

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
ННПО, факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри інтелектуальних
інформаційних систем

_____Юрій КОНДРАТЕНКО

« ____ » _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА
ІНФОРМАЦІЙНА СИСТЕМА АНАЛІЗУ ПОКАЗНИКІВ
СЕНСОРІВ ТЕХНОЛОГІЧНИХ ЛІНІЙ

Спеціальність 122 Комп'ютерні науки
Освітня програма «Інтелектуальні інформаційні системи»

Здобувач

_____Олександра ФІНЬКОВА

« ____ » _____ 2024 р.

Керівник канд. пед. наук, доцент

_____Надія БОЛЮБАШ

« ____ » _____ 2024 р.

Чорноморський національний університет імені Петра Могили
(повне найменування закладу вищої освіти)

Факультет	ННПО, факультет комп'ютерних наук
Кафедра	Інтелектуальних інформаційних систем
Рівень вищої освіти	Другий (магістерський)
Освітній ступень	Магістр
Спеціальність	122 Комп'ютерні науки
Освітня програма	Інтелектуальні інформаційні системи

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних
інформаційних систем

_____ Юрій КОНДРАТЕНКО

« ____ » _____ 2024 р.

ЗАВДАННЯ
на кваліфікаційну роботу здобувача

Фінькової Олександри Вадимівни

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: «Інформаційна система аналізу показників сенсорів технологічних ліній».

Керівник роботи: Болюбаш Надія Миколаївна, доцент кафедри інтелектуальних інформаційних систем, канд. пед. наук, доцент.

Затверджена наказом ЧНУ ім. Петра Могили від «03» червня 2024 р. № 140/1.

2. Строк представлення кваліфікаційної роботи « ____ » _____ 2024 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні: інформаційна система аналізу показників сенсорів технологічних ліній для їх прогностного технічного обслуговування з використанням методів описової та специфічної контекстної статистики і кореляційного аналізу; набори даних з показниками сенсорів технологічних ліній підприємств важкої промисловості.

4. Перелік питань, що підлягають розробці: дослідження теоретичних засад прогнозного технічного обслуговування технологічних ліній, аналіз існуючих досліджень та рішень технічного обслуговування; обґрунтування вибору моделей і методів прогнозного технічного обслуговування та інструментальних засобів розробки інформаційної системи; моделювання, проектування, здійснення програмної реалізації системи аналізу показників сенсорів технологічних ліній та оцінка її якості.

5. Перелік графічних матеріалів: презентація, рисунки, таблиці.

Керівник роботи

(Особистий підпис)

Надія БОЛЮБАШ

(Власне ім'я ПРІЗВИЩЕ)

Здобувач

(Особистий підпис)

Олександра ФІНЬКОВА

(Власне ім'я ПРІЗВИЩЕ)

Дата видачі завдання «07» червня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

кваліфікаційної роботи

Тема: Інформаційна система аналізу показників сенсорів технологічних ліній

№	Найменування роботи	Початок	Закінчення	Примітки
1	Отримання завдання на виконання КР	03.06.2024	07.06.2024	Виконано
2	Аналіз предметної області та постановка задачі	10.06.2024	20.06.2024	Виконано
3	Огляд літературних джерел за темою кваліфікаційної роботи, аналіз публікацій та існуючих підходів до прогностного обслуговування технологічних ліній	21.06.2024	01.07.2024	Виконано
4	Огляд існуючих методів та моделей прогностного технічного обслуговування, аналіз та вибір стеку технологій розробки	01.09.2024	25.10.2024	Виконано
5	Реалізація обраних технологій з аналізом отриманих результатів	26.10.2024	21.11.2024	Виконано
6	Перший попередній захист КР на засіданні комісії кафедри	22.11.2024	22.11.2024	Виконано
7	Корегування роботи за результатами попереднього захисту	23.11.2024	05.12.2024	Виконано
8	Другий попередній захист КР на засіданні комісії кафедри	06.12.2024	06.12.2024	Виконано
9	Доробка та остаточне оформлення КР	07.12.2024	10.02.2024	Виконано
10	Подання КР, її електронної копії та інших документів (відгуку, рецензії) до захисту	16.12.2024	17.12.2024	Виконано

Керівник роботи

(Особистий підпис)

Надія БОЛЮБАШ

(Власне ім'я ПРІЗВИЩЕ)

Здобувач

(Особистий підпис)

Олександра ФІНЬКОВА

(Власне ім'я ПРІЗВИЩЕ)

Дата складання календарного плану
«19» червня 2024 р.

АНОТАЦІЯ

на кваліфікаційну роботу

Фінькової Олександрі Вадимівни

на тему: «ІНФОРМАЦІЙНА СИСТЕМА АНАЛІЗУ ПОКАЗНИКІВ

СЕНСОРІВ ТЕХНОЛОГІЧНИХ ЛІНІЙ»

Кваліфікаційна робота присвячена розробці та здійсненню програмної реалізації інформаційної системи аналізу показників сенсорів технологічних ліній для виявлення проблемних даних, які впливають на якість прогнозування їх поломок. Що є актуальним в умовах високих темпів інформатизації, оскільки це дозволяє автоматизувати аналіз даних сенсорів, генерувати списки потенційних джерел проблем при тренуванні моделей для прогнозування збоїв та підвищує точність прогнозу поломок технологічних ліній.

Об'єкт дослідження – процес технічного обслуговування технологічних ліній.

Предмет дослідження – програмні засоби, методи та моделі дескриптивного і предикативного аналізу показників сенсорів технологічних ліній.

Мета дослідження – підвищення точності прогнозування поломок технологічних ліній важкої промисловості шляхом створення інформаційної системи аналізу показників сенсорів із використанням методів описової, контекстної статистики та кореляційного аналізу даних.

Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків та додатків. У першому розділі розкрито теоретичні засади прогнозного обслуговування технологічних ліній. У другому розділі проаналізовано та здійснено вибір моделей і методів прогнозного технічного обслуговування технологічних ліній. У третьому розділі обґрунтовано вибір технологій і засобів розробки інформаційної системи. У четвертому розділі описано моделювання, проектування, програмну реалізацію інформаційної системи аналізу показників сенсорів та оцінку її якості.

Кваліфікаційна робота містить 108 сторінок, 51 рисунок, 2 таблиці, 43 джерела, 1 додаток.

Ключові слова: *технологічна лінія, прогнозне технічне обслуговування, прогнозне технічне обслуговування на основі стану, описова статистика, контекстна статистика, кореляційний аналіз даних.*

ABSTRACT

of the qualification work

Finkova Oleksandra Vadymivna

on the subject: «**INFORMATION SYSTEM FOR DATA ANALYSIS OF
SENSOR INDICATORS IN TECHNOLOGICAL LINES**»

The master's qualification work is devoted to the development and implementation of the information system for the analysis of indicators of sensors of technological lines to identify problematic data that affect the quality of forecasting their breakdowns. Which is relevant in conditions of high rates of informatization, as it allows automating the analysis of sensor data, generating lists of potential sources of problems when training models for predicting failures, and increases the accuracy of forecasting breakdowns of technological lines.

Object of research – the process of technical maintenance of technological lines.

Subject of research – software tools, methods and models of descriptive and predicative analysis of indicators of sensors of technological lines.

The purpose of the study is to increase the accuracy of forecasting breakdowns of technological lines of heavy industry by creating an information system for the analysis of sensor indicators using the methods of descriptive, contextual statistics and data correlation analysis.

The master's qualification work consists of an introduction, four sections, conclusions and appendices. The first chapter reveals the theoretical principles of technical maintenance of technological lines. In the second section, models and methods of predictive maintenance of technological lines were analyzed and selected. The third section substantiates the choice of technologies and means of information system development. The fourth chapter describes the modeling, design, software implementation of the information system for the analysis of sensor indicators and its quality assessment.

The master's thesis contains 108 pages, 51 figures, 2 tables, 43 sources, 1 appendix.

