІ ВИСНОВКИ

1. Вперше розроблено розподілену інформаційну технологію дистанційного контролю транспортних засобів, яка, на відміну від відомих реляційних систем, інтегрує нереляційні NoSQL-бази даних (MongoDB), паралельні обчислення MapReduce та потокову обробку через брокери повідомлень (Apache Kafka). Застосування такого підходу вперше обґрунтовується необхідністю відмови від жорстко структурованих реляційних SQL-схем, що працюють за традиційною та повільною моделлю «запит-відповідь», на користь гнучкої гетерогенної архітектури (BASE), здатної обробляти швидкі нестационарні потоки телеметрії. Це забезпечує консистентність даних і горизонтальну масштабованість за умов нерегулярного надходження діагностичної інформації (Logged Vehicle Data, LVD).

2. Удосконалено метод подолання розрідженості навчальних вибірок баз даних LVD шляхом застосування статистичних оцінок щільності розподілу ймовірностей (за рахунок використання статистичного фільтра на основі непараметричного критерію Колмогорова-Смірнова) та алгоритмів мінімізації втрат даних при розширенні діагностичного простору ознак (із використанням алгоритму пучкового пошуку – beam search – для поетапного додавання релевантних ознак з урахуванням граничних умов розміру датасету), що підвищує якість формування навчальних датасетів для задач прогностичної аналітики.

3. Отримало подальший розвиток застосування архітектур глибокого навчання – багатошарового перцептрона (MLP) та рекурентних мереж довгої короткочасної пам'яті (LSTM) для виявлення прихованих дефектів у нестационарних потоках телеметрії OBD-II.

ЛІТЕРАТУРА

1. Грицук А.І., Грицук В.Ю., Ченцов А.В., Матейчик П.В., Краснокутська З.І. Використання інформаційних баз даних в задачах моніторингу автомобільного транспорту. *Сучасні енергетичні установки на транспорті і технології та обладнання для їх обслуговування. 9-а Міжнародна науково-практична конференція, 13-14 вересня 2018 р.* Херсон: ХДМА. 376 с. С.37-39.
2. Волков В.П., Грицук І.В., Грицук Ю.В., Волков Ю.В., Володарець М.В. Інформаційні системи моніторингу технічного стану автомобілів Харків: ФОП Панов А.М., 2018. 299 с.
3. Information model of V2I system of the vehicle technical condition remote monitoring and control in operation conditions / I. V. Gritsuk et al. *WCX world congress experience*. 400 Commonwealth Drive, Warrendale, PA, United States, 2018. URL: <https://doi.org/10.4271/2018-01-0024> (date of access: 28.12.2023)
4. Codd E. F. A relational model of data for large shared data banks. *Communications of the ACM*. 1983. Vol. 26, no. 1. P. 64–69. URL: <https://doi.org/10.1145/357980.358007> (date of access: 28.12.2023)
5. Пасічник В. В., Резніченко В. А. Організація баз даних та знань : підручник. Київ : Видавнича група BHV, 2006. 384 с.
6. DB-Engines Ranking. URL: World Wide Web: <https://db-engines.com/en/ranking> (date of access: 03.12.2021).
7. Lynch C. How do your data grow?. *Nature*. 2008. Vol. 455, no. 7209. P. 28–29. URL: <https://doi.org/10.1038/455028a> (date of access: 29.04.2023)
8. Дикань В. Л., Полякова О. М. Цифрова трансформація транспортної інфраструктури в умовах глобалізації. *Вісник економіки транспорту і промисловості*. 2021. Вип. 72-73. С. 7–15.
9. Apache Hadoop. URL: <https://hadoop.apache.org/> (дата звернення 29.04.2023)

10. Apache Spark. URL: <https://spark.apache.org/> (дата звернення 29.04.2023)
11. Big Data. URL: <https://www.it.ua/knowledge-base/technology-innovation/big-data-bolshie-dannye> (дата звернення 29.04.2023)
12. Drive Ukraine 2030. URL: <https://mtu.gov.ua/files/projects/str.html> (дата звернення 03.12.2021).
13. Грицук В.Ю., Грицук Ю.В. Обмін інформацією та міжоб'єктна взаємодія технології V2I. – Комп'ютерні інтелектуальні системи та мережі. Матеріали XI Всеукраїнської науково практичної WEB конференції аспірантів, студентів та молодих вчених (21-23 березня 2018 р.). – Кривий Ріг: ДВНЗ «Криворізький національний університет», 2018. – С.66 – 68.
14. Hritsuk V. Yu. The usage of databases and big data technology in the automobile transport. *Студентство. Наука. Іноземна мова: збірник наукових праць студентів, аспірантів та молодих науковців*. Харків : ХНАДУ, 2022. Вип. 14. 236 с. С.75-78.
15. Використання Big data у сучасному світі. URL: <https://cutt.ly/rYdSyET> (дата звернення 03.12.2021).
16. A review on autonomous vehicles: progress, methods and challenges / D. Parekh et al. *Electronics*. 2022. Vol. 11, no. 14. P. 2162. URL: <https://doi.org/10.3390/electronics11142162> (date of access: 27.11.2023).
17. Collaborative autonomous driving: vision and challenges / Z. Dong et al. *2020 international conference on connected and autonomous driving (metrocad)*, Detroit, MI, USA, 27–28 February 2020. 2020. URL: <https://doi.org/10.1109/metrocad48866.2020.00010> (date of access: 27.11.2023).
18. From assisted to automated driving. – URL: <https://www.infineon.com/cms/en/discoveries/adas-to-ad/> (дата звернення 27.11.2023)
19. Autonomous Vehicles: Potential Benefits, Challenges, And Risks – URL:

- <https://roboticsbiz.com/autonomous-vehicles-potential-benefits-challenges-and-risks/> (дата звернення 27.11.2023)
20. Advanced driver-assistance systems: Challenges and opportunities ahead – URL: <https://www.mckinsey.com/industries/semiconductors/our-insights/advanced-driver-assistance-systems-challenges-and-opportunities-ahead> (дата звернення 27.11.2023)
21. Грицук В.Ю. Розвиток систем автономного водіння за рахунок використання машинного навчання та BIG DATA. *Збірник наукових праць за матеріалами II Міжнародної науково-практичної конференції «Вища технічна освіта XXI століття: виклики, проблеми, перспективи», 14–15 грудня 2023 р.* Краматорськ – Івано-Франківськ: ДонНАБА. 2023. 338 с. С.325-328.
22. Грицук В.Ю. Технології Big Data в автомобільному сервісі: принципи та проблеми. *Комп'ютерні технології і мехатроніка. Збірник наукових праць за матеріалами V Міжнародної науково-методичної конференції.* Харків, ХНАДУ, 2023. 99 с. С.3-7.
23. Predictive maintenance in the automotive sector: a literature review / F. Arena et al. *Mathematical and computational applications*. 2021. Vol. 27, no. 1. P. 2. URL: <https://doi.org/10.3390/mca27010002> (date of access: 17.02.2024)
24. Predictive maintenance enabled by machine learning: use cases and challenges in the automotive industry / A. Theissler et al. *Reliability engineering & system safety*. 2021. Vol. 215. P. 107864. URL: <https://doi.org/10.1016/j.ress.2021.107864> (date of access: 17.02.2024)
25. Nagy J., Lakatos I. The past, the present and the future of online road vehicle diagnosis from car and scan tool communication perspective. *Advances in transdisciplinary engineering*. 2023. URL: <https://doi.org/10.3233/atde230419> (date of access: 17.02.2024)