Key words: *technological line, predictive maintenance, condition-based predictive maintenance, descriptive statistics, contextual statistics, data correlation analysis.*

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	4
ВСТУП.....	5
1 ТЕОРЕТИЧНІ ЗАСАДИ ПРОГНОЗНОГО ОБСЛУГОВУВАННЯ ТЕХНОЛОГІЧНИХ ЛІНІЙ.....	8
1.1 Предметна сфера обслуговування технологічних ліній.....	8
1.2 Існуючі рішення прогностного технічного обслуговування.....	14
1.3 Аналіз останніх досліджень та публікацій.....	22
1.4 Проблеми прогностного технічного обслуговування технологічних ліній	27
1.5 Постановка задачі.....	32
Висновки до розділу 1.....	34
2 МОДЕЛІ ТА МЕТОДИ ПРОГНОЗНОГО ТЕХНІЧНОГО ОБСЛУГОВУВАННЯ НА ОСНОВІ СТАНУ	36
2.1 Виявлення аномалій	36
2.2 Описова статистика	38
2.3 Специфічна контекстна статистика.....	42
2.4 Кореляційний аналіз показників сенсорів.....	44
Висновки до розділу 2.....	47
3 ТЕХНОЛОГІЇ І ЗАСОБИ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ	49
3.1 Мова програмування та бібліотеки Python	49
3.2 Фреймворки Flask і Plotly Dash.....	50
3.3 СУБД та сервіси для доступу й аналізу даних.....	52
3.4 Опис технологій back-end і front-end.....	55
Висновки до розділу 3.....	58
4 РОЗРОБКА ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ АНАЛІЗУ ПОКАЗНИКІВ СЕНСОРІВ.....	60
4.1 Алгоритмізація логіки вебзастосунку	60
4.2 Моделювання інформаційної системи	62

4.3 Клас RD_analysis: логіка роботи та основні етапи процесу	65
4.4 Скрипт для зв'язку між back-end і front-end.....	72
4.5 Структура інтерфейсу, налаштування системи для проведення аналізу ...	74
4.6 Аналіз та виявленням проблемних даних	85
4.7 Тестування та оцінка якості інформаційної системи	103
Висновки до розділу 4.....	112
ВИСНОВКИ	114
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	118
ДОДАТОК А Текстовий блок із результатами проведеного аналізу	121

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ШІ – штучний інтелект

AI – Artificial Intelligence

AWS – Amazon Web Services

CBPdM – Condition-based Predictive Maintenance

CMS – Condition Monitoring System

DBC – Dash Bootstrap Components

EAM – Enterprise Asset Management

HVAC – Heating, Ventilation, and Air Conditioning

I4.0 – Industry 4.0

IAM – Identity and Access Management

IIoT – Industrial Internet of Things

IoT – Internet of Things

PdM – Predictive Maintenance

PHM – Prognostics and Health Management

RCM – Reliability Centered Maintenance

RPM – Revolutions per Minute

RUL – Remaining Useful Life

SQL – Structured Query Language

WSGI – Web Server Gateway Interface

XSS – Cross-Site Scripting

ВСТУП

Актуальність. В умовах високих темпів розвитку інформаційного суспільства широко розповсюдженим у різних сферах виробництва є застосування для технічного обслуговування технологічних ліній штучного інтелекту (ШІ), Інтернету речей (ІоТ) та хмарних обчислень із метою прогнозного передбачення збоїв у роботі обладнання. Можливість отримання у режимі реального часу даних великої кількості показників, які фіксують сенсори технологічної лінії, дозволяє проводити їх розширену аналітику, яка є основою для формування економічно ефективних протоколів обслуговування.

Технологічні лінії у важкій промисловості є сукупністю машин та апаратів, сполучених транспортними пристроями, призначеними для виконання певного технологічного процесу, який є специфічним для кожної індустрії та підприємства. Їх прогностичне обслуговування базується на Big Data моделях, які аналізують великі обсяги даних, що поступають із технологічних ліній. Проведення аналізу великої кількості даних, які надають сенсори технологічних ліній, базується на застосуванні методів та алгоритмів Big Data: Machine Learning, Data mining, краудсорсингу, нейромереж, предикативного та статистичного аналізу, візуалізації, змішування та інтеграції даних. Модель прогнозування відмов зосереджена на якнайшвидшому виявленні сигнатур майбутньої несправності, зафіксованих даними сенсорів. Це сприяє зменшенню затрат на ремонт обладнання за рахунок зменшення інтервалу часу між виявленням проблеми і повною відмовою компонента.

Прогнозне технічне обслуговування технологічних ліній базується на предикативній аналітиці, яка постійно оцінює стан обладнання в режимі реального часу, допомагаючи максимізувати його продуктивність, час безвідмовної роботи та термін служби, мінімізуючи загальну вартість експлуатації та обслуговування ліній. Однак моделі, які здійснюють прогнозування збоїв та поломок у роботі обладнання, в умовах реального виробництва стикаються з необхідністю

постійного оновлення їх параметрів. Що обумовлено зміною технічних характеристик пристроїв та обладнання у процесі їх експлуатації. З метою оновлення параметрів моделей важливим є проведення якісної аналітики даних показників датчиків технологічних ліній, які дозволяють виявити критичні місця шляхом зменшення часу передбачення можливої поломки.

Метою дослідження є підвищення точності прогнозування поломок технологічних ліній важкої промисловості шляхом створення інформаційної системи аналізу показників сенсорів із використанням методів описової, контекстної статистики та кореляційного аналізу даних.

Досягнення поставленої мети обумовлює необхідність вирішення наступних **завдань**:

- дослідити теоретичні засади прогнозного обслуговування технологічних ліній, проаналізувати існуючі дослідження та рішення технічного обслуговування;
- обґрунтувати вибір моделей і методів прогнозного технічного обслуговування та інструментальних засобів розробки інформаційної системи;
- здійснити моделювання, проектування, програмну реалізацію системи аналізу показників сенсорів технологічних ліній та оцінити її якість.

Об'єктом дослідження є процес технічного обслуговування технологічних ліній.

Предметом дослідження є програмні засоби, методи та моделі дескриптивного і предикативного аналізу показників сенсорів технологічних ліній.

Методологічною основою дослідження є загальнонаукові аналітичні методи, методи дескриптивної та предикативної статистики, методи машинного навчання, прогнозного технічного обслуговування та технічного обслуговування на основі стану, методи оцінки точності прогнозу, які дозволили вивчити предмет та об'єкт дослідження, дослідити розвиток науково-методичних засад, напрямів та шляхів підвищення точності прогнозування поломок технологічних ліній.

Результати дослідження обговорювалися на XXVII Всеукраїнській науково-практичній конференції «Могилянські читання – 2024: Досвід та тенденції

розвитку суспільства в Україні: глобальний, національний та регіональний аспекти» (6-10 листопада 2024 року) та отримали схвалення.

Практичне значення отриманих результатів полягає в тому, що розроблена інформаційна система була впроваджена для аналізу великої кількості показників сенсорів технологічних ліній важкої промисловості з метою виявлення критичних даних, що супроводжувалося підвищенням точності моделей прогнозування поломок та зменшенням часу їх виявлення.

Структура кваліфікаційної роботи. Відповідно до мети, завдань і предмета дослідження кваліфікаційна робота складається із вступу, чотирьох розділів, висновку, списку використаних джерел та 1 додатку. Загальний обсяг кваліфікаційної роботи – 108 сторінок, кількість використаних джерел – 43.

1 ТЕОРЕТИЧНІ ЗАСАДИ ПРОГНОЗНОГО ОБСЛУГОВУВАННЯ ТЕХНОЛОГІЧНИХ ЛІНІЙ

1.1 Предметна сфера обслуговування технологічних ліній

Технологічна лінія є сукупністю машин та апаратів, сполучених транспортними пристроями, призначених для виконання певного технологічного процесу. В умовах високих темпів цифровізації суспільства широко розповсюдженим у різних сферах виробництва є впровадження у сферу управління та обслуговування технологічних ліній передових інформаційних технологій, які базуються на застосуванні методів та моделей штучного інтелекту (англ. Artificial Intelligence, AI), машинного навчання та Інтернету речей (англ. Internet of Things, IoT).