26. Nagy J., Lakatos I. Possibilities of using of online vehicle diagnostics in the future. *Vehicle and automotive engineering* 4. Cham, 2022. P. 71–83. URL: https://doi.org/10.1007/978-3-031-15211-5_7 (date of access: 17.02.2024)
27. NHTSA. Vehicle and Tire Recalls, 1967-Present. URL: <https://www.kaggle.com/nhtsa/safety-recalls> (дата звернення 12.04.2024)
28. Al-Tae M. A., Khader O. B., Al-Saber N. A. Remote monitoring of vehicle diagnostics and location using a smart box with global positioning system and general packet radio service. *2007 IEEE/ACS international conference on computer systems and applications*, Amman, Jordan, 13–16 May 2007. 2007. URL: <https://doi.org/10.1109/aiccsa.2007.370910> (date of access: 17.02.2024)
29. What data is available from OBD? URL: <https://www.obdsol.com/knowledgebase/on-board-diagnostics/what-data-is-available-from-obd/> (дата звернення 12.04.2024)
30. Predicting Vehicle Maintenance using on-board and off-board data URL: <https://ecosystem.evolve-h2020.eu/predicting-vehicle-maintenance-using-on-board-and-off-board-data/> (дата звернення 12.04.2024)
31. Kumar R., Jain A. Driving behavior analysis and classification by vehicle OBD data using machine learning. *The journal of supercomputing*. 2023. URL: <https://doi.org/10.1007/s11227-023-05364-3> (date of access: 17.02.2024)
32. Driving behaviour analysis using machine and deep learning methods for continuous streams of vehicular data / N. Peppes et al. *Sensors*. 2021. Vol. 21, no. 14. P. 4704. URL: <https://doi.org/10.3390/s21144704> (date of access: 17.02.2024)
33. Navali N., Vanajakshi L., Bullock D. M. Application of on-board diagnostics (OBD) data for vehicle trajectory prediction. *Proceedings of the sixth international conference of the transportation research group of India*. Singapore, 2022. P. 319–328. URL: https://doi.org/10.1007/978-981-19-4204-4_19 (date of access: 17.02.2024)

34. Грицук В.Ю., Пронін С.В. Перспективи впровадження прогностичного обслуговування транспортних засобів на основі машинного навчання. *Матеріали Всеукраїнської науково-практичної конференції здобувачів вищої освіти та молодих учених «Сучасний стан досліджень в сфері IT» (19 квітня 2024 року)*. Харків: ХНАДУ, 2024. 215 с. С.77 – 81.
35. Optimization of air traffic management efficiency based on deep learning enriched by the long short-term memory (LSTM) and extreme learning machine (ELM) / M. Yousefzadeh Aghdam et al. *Journal of Big Data*. 2021. Vol. 8, no. 1. URL: <https://doi.org/10.1186/s40537-021-00438-6> (date of access: 23.11.2023)
36. Zhang J., Zhang J. Artificial intelligence applied on traffic planning and management for rail transport: a review and perspective. *Discrete dynamics in nature and society*. 2023. Vol. 2023. P. 1–17. URL: <https://doi.org/10.1155/2023/1832501> (date of access: 23.11.2023)
37. Alsrehin N. O., Klaib A. F., Magableh A. Intelligent transportation and control systems using data mining and machine learning techniques: a comprehensive study. *IEEE access*. 2019. Vol. 7. P. 49830–49857. URL: <https://doi.org/10.1109/access.2019.2909114>
38. Sayed Mansor T., Abri R. Data-Driven optimization of urban traffic using AI and real-time analysis. *International conference on pioneer and innovative studies*. 2023. Vol. 1. P. 507–514. URL: <https://doi.org/10.59287/icpis.881> (date of access: 23.11.2023)
39. Reis Da Silva, Bruno. Use of Big Data Analytics for Public Transport Efficiency: Evidence from Natal, (RN), Brazil. URL: <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1736700&dswid=2918> (date of access: 23.11.2023)
40. Traffic flow prediction with big data: a deep learning approach / Y. Lv et al. *IEEE transactions on intelligent transportation systems*. 2014. P. 1–9. URL: <https://doi.org/10.1109/tits.2014.2345663> (date of access: 23.11.2023)

41. Traffic and mobility data collection for real-time applications / J. Lopes et al. 2010
13th international IEEE conference on intelligent transportation systems - (ITSC 2010), Funchal, Madeira Island, Portugal, 19–22 September 2010. 2010.
 URL: <https://doi.org/10.1109/itsc.2010.5625282> (date of access: 23.11.2023)
42. Jiber M., Lamouik I., Ali Y., Sabri M.A. Traffic flow prediction using neural network. *2018 International Conference on Intelligent Systems and Computer Vision (ISCV)*, Fez, Morocco, 2018, pp. 1-4.
<https://doi.org/10.1109/ISACV.2018.8354066> (date of access: 23.11.2023)
43. Palla T. Intelligent traffic management using big data analytics and IOT. *International journal for research in applied science and engineering technology*. 2021. Vol. 9, no. 10. P. 1495–1503.
 URL: <https://doi.org/10.22214/ijraset.2021.38650> (date of access: 23.11.2023)
44. Muniswamaiah M., Agerwala T., Tappert C. Data virtualization for decision making in big data. *International journal of software engineering & applications*. 2019. Vol. 10, no. 5. P. 45–53. URL: <https://doi.org/10.5121/ijsea.2019.10503> (date of access: 01.06.2024)
45. Inmon W. H., Linstedt D. Data architecture - A primer for the data scientist: big data, data warehouse and data vault. Elsevier Science & Technology, 2019. 431 p.
46. Rybitskyi, O., Golian, V., Golian, N., Dudar, Z., Kalynychenko, O., Nikitin, D. Using OBD-2 technology for vehicle diagnostic and using it in the information system. *Bulletin of National Technical University "KhPI". Series: System Analysis, Control and Information Technologies*, (1 (9), 2023, 97–103.
<https://doi.org/10.20998/2079-0023.2023.01.15> (date of access: 01.06.2024)
47. Singh S. K., Singh A. K., Sharma A. Driving analysis for load and fuel consumption using OBD-II diagnostics. *Advances in information communication technology and computing*. Singapore, 2022. P. 121–131.
 URL: https://doi.org/10.1007/978-981-19-0619-0_11 (date of access: 01.06.2024)

48. Vehicle telematics for safer, cleaner and more sustainable urban transport: a review / O. Ghaffarpasand et al. *Sustainability*. 2022. Vol. 14, no. 24. P. 16386. URL: <https://doi.org/10.3390/su142416386> (date of access: 01.09.2024)
49. Аулін В.В., Голуб Д.В., Гриньків А.В., Лисенко С.В. Методологічні і теоретичні основи забезпечення та підвищення надійності функціонування автомобільних транспортних систем: монографія під заг. ред. д.т.н., проф. Ауліна В.В. Кропивницький: Видавець "КОД", 2017. 370 с.
50. Аулін В. В., Гриньків А. В., Головатий А. О., Лисенко С. В., Голуб Д. В., Кузик О.В., Тихий А. А. Методологічні основи проектування та функціонування інтелектуальних транспортних і виробничих систем: монографія під заг. ред. д.т.н., проф. Ауліна В.В. Кропивницький: Видавець Лисенко В.Ф., 2020. 428 с.
51. Аулін В.В., Гриньків А.В., Лисенко С.В. Інформаційні аспекти діагностування агрегатів транспортних засобів. *НАУ, 2015. ІПТК-2015.* – С.58-60.
52. Маулевич В. Визначення основних діагностичних параметрів робочого процесу транспортних дизелів в експлуатації: Дисертація на здобуття наукового ступеня кандидата технічних наук (доктора філософії) за спеціальністю 05.22.20 – «Експлуатація та ремонт засобів транспорту» («05 – технічні науки»). Одеса, 2020. 173 с.
URL: https://onmu.org.ua/spec_rada/Maulevych/Maulevych_dis1.pdf.
53. Лубко Д. В., Зінов'єва О. Г., Шаров О. В. Проектування та розробка експертної системи діагностування несправностей транспортних засобів. *Системи обробки інформації*. 2019. № 1(156). С. 15–21.
URL: <https://doi.org/10.30748/soi.2019.156.02> (дата звернення: 01.06.2024)
54. Forecasting fault events for predictive maintenance using data-driven techniques and ARMA modeling / M. Baptista et al. *Computers & industrial engineering*.

2018. Vol. 115. P. 41–53. URL: <https://doi.org/10.1016/j.cie.2017.10.033> (date of access: 01.06.2024)
55. Кукурудзяк Ю. Ю. Модель ідентифікації несправностей автомобільного двигуна. *Вісник Державного університету «Житомирська політехніка»*. 2014. № 2(69). С. 149-153 URL: <https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/7115/20.pdf?sequence=1> (date of access: 01.06.2024)
56. Аулін В.В., Гриньків А.В., Бруцький О.П. Прогнозування діагностичних параметрів технічного стану систем і агрегатів транспортних засобів. *Вісник інженерної академії України*. №4, 2016. С. 202-206. URL: <https://dspace.kntu.kr.ua/server/api/core/bitstreams/c2599e7f-67c8-4725-8af7-57e97d624bdd/content> (date of access: 01.06.2024)
57. Digital twin in industry: state-of-the-art / F. Tao et al. *IEEE transactions on industrial informatics*. 2019. Vol. 15, no. 4. P. 2405–2415. URL: <https://doi.org/10.1109/tii.2018.2873186> (date of access: 01.09.2024)
58. Волков В.П., Волкова Т.В., Волков Ю.В., Грицук І.В. Сучасний стан автомобільного транспорту і перспективи розвитку технічної експлуатації автомобілів. *Технічний сервіс агропромислового, лісового та транспортного комплексів*. №16, 2019. С.77-87. URL: <http://ts.khntusg.com.ua/index.php/ts/article/view/12/8> (date of access: 01.09.2024)
59. Аулін, В. В., Панарін, Д. Є. Комплексний підхід в оптимізації та плануванні процесів експлуатації і ремонту автомобілів з використанням телеметричних систем дистанційної електронної діагностики, *Вісник ЖДТУ. Серія "Технічні науки"*, №2(69), 2014, с. 29–32. doi: 10.26642/tn-2014-2(69)-29-32. URL: <http://vtn.ztu.edu.ua/article/view/31596> (date of access: 01.09.2024)
60. Мигаль В. Д. Системи моніторингу ефективної експлуатації автомобілів: навч. посібник / В. Д. Мигаль, І. О. Шевченко, М. Л. Шуляк. Х.: ДБТУ,

- «Майдан», 2023. 288 с. URL: https://repo.btu.kharkov.ua/bitstream/123456789/33892/1/NP_Systemy%20monitorynhu_2023.pdf (date of access: 01.09.2024)
61. EDA on Car Sales Dataset in Ukraine. Exploratory Data analysis on car sales data in Ukraine from car24.com. URL: <https://www.kaggle.com/datasets/swatikhedekar/eda-on-car-sales-dataset-in-ukraine/data> (date of access: 01.09.2024)
62. Статистичні дані по галузі автомобільного транспорту. URL: <https://mtu.gov.ua/en/content/statistichni-dani-po-galuzi-avtomobilnogo-transportu.html> (date of access: 01.09.2024)
63. Car Sale Advertisements. Data collected from private car sale advertisements in Ukraine. URL: <https://www.kaggle.com/datasets/antfarol/car-sale-advertisements> (date of access: 01.09.2024)
64. A systematic literature review of machine learning methods applied to predictive maintenance / T. P. Carvalho et al. *Computers & industrial engineering*. 2019. Vol. 137. P. 106024. URL: <https://doi.org/10.1016/j.cie.2019.106024> (date of access: 01.06.2024)
65. Павленко В.М., Кужель В.П., Хорін М.Є. Сутність автомобільної діагностики при впровадженні експертних систем. Вісник машинобудування, №2 (12), 2020. С.85-92. URL: <https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/31700/%d0%9f%d0%b0%d0%b2%d0%bb%d0%b5%d0%bd%d0%ba%d0%be.pdf?sequence=1&isAllowed=y> (date of access: 01.06.2024)
66. Шокарев О.М., Болтянська Н.І. Засоби діагностики сучасних автотранспортних засобів. Технічне забезпечення інноваційних технологій в агропромисловому комплексі. II Міжнародна науково-практична конференція, 02-27 листопада 2020 р. ТДАУ ім. Д. Моторного. С. 450-454. URL:

- <http://elar.tsatu.edu.ua/bitstream/123456789/12636/1/%d0%a8%d0%be%d0%ba%d0%b0%d1%80%d0%b5%d0%b2%20%d0%9e.%d0%9c.-2.pdf> (date of access: 01.06.2024)
67. Survey of smartphone-based sensing in vehicles for intelligent transportation system applications / J. Engelbrecht et al. *IET intelligent transport systems*. 2015. Vol. 9, no. 10. P. 924–935. URL: <https://doi.org/10.1049/iet-its.2014.0248> (date of access: 29.06.2024)
68. Türker G. F., Kutlu A. Survey of Smartphone applications based on OBD-II for Intelligent Transportation Systems. *International journal of engineering research and applications*. 2016. Vol. 6. Iss. 1. Pp. 69-73
69. Baek S.-h., Jang J.-W. Implementation of integrated OBD-II connector with external network. *Information systems*. 2015. Vol. 50. P. 69–75. URL: <https://doi.org/10.1016/j.is.2014.06.011> (date of access: 29.06.2024)
70. Design and implementation of a wireless OBD II fleet management system / R. Malekian et al. *IEEE sensors journal*. 2017. Vol. 17, no. 4. P. 1154–1164. URL: <https://doi.org/10.1109/jsen.2016.2631542> (date of access: 29.06.2024)
71. Introduction to SAE J1939. URL: <https://web.archive.org/web/20100925071858/http://www.kvaser.com/en/about-can/higher-layer-protocols/36.html> (date of access: 29.06.2024)
72. Campo M. T., Mukherjee S., Daily J. Real-Time network defense of SAE J1939 address claim attacks. *SAE international journal of commercial vehicles*. 2021. Vol. 14, no. 3. URL: <https://doi.org/10.4271/02-14-03-0026> (date of access: 28.06.2024)
73. Nugroho S. A., Ariyanto E., Rakhmatsyah A. Utilization of onboard diagnostic II (OBD-II) on four wheel vehicles for car data recorder prototype. *2018 6th international conference on information and communication technology (icoict)*, Bandung, 3–5 May 2018. 2018. URL: <https://doi.org/10.1109/icoict.2018.8528741> (date of access: 29.05.2024)