Система управління технологічними процесами – це сукупність пристроїв і програмних засобів, призначених для контролю та підтримки необхідних параметрів технологічних процесів. Основні її функції включають:

- контроль процесів та регулювання параметрів;
- логічне керування процесами;
- забезпечення безпеки виробництва;
- управління тривожними повідомленнями і аварійною сигналізацією в ручному або автоматичному режимах роботи;
- обробку та відображення інформації про перебіг процесів у режимі реального часу на моніторі (моніторах) диспетчера (оператора);
- забезпечення зв'язку з зовнішніми застосунками (мережами);
- генерування звітів про хід технологічних процесів за будь-який період часу;
- запис і архівування бази даних технологічних процесів із можливістю подальшого перегляду за будь-який період часу, у зручній для сприйняття і порівняння формі.

У цілому функції системи управління технологічними процесами можна розділити на інформаційні, керуючі та допоміжні.

До *інформаційних функцій* відносять: збирання, перетворення та зберігання інформації про перебіг технологічного процесу та передачу її оперативному персоналу. Для цього в автоматичному режимі проводиться вимірювання основних параметрів технологічного процесу, виявлення та сигналізація про передаварійні та аварійні ситуації. На виклик оператора визначається час, коли відбулося відхилення в роботі агрегатів, проводиться аналіз нових досягнутих техніко-економічних показників та інші розрахунки. При цьому система може працювати в режимі порадики і на основі аналізу вихідної інформації розробляти рекомендації з управління процесом. Вона може працювати в діалоговому режимі, оператор «радиться» за різними показниками режиму роботи та вибирає оптимальний режим.

Керуючі функції системи управління технологічними процесами – вироблення та здійснення керуючих впливів на технологічні об'єкти. До основних керуючих функцій відносяться: стабілізація технологічного процесу, програмна зміна режиму роботи, захист обладнання від аварій, управління послідовністю пуску та зупинки та агрегатів, розподіл потоків та навантажень, розробка та реалізація заданого режиму роботи.

Допоміжні функції забезпечують вирішення внутрішньо системних завдань. Вони призначені для контролю функціонування технічних засобів, правильності виконання алгоритму роботи, зберігання інформації про роботу системи тощо.

Перерахуємо можливі режими роботи систем управління технологічними процесами.

1. *Ручний режим роботи* – безпосереднє керування оператором положенням регулюючого органу (по стану виконуючого органу механізму) – на основі ввідних через засоби введення операторної станції даних оперативного управління.

2. *Автоматичний режим роботи* – оператор виконує функції контролю та спостереження за роботою технологічного обладнання, яке автоматично керується

(включаючи під дією регулятора), протіканням технологічного процесу. Підтриманням заданого режиму виконується комплексом програмно-технічних засобів системи автоматично під керуванням системи верхнього рівня керування.

3. *Дистанційний режим роботи* – оператор дистанційно формує команди та задачі, які необхідні для стабілізації чи зміни параметрів технологічного режиму, зокрема з ціллю запобігання та локалізації аварій та нештатних ситуацій. Також оператор виконує функції контролю і спостереження за роботою технологічного обладнання й протіканням технологічного процесу [1].

У тренді сучасних тенденцій змін роботи технологічних ліній є націленість на реалізацію саме автоматичного режиму роботи системи управління технологічними процесами, який включає програмно-логічні контролери, програмне забезпечення для реалізації моделей автоматизованого управління технологічними процесами та датчики і первинні перетворювачі контролю технологічних параметрів та стану основного обладнання. Обмін інформацією між компонентами системи є автоматичним.

Сигнали від датчиків по лінії зв'язку подаються на контролери, які продукують інформаційні сигнали. Кожен такий сигнал містить створені на основі заданих алгоритмів рекомендації по дії на виконуючий механізм. Керуючий сигнал приймається виконуючим механізмом та виконується. Виконуючий механізм повідомляє програмований логічний контролер про виконання сигналу. Система готова до нового циклу виконання команди [2]. У передбачених програмою керування випадках по заданим алгоритмам здійснюється опитування датчиків і виконуючих механізмів, інформуючи про їх стан. У цілях безпеки, в випадку виникнення аварійної ситуації та для реалізації прогностичного технологічного обслуговування сигнали фіксуються й архівуються.

Технічне обслуговування технологічної лінії є важливим аспектом її надійної та ефективної роботи [3]. Різні типи стратегій обслуговування включають традиційні види:

– *реактивне технічне обслуговування*: відбувається, коли виникає повна

поломка або збій, використовується, коли обладнання легко замінюється або ремонтується без шкоди для діяльності;

- *коригувальне обслуговування*: відбувається, коли помічається проблема;
- *профілактичне заздалегідь визначене технічне обслуговування*: включає регулярні та періодичні (на основі часу) графіки, використовується для обладнання, вартість простою якого невисока, а ремонт не займає багато часу [4].

Прогресивна стратегія обслуговування технологічних ліній базується на RCM-методології (англ. Reliability Centered Maintenance), спрямованій на забезпечення надійності обладнання. Головним принципом цієї методології є недопущення відхилення параметрів стану обладнання технологічних ліній до тих значень, які можуть призвести до її поломок. У цьому випадку метою обслуговування є не підтримки кожної одиниці обладнання, а забезпечення надійності критичних для діяльності технологічної лінії виборництва процесів.

Впровадження RCM-методології супроводжується застосуванням різних EAM-систем (англ. Enterprise Asset Management, EAM), які є прикладним програмним забезпеченням для управління основними фондами підприємств, спрямованими на управління фізичними активами, режимами роботи технологічних ліній, ризиками і втратами на протязі усього життєвого циклу їх діяльності. Перевагами застосування EAM-систем є скорочення витрат на технічне обслуговування обладнання, забезпечення обслуговування та ремонту обладнання без зниження рівня надійності.

Здійснюється також впровадження автоматизованих систем моніторингу стану (англ. Condition Monitoring System, CMS) – передових технологій, призначених для моніторингу стану промислового обладнання у режимі реального часу. Використовуючи датчики та діагностичні інструменти, CMS може оцінювати різні параметри, такі як температура, вібрація, тиск та вологість для виявлення аномалій. Основна мета цих систем — прогнозувати потенційні збої до того, як вони відбудуться, що дозволяє підприємствам вживати превентивних заходів.

В основі CMS лежить збір даних. Датчики встановлюються на критично важливі машини та обладнання технологічної лінії для вимірювання змінних, що вказують на стан працездатності та продуктивність обладнання, таких як вібрація, шум або коливання температури. Потім ці дані передаються до централізованої системи, де передові алгоритми аналізують їх для виявлення закономірностей або відхилень від нормальних умов експлуатації. Як тільки потенційну проблему виявлено, система попереджає команди технічного обслуговування, дозволяючи їм усунути проблему до того, як вона призведе до збою або дорогого простою.

RCM-методологія включає більш іноваційні підходи до технічного обслуговування технологічних ліній:

– *прогнозне технічне обслуговування* (англ. Predictive Maintenance, PdM) – комплексний підхід, який дозволяє визначити стан технологічної лінії, яка знаходиться в експлуатації та оцінити, коли слід провести технічне обслуговування на основі сукупних даних датчиків і тенденції для прогнозування майбутньої деградації та відмови обладнання;

– *прогнозне технічне обслуговування на основі стану* (англ. Condition-based Predictive Maintenance, CBPM) є стратегією, яка підтримує рішення щодо прогнозного технічного обслуговування залежно від процесу моніторингу стану, використовуючи дані, зібрані під час моніторингу, щоб виконати технічне обслуговування саме в той момент, коли це необхідно, і до того, як станеться критичний збій.