74. Uyaam M., Pirapuraj P. Intelligent Vehicle Diagnostic System for Service Center using OBD-II and IoT *Proceedings of Papers (ICST 2021) 1st International Conference on Science and Technology* 2021, P. 209-214 URL: http://ir.lib.seu.ac.lk/bitstream/123456789/5768/1/30%20P_ICST2021%20Final%20v11_.pdf (date of access: 29.05.2024)
75. Prytz R. Machine learning methods for vehicle predictive maintenance using off-board and on-board data : licentiate thesis. 2014. URL: <http://urn.kb.se/resolve?urn=urn:nbn:se:hh:diva-27869> (date of access: 29.06.2024)
76. Self-monitoring for maintenance of vehicle fleets / T. Rögnvaldsson et al. *Data mining and knowledge discovery*. 2017. Vol. 32, no. 2. P. 344–384. URL: <https://doi.org/10.1007/s10618-017-0538-6> (date of access: 29.05.2024)
77. Predicting the need for vehicle compressor repairs using maintenance records and logged vehicle data / R. Prytz et al. *Engineering applications of artificial intelligence*. 2015. Vol. 41. P. 139–150. URL: <https://doi.org/10.1016/j.engappai.2015.02.009> (date of access: 27.05.2024)
78. Kohavi R., John G. H. Wrappers for feature subset selection. *Artificial intelligence*. 1997. Vol. 97, no. 1-2. P. 273–324. URL: [https://doi.org/10.1016/s0004-3702\(97\)00043-x](https://doi.org/10.1016/s0004-3702(97)00043-x) (date of access: 19.07.2024)
79. Tan P.-N. Introduction to data mining by Pang-Ning Tan, Michael Steinbach, Vipin Kumar (2005) paperback. Pearson, 2005
80. Batista G., Bazzan, A., Monard, M.-C. Balancing Training Data for Automated Annotation of Keywords: a Case Study. *the Proc. Of Workshop on Bioinformatics*, 2003, p. 10-18. URL: <https://www.inf.ufrgs.br/maslab/pergamus/pubs/balancing-training-data-for.pdf> (date of access: 19.07.2024)
81. SMOTE: synthetic minority over-sampling technique / N. V. Chawla et al. *Journal of artificial intelligence research*. 2002. Vol. 16. P. 321–357. URL: <https://doi.org/10.1613/jair.953> (date of access: 19.07.2024)

82. A survey of end-to-end driving: architectures and training methods / A. Tampuu et al. *IEEE transactions on neural networks and learning systems*. 2020. P. 1–21. URL: <https://doi.org/10.1109/tnnls.2020.3043505> (date of access: 29.06.2024)
83. Perrotta F., Parry T., Neves L. C. Application of machine learning for fuel consumption modelling of trucks. *2017 IEEE international conference on big data (big data)*, Boston, MA, 11–14 December 2017. 2017. URL: <https://doi.org/10.1109/bigdata.2017.8258382> (date of access: 29.06.2024)
84. Vehicle remote health monitoring and prognostic maintenance system / U. Shafi et al. *Journal of advanced transportation*. 2018. Vol. 2018. P. 1–10. URL: <https://doi.org/10.1155/2018/8061514> (date of access: 29.09.2024)
85. Impact of driver behavior on fuel consumption: classification, evaluation and prediction using machine learning / P. Ping et al. *IEEE access*. 2019. Vol. 7. P. 78515–78532. URL: <https://doi.org/10.1109/access.2019.2920489> (date of access: 29.09.2024)
86. Basak S., Sengupta S., Dubey A. Mechanisms for integrated feature normalization and remaining useful life estimation using lstms applied to hard-disks. *2019 IEEE international conference on smart computing (SMARTCOMP)*, Washington, DC, USA, 12–15 June 2019. 2019. URL: <https://doi.org/10.1109/smartcomp.2019.00055> (date of access: 29.09.2024)
87. In-Vehicle communication cyber security: challenges and solutions / R. S. Rathore et al. *Sensors*. 2022. Vol. 22, no. 17. P. 6679. URL: <https://doi.org/10.3390/s22176679> (date of access: 29.09.2024)
88. Development of a sensor-based learning approach to prognostics in intelligent vehicle health monitoring / I. S. Cole et al. *2008 international conference on prognostics and health management (PHM)*, Denver, CO, USA, 6–9 October 2008. 2008. URL: <https://doi.org/10.1109/phm.2008.4711441> (date of access: 29.09.2024)

89. Predicting the need for vehicle compressor repairs using maintenance records and logged vehicle data / R. Prytz et al. *Engineering applications of artificial intelligence*. 2015. Vol. 41. P. 139–150.
URL: <https://doi.org/10.1016/j.engappai.2015.02.009> (date of access: 29.09.2024)
90. Remote vehicle state of health monitoring and its application to vehicle no-start prediction / Y. Zhang et al. *2009 IEEE autotestcon*, Anaheim, CA, USA, 14–17 September 2009. 2009. URL: <https://doi.org/10.1109/autest.2009.5314011> (date of access: 29.09.2024).
91. Kohnhauser F., Pullen D., Katzenbeisser S. Ensuring the safe and secure operation of electronic control units in road vehicles. *2019 IEEE security and privacy workshops (SPW)*, San Francisco, CA, USA, 19–23 May 2019. 2019.
URL: <https://doi.org/10.1109/spw.2019.00032> (date of access: 29.09.2024)
92. Natale M. D., Zeng H., Giusto P. Understanding and using the controller area network communication protocol: theory and practice. Springer, 2012. 244 p.
93. Ruff M. Evolution of local interconnect network (LIN) solutions. *2003 IEEE 58th vehicular technology conference. VTC 2003-fall (IEEE cat. no.03ch37484)*, Orlando, FL, USA, 9 October 2003. 2003.
URL: <https://doi.org/10.1109/vetecf.2003.1286317> (date of access: 25.09.2024).
94. Trends in automotive communication systems / N. Navet et al. *Proceedings of the IEEE*. 2005. Vol. 93, no. 6. P. 1204–1223.
URL: <https://doi.org/10.1109/jproc.2005.849725> (date of access: 29.08.2024)
95. Mathur, G., Mathur, R. An Analytical Study of Communication Protocols Used in Automotive Industry. URL: https://www.academia.edu/10781862/An_Analytical_Study_of_Communication_Protocols_Used_in_Automotive_Industry (date of access: 29.08.2024)
96. Shaw R., Jackman B. An introduction to FlexRay as an industrial network. *2008 IEEE international symposium on industrial electronics (ISIE 2008)*, Cambridge,

- UK, 30 June – 2 July 2008. 2008.
 URL: <https://doi.org/10.1109/isie.2008.4676987> (date of access: 29.08.2024).
97. Somers, B. Investigation of a Flexray - CAN Gateway in the Implementation of Vehicle Speed Control. Master's Thesis, 2009.
 URL: https://repository.wit.ie/1435/1/Investigation_of_a_Flexray_CAN_Gateway_in_the_Implementation_of_Vehicle_Speed_Control.pdf (date of access: 29.08.2024).
98. Sumorek, A.; Buczaj, M. New elements in vehicle communication “media oriented systems transport” protocol. *Teka Kom. Motoryz. I Energetyki Rol.* 2012, 12, p. 275–279.
99. A fuzzy diagnosis of multi-fault state based on information fusion from multiple sensors / H. Xue et al. *Journal of vibroengineering.* 2016. Vol. 18, no. 4. P. 2135–2148. URL: <https://doi.org/10.21595/jve.2016.16712> (date of access: 29.08.2024).
100. Dubois D. Possibility theory: an approach to computerized processing of uncertainty. Springer, 2012. 280 p.
101. Framework for building low-cost OBD-II data-logging systems for battery electric vehicles / C. Ramai et al. *Vehicles.* 2022. Vol. 4, no. 4. P. 1209–1222. URL: <https://doi.org/10.3390/vehicles4040064> (date of access: 29.08.2024).
102. Hosseini-Asl E., Zurada J. M., Nasraoui O. Deep learning of part-based representation of data using sparse autoencoders with nonnegativity constraints. *IEEE transactions on neural networks and learning systems.* 2016. Vol. 27, no. 12. P. 2486–2498. URL: <https://doi.org/10.1109/tnnls.2015.2479223> (date of access: 17.09.2024)
103. Automobile maintenance prediction using deep learning with GIS data / C. Chen et al. *Procedia CIRP.* 2019. Vol. 81. P. 447–452. URL: <https://doi.org/10.1016/j.procir.2019.03.077> (date of access: 17.09.2024).
104. Ghosh A. Possibilities and challenges for the inclusion of the electric vehicle (EV) to reduce the carbon footprint in the transport sector: a review. *Energies.*