Поява технології промислового Інтернету речей (англ. Industrial Internet of Things, IIoT) суттєво оптимізувала управління промисловими операціями шляхом підключення промислових активів до інформаційних систем. IIoT є основою для реалізації ініціативи Industry 4.0. Прогнозне технічне обслуговування на основі стану (англ. Condition-based Predictive maintenance, CBPM), також відоме як Технічне обслуговування 4.0 (англ. Maintenance 4.0), є одним із головних напрямків Industry 4.0 та IIoT. Проте впровадження CBPM у промисловості сьогодні ще не має стандартної структури чи еталонної архітектури.

CBPdM стежить за процесами збору, вилучення та попередньої обробки даних, щоб підтримувати прийняття рішень за допомогою інформації про стан працездатності системи. Інформацію, отриману в результаті процесу виявлення аномалій і діагностики несправностей, можна використовувати на рівні PdM для кращого опису справності системи то більш точного прогнозу майбутньої поломки. На рисунку 1.1 продемонстровано основні компоненти CVPdM і PdM, тобто виявлення аномалій, діагностику несправностей і прогноз [5].

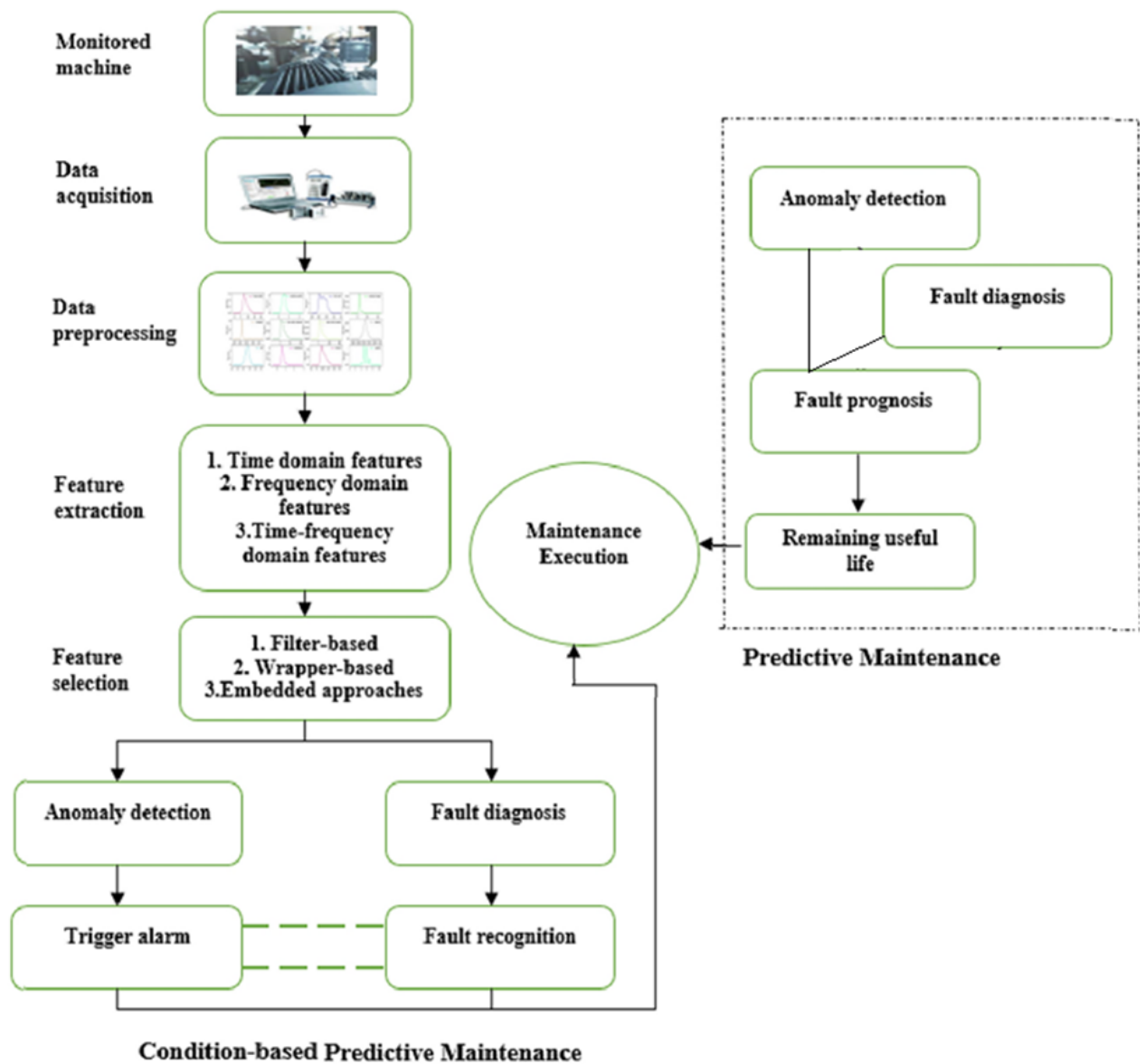


Рисунок 1.1 – Основні компоненти CVPdM і PdM

Прогнозне технічне обслуговування використовує часові ряди історичних даних і даних про несправності, щоб передбачити майбутню потенційну працездатність обладнання та заздалегідь передбачити проблеми. Це дозволяє підприємствам оптимізувати планування технічного обслуговування та підвищити надійність. Моделі, які здійснюють управління технологічними лініями та їх прогнозне технічне обслуговування, постійно вдосконалюються, їх розробкою та супроводом займаються фахівці відділів Data Science. Однак є проблема, обумовлена різними технічними причинами, що потребують корегування та оновлення параметрів моделей. З цією метою здійснюється прогнозне обслуговування технологічних ліній на основі стану, яке передбачає аналіз даних датчиків із метою виявлення проблемних даних та прогнозування можливих проблем у обслуговуванні технологічних ліній. З урахуванням цієї аналітики відбувається оновлення моделей та забезпечується уникнення збоїв у їх роботі.

1.2 Існуючі рішення прогнозного технічного обслуговування

Прогнозне технічне обслуговування забезпечує своєчасне технічне обслуговування, яке мінімізує час простою та підвищує ефективність роботи. Технологічна основа прогнозного обслуговування ґрунтується на інтеграції Інтернету речей (IoT), штучного інтелекту (AI) і хмарних обчислень:

- датчики IoT, встановлені в обладнанні, збирають величезну кількість даних у режимі реального часу, відстежуючи різні параметри, такі як температура, вібрація та тиск;
- потім ці дані обробляються та аналізуються за допомогою штучного інтелекту та алгоритмів машинного навчання для виявлення закономірностей і аномалій, які вказують на можливі збої;
- хмарні обчислення відіграють ключову роль у цьому процесі, полегшуючи зберігання, обробку та аналіз великих обсягів даних.

Разом ці технології дають компаніям можливість приймати обґрунтовані рішення щодо обслуговування активів. Це допомагає їм забезпечити оптимальну продуктивність і подовжити термін служби активів.

Робоче місце фахівця Data Science, який здійснює прогнозне технічне обслуговування технологічних ліній сьогодні може знаходитися далеко від виробництва важкої промисловості – фабрики, шахти, заводу чи іншого середовища, де важке обладнання покладається на його розробки. Адже Інтернет речей (IoT) – термін, що стосується приблизно мільярдів пристроїв, які можуть збирати дані за допомогою датчиків і передавати ці дані – змінює цю картину для деяких практиків. Важке обладнання може містити сотні чи тисячі датчиків, і з розвитком IoT дані, зібрані цими датчиками, можна накопичувати та аналізувати для створення економічної цінності [6].

Інтернет речей – це мережа взаємопов'язаних пристроїв, які з'єднуються й обмінюються даними з іншими пристроями IoT і хмарою. Пристрої IoT, як правило, оснащені такими технологіями, як датчики та програмне забезпечення, і можуть включати механічні та цифрові машини і споживчі об'єкти. Ці пристрої охоплюють усе: від предметів повсякденного побуту до складних промислових інструментів. Організації в різних галузях все частіше використовують IoT, щоб працювати ефективніше, надавати клієнтові покращене обслуговування, покращувати процес прийняття рішень і підвищувати цінність бізнесу.