2020. Vol. 13, no. 10. P. 2602. URL: <https://doi.org/10.3390/en13102602> (date of access: 19.09.2024).
105. Wang S., Ding Z., Fu Y. Feature selection guided auto-encoder. *Proceedings of the AAAI conference on artificial intelligence*. 2017. Vol. 31, no. 1. URL: <https://doi.org/10.1609/aaai.v31i1.10811> (date of access: 19.09.2024).
 106. Balin, M.F., Abid, A., Zou, J.. Concrete Autoencoders: Differentiable Feature Selection and Reconstruction. *Proceedings of the 36th International Conference on Machine Learning, Proceedings of Machine Learning Research*. PMLR 97:444-453, 2019. URL: <https://proceedings.mlr.press/v97/balin19a.html> (date of access: 19.09.2024).
 107. Salinas S. O., Nieto Lemus A. C. Data warehouse and big data integration. *International journal of computer science and information technology*. 2017. Vol. 9, no. 2. P. 01–17. URL: <https://doi.org/10.5121/ijcsit.2017.9201> (date of access: 19.09.2025).
 108. Polymorphic compressed replication of columnar data in scale-up hybrid memory systems / M. Zarubin et al. *SYSTOR '20: the 13th ACM international systems and storage conference*, Haifa Israel. New York, NY, USA, 2020. URL: <https://doi.org/10.1145/3383669.3398283> (date of access: 19.09.2025)
 109. Santoso L. W., Yulia. Data warehouse with big data technology for higher education. *Procedia computer science*. 2017. Vol. 124. P. 93–99. URL: <https://doi.org/10.1016/j.procs.2017.12.134> (date of access: 19.09.2025).
 110. Kleppmann M. Designing data-intensive applications: the big ideas behind reliable, scalable, and maintainable systems. O'Reilly Media, Incorporated, 2017. 616 p.
 111. Linking context to data warehouse design / A. Bouchra et al. *International journal of advanced computer science and applications*. 2019. Vol. 10, no. 1. URL: <https://doi.org/10.14569/ijacsa.2019.0100102> (date of access: 19.09.2025).

112. Kimball R., Ross M. The kimball group reader. Indianapolis, IN, USA : John Wiley & Sons, Inc, 2015. URL: <https://doi.org/10.1002/9781119228912> (date of access: 17.10.2025).
113. Kwon T.M. Development of Data Warehouse and Applications for Continuous Vehicle Class and Weigh-in-Motion Data. Retrieved from the University Digital Conservancy,2009. URL: <https://hdl.handle.net/11299/150973> (date of access: 17.10.2025).
114. Formalisms for representing ontologies: state of the art survey./ Baader F. et al. 2006. URL: https://www.researchgate.net/publication/236655901_Formalisms_for_representin_g_ontologies_state_of_the_art_survey (date of access: 17.10.2025).
115. Data integration in data warehousing / D. Calvanese et al. *International journal of cooperative information systems*. 2001. Vol. 10, no. 03. P. 237–271. URL: <https://doi.org/10.1142/s0218843001000345> (date of access: 17.10.2025).
116. Antunes A. L., Cardoso E., Barateiro J. Incorporation of ontologies in data warehouse/business intelligence systems - A systematic literature review. *International journal of information management data insights*. 2022. Vol. 2, no. 2. P. 100131. URL: <https://doi.org/10.1016/j.jjime.2022.100131> (date of access: 15.03.2024).
117. Kakish K., Kraft T.A. ETL Evolution for Real-Time Data Warehousing. 2012 Proceedings of the Conference on Information Systems Applied Research New Orleans Louisiana, USA. P. 1-12. URL: https://www.researchgate.net/profile/Theresa-Kraft/publication/280837435_ETL_Evolution_for_Real-Time_Data_Warehousing/links/560094c608ae07629e52af49/ETL-Evolution-for-Real-Time-Data-Warehousing.pdf (date of access: 15.03.2024).

118. Data warehousing process modeling from classical approaches to new trends: main features and comparisons / A. Dhaouadi et al. *Data*. 2022. Vol. 7, no. 8. P. 113. URL: <https://doi.org/10.3390/data7080113> (date of access: 15.03.2024).
119. El-Sappagh S. H. A., Hendawi A. M. A., El Bastawissy A. H. A proposed model for data warehouse ETL processes. *Journal of king saud university - computer and information sciences*. 2011. Vol. 23, no. 2. P. 91–104. URL: <https://doi.org/10.1016/j.jksuci.2011.05.005> (date of access: 15.03.2024).
120. Zagan E., Danubianu M. Data lake architecture for storing and transforming web server access log files. *IEEE access*. 2023. P. 1. URL: <https://doi.org/10.1109/access.2023.3270368> (date of access: 15.03.2024).
121. Cluster-size optimization within a cloud-based ETL framework for Big Data / E. Zdravevski et al. *2019 IEEE international conference on big data (big data)*, Los Angeles, CA, USA, 9–12 December 2019. 2019. URL: <https://doi.org/10.1109/bigdata47090.2019.9006547> (date of access: 16.03.2024).
122. Kohler J. Comparison of a faas- and saas-based ETL process in a cloud environment. *2023 IEEE 6th international conference on cloud computing and artificial intelligence: technologies and applications (cloudtech)*, Marrakech, Morocco, 21–23 November 2023. 2023. URL: <https://doi.org/10.1109/cloudtech58737.2023.10366090> (date of access: 16.03.2024).
123. Jo J., Lee K.-W. MapReduce-Based D_ELT framework to address the challenges of geospatial big data. *ISPRS international journal of geo-information*. 2019. Vol. 8, no. 11. P. 475. URL: <https://doi.org/10.3390/ijgi8110475> (date of access: 16.03.2024).
124. Jo J., Lee K.-W. High-Performance geospatial big data processing system based on mapreduce. *ISPRS international journal of geo-information*. 2018.

- Vol. 7, no. 10. P. 399. URL: <https://doi.org/10.3390/ijgi7100399> (date of access: 16.03.2024).
125. What is MapReduce? URL: <https://ua.talend.com/resources/what-is-mapreduce/> (date of access: 16.03.2024).
 126. Dayalan M. MapReduce: simplified data processing on large cluster. *International journal of research and engineering*. 2018. Vol. 5, no. 5. P. 399–403. URL: <https://doi.org/10.21276/ijre.2018.5.5.4> (date of access: 16.03.2024).
 127. Bryant RE: Data-intensive supercomputing: The case for DISC. 2007, Pittsburgh, PA – USA: School of Computer Science, Carnegie Mellon University, 1-20.
 128. The Apache Software Foundation. URL: <http://apache.org/> (date of access: 16.03.2024).
 129. LandQv2: a mapreduce-based system for processing arable land quality big data / X. Yao et al. *ISPRS international journal of geo-information*. 2018. Vol. 7, no. 7. P. 271. URL: <https://doi.org/10.3390/ijgi7070271> (date of access: 16.03.2024).
 130. Čerešňák R., Kvet M. Comparison of query performance in relational a non-relation databases. *Transportation research procedia*. 2019. Vol. 40. P. 170–177. URL: <https://doi.org/10.1016/j.trpro.2019.07.027> (date of access: 16.03.2024).
 131. Analysis And Comparision of Document-Based Databases with Relational Databases: MongoDB vs MySQL / R. Deari et al. *2018 international conference on information technologies (infotech)*, Varna, 20–21 September 2018. 2018. URL: <https://doi.org/10.1109/infotech.2018.8510719> (date of access: 10.09.2024).
 132. Performance Benefits of DataMPI: A Case Study with BigDataBench / F. Liang et al. *Big data benchmarks, performance optimization, and emerging hardware*. Cham, 2014. P. 111–123. URL: https://doi.org/10.1007/978-3-319-13021-7_9 (date of access: 10.09.2024).

133. Bala M., Boussaid O., Alimazighi Z. P-ETL: Parallel-ETL based on the MapReduce paradigm. *2014 IEEE/ACS 11th international conference on computer systems and applications (AICCSA)*, Doha, Qatar, 10–13 November 2014. 2014. URL: <https://doi.org/10.1109/aiccsa.2014.7073177> (date of access: 10.10.2024).
134. Apache Kafka. URL: <https://kafka.apache.org/> (date of access: 10.10.2024).
135. Rossi T. S. Big data analytics towards predictive maintenance at the INFN-CNAF computing centre : master's thesis. 2019. URL: <http://amslaurea.unibo.it/18430/> (date of access: 10.10.2024).
136. Leavitt N. Will nosql databases live up to their promise?. *Computer*. 2010. Vol. 43, no. 2. P. 12–14. URL: <https://doi.org/10.1109/mc.2010.58> (date of access: 10.10.2024).
137. Automatic nosql to relational database transformation with dynamic schema mapping / Z. Aftab et al. *Scientific programming*. 2020. Vol. 2020. P. 1–13. URL: <https://doi.org/10.1155/2020/8813350> (date of access: 10.10.2024).
138. Boicea A., Radulescu F., Agapin L. I. MongoDB vs oracle -- database comparison. *2012 third international conference on emerging intelligent data and web technologies (EIDWT)*, Bucharest, Romania, 19–21 September 2012. 2012. URL: <https://doi.org/10.1109/eidwt.2012.32> (date of access: 10.10.2024).
139. Arora, R., Aggarwal, R.R. An Algorithm for Transformation of Data from MySQL to NoSQL (MongoDB). *IJASCSE*, Volume 2, Special Issue 1, 2013. P. 6–12. URL: https://volume-2-special-issue-1.ijascse.org/An_algorithm.pdf (date of access: 10.10.2024).
140. Variety, a Schema Analyzer for MongoDB. URL: <https://github.com/variety/variety> (date of access: 10.10.2024).
141. Widenius M., Axmark D., Ab M. MySQL reference manual. O'Reilly Media, Inc., 2002. 712 p.
142. Deb N. R. P. Aspects of semantic ETL : doctoral thesis. 2020. 243 p. URL: <http://hdl.handle.net/10803/669830> (date of access: 10.11.2024)

143. Roatis A. Efficient querying and analytics of semantic web data : thesis. 2014.
URL: <http://www.theses.fr/2014PA112218/document> (date of access: 11.11.2024).
144. Christophides V. Resource description framework (RDF) schema (RDFS). *Encyclopedia of database systems*. New York, NY, 2018. P. 3225–3228.
URL: https://doi.org/10.1007/978-1-4614-8265-9_1319 (date of access: 11.11.2024)
145. Aydin G., Hallac I. R., Karakus B. Architecture and implementation of a scalable sensor data storage and analysis system using cloud computing and big data technologies. *Journal of sensors*. 2015. Vol. 2015. P. 1–11.
URL: <https://doi.org/10.1155/2015/834217> (date of access: 11.11.2024).
146. SETL: a programmable semantic extract-transform-load framework for semantic data warehouses / R. P. Deb Nath et al. *Information systems*. 2017. Vol. 68. P. 17–43. URL: <https://doi.org/10.1016/j.is.2017.01.005> (date of access: 11.11.2024).
147. SETLBI: an integrated platform for semantic business intelligence / R. P. Deb Nath et al. *WWW '20: the web conference 2020*, Taipei Taiwan. New York, NY, USA, 2020. URL: <https://doi.org/10.1145/3366424.3383533> (date of access: 11.11.2024).
148. Mehmood E., Anees T. Performance analysis of not only SQL semi-stream join using mongodb for real-time data warehousing. *IEEE access*. 2019. Vol. 7. P. 134215–134225. URL: <https://doi.org/10.1109/access.2019.2941925> (date of access: 11.11.2024).
149. Forensic investigation framework for the document store NoSQL DBMS: MongoDB as a case study / J. Yoon et al. *Digital investigation*. 2016. Vol. 17. P. 53–65. URL: <https://doi.org/10.1016/j.diin.2016.03.003> (date of access: 11.11.2024).
150. MongoDB: the definitive guide. O'Reilly Media, Incorporated, 2013. 432 p.

151. Abadi D. J. Tradeoffs between parallel database systems, hadoop, and hadoopdb as platforms for petabyte-scale analysis. *Lecture notes in computer science*. Berlin, Heidelberg, 2010. P. 1–3. URL: https://doi.org/10.1007/978-3-642-13818-8_1 (date of access: 11.11.2024).
152. Schema management for document stores / L. Wang et al. *Proceedings of the VLDB Endowment*. 2015. Vol. 8, no. 9. P. 922–933. URL: <https://doi.org/10.14778/2777598.2777601> (date of access: 11.11.2024).
153. Parametric schema inference for massive JSON datasets / M.-A. Baazizi et al. *The VLDB Journal*. 2019. Vol. 28, no. 4. P. 497–521. URL: <https://doi.org/10.1007/s00778-018-0532-7> (date of access: 11.11.2024).
154. Data Mining Fruitful and Fun. URL: <https://orangedatamining.com/> (доступ 31.08.2024)
155. Vehicle Maintenance Data. URL: <https://www.kaggle.com/datasets/chavindudulaj/vehicle-maintenance-data> (доступ 31.08.2024)
156. Automotive Vehicles Engine Health Dataset. URL: <https://www.kaggle.com/datasets/parvmodi/automotive-vehicles-engine-health-dataset/data> (доступ 31.08.2024)
157. Kizito R., Scruggs, Ph., Li X., Kress, R., Devinney M., Berg Th. The Application of Random Forest to Predictive Maintenance. *Proceedings of the 2018 IJSE Annual Conference*. P. 354-359. URL: https://www.researchgate.net/profile/Rodney-Kizito/publication/339311213_The_Application_of_Random_Forest_to_Predictive_Maintenance/links/5e4ab9db299bf1cdb9339d68/The-Application-of-Random-Forest-to-Predictive-Maintenance.pdf (доступ 31.08.2024)
158. Machine learning for predictive maintenance: a multiple classifier approach / G. A. Susto et al. *IEEE transactions on industrial informatics*. 2015. Vol. 11,

- no. 3. P. 812–820. URL: <https://doi.org/10.1109/tii.2014.2349359> (date of access: 31.08.2024).
159. Liaw, A. and Wiener, M., 2002, "Classification and Regression by Random Forest," *R News*, 2, 18-22.
160. Hybrid approach combining simulated annealing and deep neural network models for diagnosing and predicting potential failures in smart manufacturing / Y.-H. Hung et al. *Sensors and materials*. 2024. Vol. 36, no. 1. P. 49. URL: <https://doi.org/10.18494/sam4529> (date of access: 31.08.2024).
161. Barron A. R. Approximation and estimation bounds for artificial neural networks. *Machine learning*. 1994. Vol. 14, no. 1. P. 115–133. URL: <https://doi.org/10.1007/bf00993164> (date of access: 31.08.2024).
162. Thimm G., Fiesler E. High-order and multilayer perceptron initialization. *IEEE transactions on neural networks*. 1997. Vol. 8, no. 2. P. 349–359. URL: <https://doi.org/10.1109/72.557673> (date of access: 31.08.2024).
163. Kolen J. F., Pollack J. B. Backpropagation without weight transport. *1994 IEEE international conference on neural networks (ICNN'94)*, Orlando, FL, USA. URL: <https://doi.org/10.1109/icnn.1994.374486> (date of access: 31.08.2024).
164. Introduction to the theory of neural computation / J. Hertz et al. *Physics today*. 1991. Vol. 44, no. 12. P. 70. URL: <https://doi.org/10.1063/1.2810360> (date of access: 31.08.2024).
165. Platt J.C. Learning by Combining Memorization and Gradient Descent In R P Lippman et al (eds), *Advances in Neural Information Processing Systems* vol. III pp 714-720. Morgan Kaufmann, San Mateo 1991
166. Pinkus A. Approximation theory of the MLP model in neural networks. *Acta numerica*. 1999. Vol. 8. P. 143–195. URL: <https://doi.org/10.1017/s0962492900002919> (date of access: 31.08.2024).