Завдяки IoT дані можна передавати через мережу, не вимагаючи взаємодії «людина-людина» або «людина-комп'ютер». Річчю в Інтернеті речей може бути людина з імплантатом серцевого монітора, сільськогосподарська тварина з транспондером з біочіпом, автомобіль із вбудованими датчиками, які сповіщають водія про низький тиск у шинах, або будь-який інший природний чи штучно зроблений об'єкт, якому можна призначити адресу Інтернет-протоколу та який може передавати дані через мережу.

Підвищення можливостей підключення важкого обладнання та, загалом, підключення будь-якого пристрою з датчиком, обумовлено рядом факторів.

Зменшення витрат на пропускну здатність, доступність Wi-Fi і стільникових мереж, а також надійні хмарні інфраструктури полегшують збір, передачу, зберігання та аналіз даних датчиків (рис. 1.2).

Враховуючи ці нові можливості, традиційні промислові компанії, технологічні компанії та безліч стартапів змагаються за місце на промисловому ринку Інтернету речей. Для цього багато хто покладається на науковців, які аналізують, візуалізують і створюють прогнози на основі цих нових потоків даних.

Системи Інтернету речей функціонують шляхом збору даних із датчиків, вбудованих у пристрої Інтернету речей, які потім передаються через шлюз Інтернету речей для аналізу програмою або внутрішньою системою [7].

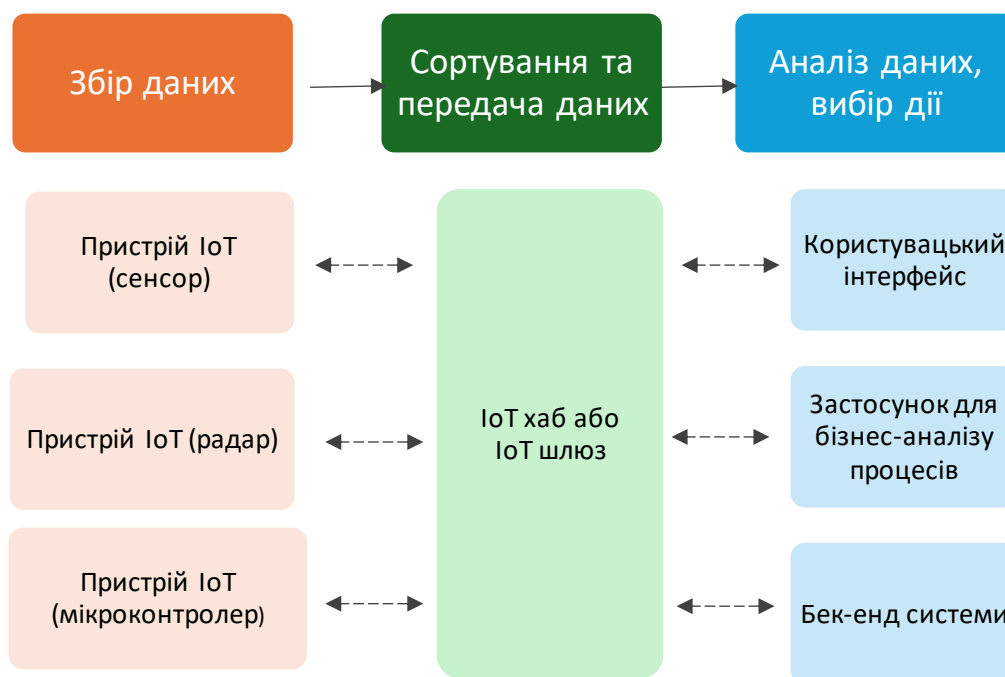


Рисунок 1.2 – Приклад ІоТ системи

Іноваційні технології технічного обслуговування інтенсивно впроваджуються у різні сфери діяльності сучасного суспільства. Здійснимо їх огляд.

1. *Прогнозне технічне обслуговування у виробництві.* Простій у виробництві може призвести до значних фінансових втрат із перервами у виробництві, затримками поставок і збільшенням витрат на обслуговування. Прогнозне технічне

обслуговування допомагає мінімізувати ці втрати, дозволяючи виробникам контролювати стан критичних активів у режимі реального часу та вживати профілактичних заходів до того, як виникнуть поломки [8]. Наприклад:

- моніторинг вібрації у верстатах з ЧПК (числовим програмним керуванням): використання датчиків для виявлення незвичайних вібрацій або звуків, які вказують на знос, що забезпечує своєчасне обслуговування;

- відстеження температури в двигунах: впровадження тепловізорів або датчиків для моніторингу температури та запобігання перегріву;

- якість оливи в гідравлічних системах: аналіз якості оливи для прогнозування та запобігання механічних несправностей, забезпечення оптимальної роботи гідравлічних систем;

- моніторинг тиску повітря у пневматичних системах: використання датчиків для відстеження рівнів тиску повітря та прогнозування потенційних витоків або несправностей;

- аналіз зносу конвеєрних стрічок: використання машинного бачення або датчиків для моніторингу зносу, запобігання неочікуваним поломкам.

2. Прогнозне технічне обслуговування в нафтогазовій промисловості. Нафтогазова промисловість стикається з унікальними проблемами, коли мова заходить про моніторинг і обслуговування обладнання, особливо у віддалених і морських місцях. Традиційні підходи до технічного обслуговування, які покладаються на планові перевірки, не можуть забезпечити достатню видимість стану критичних активів у реальному часі. Прогнозне технічне обслуговування постає як рішення для цих проблем, уможливаючи точне технічне обслуговування на основі виявлення аномалій без шкоди для стандартів продуктивності чи безпеки. Наприклад:

- моніторинг компресора: використання датчиків для відстеження продуктивності, вібрації та температури компресорів, щоб запобігти збоєм і оптимізувати ефективність;

- цілісність трубопроводу: використання ультразвукових датчиків або інтелектуальних пристроїв для визначення корозії, тріщин або витоків у трубопроводах, уможливаючи завчасний ремонт;
- справність бурового обладнання: впровадження датчиків для моніторингу стану свердл, насосів та іншого важливого бурового обладнання, прогнозування зносу;
- ефективність клапанів і насосів: аналіз даних від датчиків IoT для оцінки працездатності та продуктивності клапанів і насосів, сприяючи своєчасному техобслуговуванню;
- виявлення витoku газу: використання вдосконалених датчиків і алгоритмів машинного навчання для виявлення й попередження про потенційні витoki газу до того, як вони стануть загрозою для навколишнього середовища чи безпеки.

3. Прогнозне технічне обслуговування в технічному обслуговуванні літаків. Технічне обслуговування літаків регулюється суворими правилами та стандартами для забезпечення безпеки пасажирів. Традиційні методи технічного обслуговування, які покладаються на планові перевірки, іноді можуть призводити до непотрібних перевірок або невиявлених проблем, які ставлять під загрозу безпеку та продуктивність. Прогнозне технічне обслуговування пропонує більш тонкий підхід, використовуючи дані в реальному часі, щоб отримати інформацію про фактичний стан компонентів літака, сприяючи своєчасному втручання на основі потреб, а не за заздалегідь визначеним графіком [9]. Приклади:

- моніторинг двигуна: використання датчиків для постійного моніторингу продуктивності двигуна, виявлення аномалій у вібрації, температурі або тиску, які можуть вказувати на загрозові збої;
- справність шасі: встановлення датчиків у вузлі шасі для оцінки стану різних компонентів і виявлення зносу та інших проблем якомога раніше;
- системи авіоніки: реалізація моніторингу систем авіоніки в реальному часі для виявлення потенційних електронних або системних збоїв до їх виникнення;

– моніторинг гідравлічних систем: аналіз даних від датчиків для підтримки оптимальної продуктивності гідравлічних систем, прогнозування та запобігання таким проблемам, як витіки або втрата тиску;

– структурна цілісність: використання вдосконалених датчиків і технологій візуалізації для виявлення структурних проблем, наприклад корозії або тріщин, що дозволяє проводити профілактичне обслуговування.