167. Trenn S. Multilayer perceptrons: approximation order and necessary number of hidden units. *IEEE transactions on neural networks*. 2008. Vol. 19, no. 5. P. 836–844. URL: <https://doi.org/10.1109/tnn.2007.912306> (date of access: 31.08.2024).
168. Martens J., Sutskever I. Training deep and recurrent networks with hessian-free optimization. *Lecture notes in computer science*. Berlin, Heidelberg, 2012. P. 479–535. URL: https://doi.org/10.1007/978-3-642-35289-8_27 (date of access: 31.08.2024).
169. Training recurrent networks by evolino / J. Schmidhuber et al. *Neural computation*. 2007. Vol. 19, no. 3. P. 757–779. URL: <https://doi.org/10.1162/neco.2007.19.3.757> (date of access: 31.08.2024).
170. Gers F., Schraudolph N., Schmidhuber J. Learning Precise Timing with LSTM Recurrent Networks. *Journal of Machine Learning Research*. 2002. 3. 115–143. URL: 10.1162/153244303768966139 (date of access: 31.08.2024).
171. Gers F. A., Schmidhuber J., Cummins F. Learning to forget: continual prediction with LSTM. *Neural computation*. 2000. Vol. 12, no. 10. P. 2451–2471. URL: <https://doi.org/10.1162/089976600300015015> (date of access: 31.08.2024).
172. Deep convolutional neural network architecture design as a bi-level optimization problem / H. Louati et al. *Neurocomputing*. 2021. Vol. 439. P. 44–62. URL: <https://doi.org/10.1016/j.neucom.2021.01.094> (date of access: 31.08.2024).
173. AdaCB: an adaptive gradient method with convergence range bound of learning rate / X. Liao et al. *Applied sciences*. 2022. Vol. 12, no. 18. P. 9389. URL: <https://doi.org/10.3390/app12189389> (date of access: 31.08.2024).
174. A survey of optimization methods from a machine learning perspective / S. Sun et al. *IEEE transactions on cybernetics*. 2020. Vol. 50, no. 8. P. 3668–3681. URL: <https://doi.org/10.1109/tcyb.2019.2950779> (date of access: 31.08.2024).
175. Wilson A.C., Roelofs R., Stern M., Srebro N., Recht B. The marginal value of adaptive gradient methods in machine learning, in *Advances in Neural Information*

- Processing Systems. 2017. URL: <https://arxiv.org/abs/1705.08292> (date of access: 31.08.2024).
176. C. Szegedy et al., Going deeper with convolutions, *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 2015, pp. 1-9, URL: doi: 10.1109/CVPR.2015.7298594 (date of access: 31.08.2024).
 177. Hartigan J. A., Wong M. A. Algorithm AS 136: a k-means clustering algorithm. *Applied statistics*. 1979. Vol. 28, no. 1. P. 100. URL: <https://doi.org/10.2307/2346830> (date of access: 31.08.2024).
 178. Zhang Z. Improved adam optimizer for deep neural networks. *2018 IEEE/ACM 26th international symposium on quality of service (iwqos)*, Banff, AB, Canada, 4–6 June 2018. 2018. URL: <https://doi.org/10.1109/iwqos.2018.8624183> (date of access: 31.08.2024).
 179. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*. 2017. Vol. 60, no. 6. P. 84–90. URL: <https://doi.org/10.1145/3065386> (date of access: 31.08.2024).
 180. Ruder S, Park SM, Sim KB (2017) An overview of gradient descent optimization algorithms, URL: [arXiv:1609.04747v2](https://arxiv.org/abs/1609.04747v2) (date of access: 31.08.2024).
 181. Reyad M., Sarhan A. M., Arafa M. A modified Adam algorithm for deep neural network optimization. *Neural computing and applications*. 2023. URL: <https://doi.org/10.1007/s00521-023-08568-z> (date of access: 31.08.2024).
 182. Sashank J. Reddi, Satyen Kale & Sanjiv Kumar. On the Convergence of Adam and beyond. Published as a conference paper at ICLR 2018, 2018 URL: <https://arxiv.org/abs/1904.09237> (date of access: 31.08.2024).
 183. Wang Y., Zhou P., Zhong W. An optimization strategy based on hybrid algorithm of adam and SGD. *MATEC web of conferences*. 2018. Vol. 232. P. 03007. URL: <https://doi.org/10.1051/mateconf/201823203007> (date of access: 31.08.2024).

184. Hoo Z. H., Candlish J., Teare D. What is an ROC curve?. *Emergency medicine journal*. 2017. Vol. 34, no. 6. P. 357–359. URL: <https://doi.org/10.1136/emered-2017-206735> (date of access: 31.08.2024).
185. Assessing the accuracy of prediction algorithms for classification: an overview / P. Baldi et al. *Bioinformatics*. 2000. Vol. 16, no. 5. P. 412–424. URL: <https://doi.org/10.1093/bioinformatics/16.5.412> (date of access: 31.08.2024).
186. He H., Sun X. F-Score driven max margin neural network for named entity recognition in chinese social media. *Proceedings of the 15th conference of the european chapter of the association for computational linguistics: volume 2, short papers*, Valencia, Spain. Stroudsburg, PA, USA, 2017. URL: <https://doi.org/10.18653/v1/e17-2113> (date of access: 31.08.2024).
187. Chicco D., Jurman G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics*. 2020. Vol. 21, no. 1. URL: <https://doi.org/10.1186/s12864-019-6413-7> (date of access: 31.08.2024).
188. Ioualalen A., Martel M. Neural network precision tuning. *Quantitative evaluation of systems*. Cham, 2019. P. 129–143. URL: https://doi.org/10.1007/978-3-030-30281-8_8 (date of access: 31.08.2024).
189. Moreno M., Gutiérrez P. A., Hervás-Martínez C. A structural distance-based crossover for neural network classifiers. *International journal of pattern recognition and artificial intelligence*. 2012. Vol. 26, no. 06. P. 1250012. URL: <https://doi.org/10.1142/s0218001412500127> (date of access: 31.08.2024).
190. Introduction to TensorFlow. URL: <https://www.tensorflow.org/learn> (дата доступа 28.01.2025).
191. PyTorch. URL: <https://pytorch.org/> (дата доступа 28.01.2025).