4. *Прогнозне технічне в галузі охорони здоров'я.* У сфері охорони здоров'я надійність обладнання залежить не лише від оперативної ефективності, а й нерозривно пов'язана з безпекою пацієнтів і якістю медичної допомоги. Прогнозне технічне обслуговування дає медичним організаціям можливість ефективно контролювати своє обладнання та інфраструктуру, плануючи технічне обслуговування на основі даних про фактичний знос і продуктивність, щоб забезпечити оптимальну доступність і надійність обладнання [10]. Приклади:

– сканери МРТ і КТ: датчики відстежують температуру магніту, рівні охолоджувальної рідини та стан калібрування на наявність таких аномалій, як перегрів, низький рівень охолоджувальної рідини або відхилення калібрування;

– вентилятори: виявлення зменшення потоку повітря, аномалій тиску або засмічення фільтрів дозволяє негайно вжити коригувальні дії, забезпечуючи безперервну оптимальну продуктивність, комфорт і безпеку пацієнта;

– обладнання для стерилізації: використання датчиків для виявлення таких проблем, як відхилення температури або невідповідність тиску, гарантує, що технічне обслуговування відбувається до того, як якість стерилізації буде скомпрометована;

– інфузійні насоси: моніторинг механічних і електронних компонентів для виявлення аномалій, таких як нерегулярна швидкість потоку або електричні несправності, забезпечує точну доставку ліків і безпеку пацієнтів;

– діагностичне обладнання: використання датчиків для відстеження роботи такого обладнання, як ЕКГ та ультразвукові апарати, виявлення таких проблем, як перешкоди сигналу або погіршення чіткості зображення.

5. Прогнозне технічне обслуговування в енергетиці. Енергетичний сектор відіграє вирішальну роль у живленні нашого сучасного світу. Ефективність і надійність обладнання в цьому секторі безпосередньо впливають на доставку енергії до мільйонів будинків і підприємств [11]. Через зростаючі потреби в енергії простої або збій обладнання призводять до значних перебоїв у обслуговуванні та фінансових втрат. Приклади:

– моніторинг трансформаторів: справність трансформаторів відстежується шляхом моніторингу якості масла, температури та електричних струмів. Аномалії, такі як деградація масла або коливання струму, визначаються для своєчасного технічного обслуговування, запобігання збоїв і відключень;

– продуктивність генератора: використовуючи датчики в режимі реального часу для моніторингу вібрації, температури та електричної потужності, підприємства можуть точно виявляти такі проблеми, як перегрів або електричний дисбаланс, щоб оптимізувати роботу генератора та довговічність;

– системи електромережі: впровадження передових технологій моніторингу для виявлення коливань або дисбалансу в електромережі дозволяє негайно втручатися, щоб підтримувати стабільність мережі та запобігати знеструмленням;

– справність ядерного реактора: вдосконалені датчики безперервно відстежують рівень радіації, температуру та тиск, виявляючи порушення та забезпечуючи безпечну та ефективну роботу реактора;

– системи зберігання акумуляторів: датчики відстежують температуру, цикли заряду та напругу, виявляючи аномалії, які викликають технічне обслуговування, забезпечуючи оптимізовану роботу акумулятора та надійне зберігання енергії.

6. Приклади прогнозованого обслуговування в управлінні нерухомістю та об'єктами. У секторі нерухомості та управління об'єктами оптимальна продуктивність і довговічність будівельних систем і обладнання є найважливішими для забезпечення безпечного та комфортного простору для мешканців. Неочікувані поломки обладнання можуть призвести до дорогого ремонту, зниження вартості майна та незадоволення мешканців. Приклади:

– системи HVAC (Heating, Ventilation, and Air Conditioning): датчики відстежують температуру, повітряний потік і рівні вологості, виявляючи порушення, такі як несподівані температурні коливання або зменшення повітряного потоку, і полегшуючи попереджувальне обслуговування для забезпечення комфорту та якості повітря [12];

– ефективність ліфтів і ескалаторів: використання датчиків для відстеження механічних характеристик і зносу, визначення таких проблем, як напруга двигуна або зміщення;

– системи безпеки: впровадження моніторингу в реальному часі для оцінки працездатності сигналізації, камер і датчиків, виявлення збоїв або вразливостей;

– сантехнічні системи: використання датчиків і аналітики для моніторингу тиску та швидкості потоку, раннього виявлення витоків або засмічень, щоб запобігти пошкодженню водою та підтримувати ефективність системи;

– електричні системи: використання датчиків для моніторингу навантаження, напруги та струму, виявлення незвичайних коливань або перевантажень для забезпечення безпеки та надійності системи.

Українські підприємства все більше усвідомлюють важливість впровадження прогностного технічного обслуговування та технічного обслуговування на основі стану для підвищення ефективності виробництва.

Зокрема, компанія SHEN пропонує системний підхід до технічного обслуговування виробничих об'єктів, заснований на передових технологіях та досвіді професіоналів, що сприяє оптимізації процесів обслуговування та підвищенню надійності обладнання.

Компанія Danfoss пропонує концепцію прогностного технічного обслуговування в контексті моніторингу стану обладнання як економічно ефективну, спрямовану на підвищення продуктивності та зниження витрат на технічне обслуговування завдяки впровадженню інтелектуальних приводів та периферійних обчислень.

1.3 Аналіз останніх досліджень та публікацій

Прогнозне технічне обслуговування та технічне обслуговування на основі стану є актуальною темою зарубіжних і вітчизняних досліджень, що обумовлено зростанням кількості промислових об'єктів, а також кількості даних, яка отримується від різноманітного обладнання технологічних ліній у всьому світі.

У ході проведеного аналізу було встановлено, що в умовах переходу до четвертої промислової революції – Industry 4.0 (I4.0), яка базується на конвергенції кіберфізичних систем і промислового інтернету речей, удосконалення функцій прогнозного технічного обслуговування реалізується шляхом створення CMS-систем моніторингу стану. Сучасні дослідження науковців у сфері прогнозного технічного обслуговування зосереджені навколо трьох основних напрямів: на основі фізичних моделей, на основі даних та гібридні моделі [13].

Прогнозне технічне обслуговування базується на використанні великих обсягів даних для виявлення потенційних несправностей обладнання та оптимізації графіків технічного обслуговування. У даній кваліфікаційній роботі здійснюється розробка проекту, який передбачає аналіз великих обсягів даних для моделей прогнозного технічного обслуговування технологічних ліній, оскільки ефективність PdM залежить від якості цих даних, а проблеми, пов'язані з їхньою точністю, повнотою та консистентністю, є значним викликом. У ході вивчення цього питання було досліджено ряд наукових статей, які аналізують проблеми якості даних у PdM, а також рішення, що пропонуються авторами.

А. Ucar, М. Karakose та N. Kirimca розглядають прогнозне технічне обслуговування (PdM) як політику, яка використовує дані й аналітику, щоб передбачити, коли один із компонентів реальної системи буде знищено та з'являться деякі аномалії, щоб технічне обслуговування можна було виконати до того, як станеться поломка. Автори зазначають, що використання передових технологій, таких як аналітика даних і штучний інтелект, покращує продуктивність

і точність систем прогнозного технічного обслуговування, а також підвищує їхню автономність і адаптивність у складних і динамічних робочих середовищах [14].

У даній публікації розглядаються останні розробки PdM на основі ШІ, зосереджуючись на ключових компонентах, надійності та майбутніх тенденціях. Спочатку аналізуються найсучасніші методи (state-of-the-art, SOTA), виклики та можливості, пов'язані з PdM на основі ШІ. Пізніше обговорюються інтеграція технологій штучного інтелекту в PdM у застосунки, використовувани на підприємствах, взаємодія людини та робота, етичні проблеми, що виникають у зв'язку з використанням штучного інтелекту, а також можливості тестування та перевірки розроблених політик. Це дослідження демонструє потенційні робочі сфери для майбутніх досліджень, такі як цифровий двійник, метавсесвіт, генеративний ШІ, роботи для співпраці (коботи), технологія блокчейн, надійний ШІ та промисловий Інтернет речей (англ. Industrial Internet of Things, IIoT), використовуючи комплексне дослідження поточної SOTA техніки, можливості та виклики, пов'язані з PdM на основі ШІ.

В ході дослідження автори виявили такі *актуальні проблеми*: проблеми калібрування сенсорів, що призводять до систематичних похибок даних; низька роздільність або пропущені значення у часових рядах даних. Для вирішення цих проблем було *запропоновано рішення*: використання просунутих методів для заповнення пропущених даних; розробка протоколів калібрування для забезпечення точності сенсорів.

М. Achouch, М. Dimitrova, К. Ziane та інші розглядають прогнозне технічне обслуговування як концепцію, притаманну епісі четвертої промислової революції, яке сьогодні відіграє ключову роль у стійкому виробництві та виробничих системах завдяки впровадженню цифрової версії обслуговування машин [15]. Авторами зазначено, що дані, отримані з виробничих процесів, зросли експоненціально завдяки поширенню сенсорних технологій. Навіть якщо Maintenance 4.0 стикається з організаційними, фінансовими проблемами чи навіть проблемами джерела даних і ремонту машин, воно залишається сильною стороною для компаній, які його

використовують. Це дозволяє мінімізувати час простою машини та пов'язані з цим витрати, максимізувати життєвий цикл машини та покращити якість і швидкість виробництва. Цей підхід зазвичай характеризується дуже точним робочим процесом, починаючи з розуміння проекту та збору даних і закінчуючи етапом прийняття рішень.

У дослідженні дано вичерпний огляд літератури щодо методів і прикладних інструментів для інтелектуальних моделей прогнозованого технічного обслуговування в Industry 4.0 шляхом визначення та класифікації життєвого циклу проектів технічного обслуговування та викликів, що виникають, а також представлено моделі, пов'язані з цим типом технічного обслуговування: технічне обслуговування на основі стану, прогнозування та управління працездатністю (англ. Prognostics and Health Management, PHM) і залишковий термін корисного використання (англ. Remaining Useful Life, RUL). Представлено новий промисловий робочий процес прогностичного обслуговування, включаючи етап підтримки прийняття рішень, на якому дано рекомендації щодо платформи прогностичного обслуговування. Ця платформа забезпечує керування та безперервний обмін даними між обладнанням протягом усього його життєвого циклу в контексті інтелектуального обслуговування.

Проведене дослідження дозволило авторам виявити наступні *проблеми*: фрагментарні та неповні канали збору даних через недостатню інтеграцію IoT-систем; відсутність синхронізації даних у реальному часі. *Запропоновані рішення*: використання розподілених систем для збору та синхронізації даних у реальному часі; обробка та очищення даних, аналіз основних режимів роботи, для яких характерні відмови у роботі механізмів.

М. Patel, J. Vasa та В. Patel, досліджуючи роботу PdM, відмітили важливість датчиків та інструментів, які складають основу PdM [16]. У статті розглядаються обмеження PdM і важливість технічного обслуговування, орієнтованого на надійність (англ. Reliability-centered Maintenance, RCM). Переваги PdM включають скорочення часу простою, зниження експлуатаційних витрат і оптимізацію

ефективності. Однак дослідники виявили проблеми PdM, такі як занепокоєння щодо якості даних, потреба в початкових інвестиціях і потреба в навчанні кваліфікованих працівників. Авторами досліджено, як PdM поєднується з Індустрією 4.0, що призведе до революції у виробництві через фабрики та технологію цифрових близнюків. Зроблено висновок, що PdM поширить свій вплив на такі сектори, як охорона здоров'я, транспорт та енергетика, гарантуючи надійність життєво важливих систем у цих областях.

Проведене дослідження виявило *проблеми проблеми*: шум і неконсистентність у даних сенсорів; недостатня деталізація даних для точних прогнозів. Для їх подолання було *запропоновано рішення*: використання методів очищення даних для усунення шуму; проектування сенсорних систем для збору високоточних даних.

О. Serradilla, Е. Zugasti та U. Zurutuza присвятили свою роботу огляду та систематизації існуючих архітектур в PdM [17]. Їх робота була спрямована на полегшення вибору найбільш підходящої архітектури для кожного випадку використання шляхом огляду найсучасніших архітектур глибокого навчання, й їх інтеграції з етапами прогнозованого технічного обслуговування, щоб відповідати вимогам промислових компаній – виявлення аномалій, аналіз причин, оцінка залишкового терміну корисного використання. Серед *виявлених проблем*: висока варіативність даних через операційні відмінності між промисловими установками; шум і надмірні дані, що ускладнюють витягування особливостей для моделей глибокого навчання. *Запропоновані рішення*: використання методів інженерії особливостей для зменшення шуму та розмірності даних; стандартизація процесів попередньої обробки даних.

А. Т. Keleko, В. Kamsu-Foguem, R. Н. Ngouna та А. Tongne присвятили свою роботу вивченню питань промислового обслуговування, однієї з критичних рушійних сил Індустрії 4.0 (I4.0), що сприяло появі нових промислових викликів [18]. У цьому контексті прогнозне технічне обслуговування 4.0 (PdM4.0) досягло значного прогресу, надаючи кілька потенційних переваг, серед яких:

підвищення продуктивності, особливо за рахунок покращення доступності та якості та забезпечення економії коштів завдяки автоматизованим процесам моніторингу виробничих систем, ранній виявлення несправностей, скорочення часу простою машин і прогнозування ресурсу обладнання.

Автори проаналізували фактори, які можуть перешкоджати успішному використанню систем на основі штучного інтелекту в І4.0, зокрема процес збору даних, потенційний вплив етики, соціально-економічні проблеми та прозорість для всіх зацікавлених сторін і запропонували власне визначення надійного ШІ для І4.0. *Виявлені проблеми:* неконсистентність у зборі даних у розподілених системах; проблеми обробки даних у реальному часі через затримки в мережі. *Запропоновані рішення:* використання алгоритмів попередньої обробки даних у реальному часі; інтеграція розподілених систем управління даними.

К. Shen, Н. Ong, D.t Niyato та С. Yuen дослідили застосування безмодельного алгоритму Deep Reinforcement Learning для прогнозованого обслуговування обладнання та виявили причини переходу до такого підходу [19]. Вихід з ладу критично важливого обладнання призведе до зупинки виробництва та грошових втрат. Ризик незапланованого простою можна мінімізувати за допомогою прогнозного технічного обслуговування обладнання, що приносить дохід, щоб забезпечити оптимальну продуктивність і безпечну роботу обладнання. Однак підвищена сенсоризація обладнання створює потік даних, і використання лише існуючої прогностичної моделі на основі машинного навчання стає недостатньою для своєчасного прогнозування стану обладнання.

Автори пропонують безмодельний алгоритм Deep Reinforcement Learning для прогнозованого обслуговування обладнання в контексті сенсорної мережі на основі обладнання. У кожному обладнанні сенсорний пристрій збирає необроблені дані сенсора, а стан справності обладнання аналізується на наявність аномальних подій. На відміну від традиційних регресійних моделей чорної скриньки, запропонований алгоритм самостійно вивчає оптимальну політику обслуговування та надає дієві рекомендації для кожного обладнання. *Зазначені проблеми:* шум у сенсорах,

викликаний зовнішніми впливами; неповні потоки даних через проблеми з підключенням IoT-пристроїв. *Запропоновані рішення:* використання фільтрів попередньої обробки для очищення шумних даних; застосування моделей підкріплення для роботи з неповними даними.

Узагальнюючи проведений аналіз наукових досліджень, можна вказати наступні ключові виклики у роботі з даними для прогнозного обслуговування:

– *проблеми якості даних:* шум і неконсистентність у сенсорних даних, пропущені або недостатні дані через проблеми з IoT-системами;

– *проблеми кількості даних:* нестача маркованих даних через рідкісні випадки відмов, дисбаланс у даних (домінування нормальної роботи над несправностями).