192. Neural network models // scikit-learn developers. URL: https://scikit-learn.org/stable/modules/neural_networks_supervised.html (дата доступа 28.01.2025).
193. ONNX Runtime. URL: <https://onnxruntime.ai/> (дата доступа 28.01.2025).
194. Docker. URL: <https://www.docker.com/> (дата доступа 28.01.2025).
195. NVIDIA Triton Inference Server. URL: <https://developer.nvidia.com/triton-inference-server> (дата доступа 28.01.2025).
196. TensorFlow Lite. URL: <https://www.tensorflow.org/lite> (дата доступа 28.01.2025).
197. Turk M., Pentland A. Eigenfaces for recognition. *Journal of cognitive neuroscience*. 1991. Vol. 3, no. 1. P. 71–86. URL: <https://doi.org/10.1162/jocn.1991.3.1.71> (date of access: 28.01.2025).
198. Shahadi H. I., Haider H. T., Muhsen D. H. Vehicle monitoring based on GSM technology for safety and security. *IOP conference series: materials science and engineering*. 2018. Vol. 433. P. 012089. URL: <https://doi.org/10.1088/1757-899x/433/1/012089> (date of access: 28.01.2025).
199. brettwooldridge/HikariCP. URL: <https://github.com/brettwooldridge/HikariCP> (дата доступа 28.01.2025).
200. ITU-T Recommendation G.704. URL: https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-G.704-199810-1!!PDF-E&type=items (дата доступа 28.01.2025)
201. Lee H. K., Malkin T., Nahum E. Cryptographic strength of ssl/tls servers. *The 7th ACM SIGCOMM conference*, San Diego, California, USA, 24–26 October 2007. New York, New York, USA, 2007. URL: <https://doi.org/10.1145/1298306.1298318> (date of access: 28.01.2025).
202. Elasticsearch: The Official Distributed Search & Analytics Engine. URL: <https://www.elastic.co/elasticsearch> (date of access: 28.01.2025).
203. Apache Airflow. URL: <https://airflow.apache.org/> (date of access: 28.01.2025).

204. CronJob. URL: <https://kubernetes.io/docs/concepts/workloads/controllers/cron-jobs/> (date of access: 28.01.2025).
205. Amazon RDS Multi-AZ. URL: <https://aws.amazon.com/rds/features/multi-az/> (date of access: 28.01.2025).
206. Comparison among cloud technologies and cloud performance / O. Alzakholi et al. *Journal of applied science and technology trends*. 2020. Vol. 1, no. 2. P. 40–47. URL: <https://doi.org/10.38094/jastt1219> (date of access: 28.01.2025).
207. Sequential series-based prediction model in adaptive cloud resource allocation for data processing and security / I. Petrovska et al. 2023 *13th international conference on dependable systems, services and technologies (DESSERT)*, Athens, Greece, 13–15 October 2023. 2023. URL: <https://doi.org/10.1109/dessert61349.2023.10416496> (date of access: 28.01.2025).
208. Kafka mirroring (MirrorMaker). URL: <https://cwiki.apache.org/confluence/pages/viewpage.action?pageId=27846330> (date of access: 28.01.2025).
209. Puppet Infrastructure & IT Automation at Scale. URL: <https://www.puppet.com/> (date of access: 28.01.2025).
210. Automate infrastructure on any cloud with Terraform. URL: <https://www.terraform.io/> (date of access: 28.01.2025).
211. Волк М.О., Поповкін М.М. Методи моделювання масштабованих хмарних ресурсів. *Системи управління, навігації та зв'язку*. 2024. № 3. С. 97-100.
212. PostgreSQL: The World's Most Advanced Open Source Relational Database. URL: <https://www.postgresql.org/> (date of access: 28.01.2025).
213. MySQL. URL: <https://www.mysql.com/> (date of access: 28.01.2025).
214. Apache Hive. URL: <https://hive.apache.org/> (date of access: 28.01.2025).
215. Amazon Redshift. URL: <https://aws.amazon.com/redshift/> (date of access: 28.01.2025).

216. Prediction at scale: How industry can get more value out of maintenance / McKinsey & Company. URL: <https://www.mckinsey.com/capabilities/operations/our-insights/prediction-at-scale-how-industry-can-get-more-value-out-of-maintenance> (дата звернення: 28.01.2026).
217. Predictive Maintenance Solutions / Deloitte. URL: <https://www.deloitte.com/us/en/services/consulting/services/predictive-maintenance-and-the-smart-factory.html> (дата звернення: 28.01.2026).
218. Saaty T. L. How to make a decision: the analytic hierarchy process. *European journal of operational research*. 1990. Vol. 48, no. 1. P. 9–26. URL: [https://doi.org/10.1016/0377-2217\(90\)90057-i](https://doi.org/10.1016/0377-2217(90)90057-i) (date of access: 25.01.2026).
219. Saaty T. L. Transport planning with multiple criteria: the analytic hierarchy process applications and progress review. *Journal of advanced transportation*. 1995. Vol. 29, no. 1. P. 81–126. URL: <https://doi.org/10.1002/atr.5670290109> (date of access: 25.01.2026).

ДОДАТОК А. АКТИ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОЇ РОБОТИ



«ЗАТВЕРДЖУЮ»
Перший проректор
Михайло ШУЛЯК
» _____ 2026 р.

АКТ

Про впровадження в навчальний процес результатів дисертаційної роботи
здобувача наукового ступеня доктор філософії за спеціальністю 122
Комп'ютерні науки

**«Методи та моделі інформаційних технологій для моніторингу
технічного стану транспортних засобів»**

Грицука Валерія Юрійовича

Результати дисертаційної роботи Грицука Валерія Юрійовича впроваджено в навчальний процес кафедри комп'ютерних наук і інформаційних систем Харківського національного автомобільно-дорожнього університету.

При проведенні навчального процесу зі студентами спеціальності F3 «Комп'ютерні науки» у межах дисципліни «Інтелектуальний аналіз даних», розглядаються теоретичні та прикладні аспекти предиктивної аналітики. Зокрема вивчаються запропоновані автором схема збору необхідної інформації, моделі даних, методи та моделі аналізу та прогнозу технічного стану транспортних засобів з використанням відповідних програмних продуктів для підвищення якості автомобільного сервісу.

Використання матеріалів і результатів дисертаційної роботи Грицука Валерія Юрійовича в навчальному процесі кафедри автоматизації та комп'ютерно-інтегрованих технологій сприяє підвищенню якості підготовки здобувачів освіти, розвитку компетентностей та вдосконаленню навчально-методичного забезпечення дисциплін.

Завідувачка кафедри Комп'ютерних наук
і інформаційних систем

д.т.н., доцент

Ганна Плехова

Науковий керівник

к.т.н., доцент

Сергій ПРОНІН



ТОВАРИСТВО З ОБМЕЖЕНОЮ ВІДПОВІДАЛЬНІСТЮ
«ІНТРАНС-КІП»
 код ЄДРПОУ 40965143

Юрид. адреса: просп. Науки, буд. 25, оф. 415, м. Харків, Україна, 61166
 тел./факс (057) 720-96-56
 e-mail: intrans.kip@gmail.com

АКТ

Про впровадження результатів дисертаційної роботи здобувача наукового ступеня
 доктор філософії за спеціальністю 122 Комп'ютерні науки
**«Методи та моделі інформаційних технологій для моніторингу технічного
 стану транспортних засобів»**
 Грицука Валерія Юрійовича

Комісія виконала оцінку роботи спроможності та якості технології представлених в дисертаційному дослідженні аспіранта ХНАДУ Грицука В.Ю. Основними рисами розробки є використання інформаційних систем збору обробки та збереження інформації для прогностичного моніторингу технічного стану транспортних засобів. Отримані практичні результати можна використати для підвищення якості обслуговування парку рухомого складу. В умовах експлуатації рухомого складу застосування запропонованої технології дозволить оптимізувати витрати на ремонт та технічне обслуговування автомобільної техніки.

Основним практичним результатом дослідження є представлення суб'єктам транспортного обслуговування методики яка дозволяє перейти від реактивної моделі ремонту до прогностичної аналітики, що оптимізує логістику запасних частин, мінімізує час простою комерційного транспорту та знижує ризик відмов критичних вузлів. Отриманий в дослідженні підхід до прогнозування може бути інтегрований у діяльність організації для оперативного прийняття рішень по обслуговуванню рухомого складу.

Виконавчий директор ТОВ «ІНТРАНС-КІП»



Ірина ШИРІНА