Для покращення точності прогнозів і ефективності систем PdM дослідники пропонують використовувати сучасні підходи, такі як алгоритми заповнення пропущених даних, інженерія особливостей, стандартизація даних і навчання на основі малих вибірок для подолання зазначених викликів.

1.4 Проблеми прогнозного технічного обслуговування технологічних ліній

Прогнозне технічне обслуговування та технічне обслуговування на основі стану охоплює різні сфери, у даній роботі передбачено розробку проєкту для проведення аналізу показників датчиків технологічних ліній у важкій промисловості. Від локомотивів до вітрових турбін і сонячних електростанцій, дослідники даних, близькі до цих галузей, першими перетворюють дані, зібрані на цих машинах, на послідовні джерела економічної цінності. Проаналізуємо проблеми, пов'язані зі створенням і впровадженням аналітики з використанням даних обладнання саме у цій сфері [20].

Машини та компоненти неминуче зношуються, деградують і ламаються. Ці поломки можуть коштувати дорого і негативно впливати на основні показники

продуктивності. Компанії, які використовують важке обладнання, розробляють стратегії надійності, щоб справлятися з зносом і поломками, одночасно зберігаючи продуктивність. Наприклад, періодична заміна масла є частиною стратегії надійності. Рішення вимикати частину обладнання до відмови також є правильною стратегією надійності. З огляду на те, що надійність важлива для компаній у багатьох галузях промисловості, фахівців відділів Data Science надають більш глибокий аналіз прогнозного технічного обслуговування, що є одним із технічних способів, за допомогою яких можна допомогти підвищити надійність обладнання.

Прогнозне обслуговування базується на математичних моделях, які аналізують великі обсяги даних показників технологічних ліній. Сигнали майбутньої несправності, наприклад, підвищення температури або падіння тиску, що призводить до збою машини, можуть бути зафіксовані даними датчиків. Прогнозна модель (модель прогнозування відмов) зосереджена на якнайшвидшому виявленні цих сигнатур. Якщо проблему виявити досить рано, ремонт може бути незначним, наприклад, затягнути болт чи замінити мастило. Чим довше потенційна несправність залишається непоміченою, тим дорожчим виявиться ремонт [21].

Розглянемо базові етапи прогнозного технічного обслуговування технологічних ліній, зображені на рисунку 1.3. Точка P позначає початок несправності, точка D – момент, коли несправність можна помітити завдяки даним датчиків, точка Z – момент, коли весь механізм ламається. Примітно, що D відрізняється від P , що вказує на те, що фактичний початок несправного компонента може відбутися значно раніше, ніж це фактично можна виявити за наявними даними.

Період між виявленням проблеми (D) і повною відмовою компонента (Z) назвемо інтервалом DZ . Саме в цей період можна спробувати вжити коригувальні дії для зменшення загальних витрат. Різні компоненти матимуть різні криві i , отже, різні інтервали DZ . Наприклад, один несправний болт на деяких машинах може бути не виявлений до поломки. У цьому випадку точка D дорівнює точці Z .

Прогнозне технічне обслуговування пов'язане з переміщенням точки D на якомога більш ранній час [22].

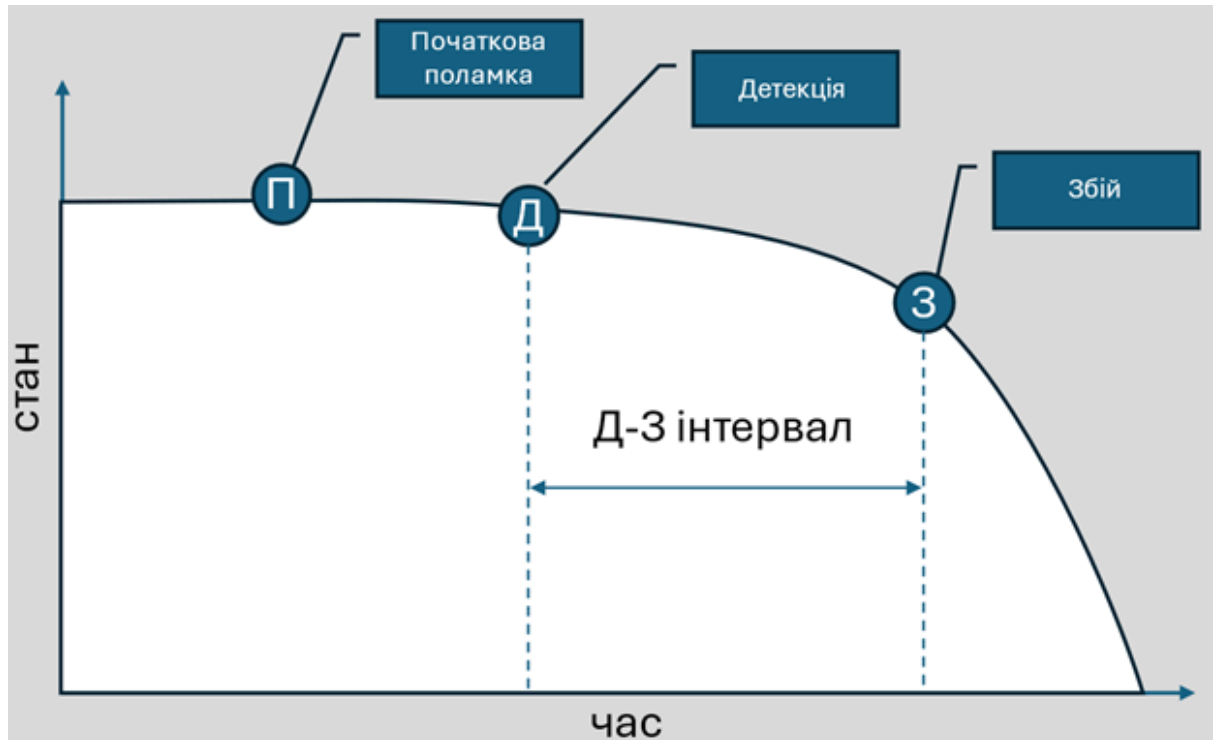


Рисунок 1.3 – Важливі етапи прогнозного технічного обслуговування

Як і у випадку з багатьма проблемами Data Science, суть вирішення проблеми прогнозованого обслуговування включає збір даних, проведення аналізу, створення та розгортання моделі, а також відстеження результатів і відгуків, щоб переконатися, що модель працює належним чином [23]. Багато технічних і статистичних проблем роблять це складним завданням. Перейдемо до аналізу цих проблем.

1. Проблеми з даними та впровадженням.

1.1. Відсутність повної та якісної інформації про несправності є, мабуть, найважчою проблемою для вирішення. На відміну від датчиків, які автоматично збирають дані, документування несправностей та їх усунення зазвичай залежить від механіків цеху, які ведуть цю документацію. Не дивно, що якість даних для Data Science не є пріоритетом для більшості механіків. Деякі підприємства працюють і

на паперових носіях. Це додає ще один рівень складності для отримання потрібних даних у руки спеціалістів із обробки даних.

1.2. Для передачі даних багато мобільних машин покладаються на стільникове або супутникове з'єднання. Для старих немобільних машин надсилання даних у хмару часто означає модернізацію старішого обладнання. В обох випадках можлива втрата даних і з'єднання може бути дорогим. Це створює дані, які можуть мати пропуски, бути невідповідними, містити дублікати та можуть змусити шукати компроміси щодо того, які дані збирати, ще до того, як дослідник даних побачить зразки даних. Крім того, критична або нестабільна робота машини також може спричинити проблеми з датчиками, створюючи сценарій, за якого пропуски в даних існують саме в критичні періоди, коли дані потрібні.

1.3. Крім підключення, конфігурації датчиків також є джерелом проблем, оскільки не всі датчики встановлюються абсолютно однаково навіть на однакових типах машин. Це спричиняє проблеми з моделюванням, оскільки може знадобитися певний тип центрального калібрування, перш ніж модель можна буде впевнено застосувати в потрібному масштабі. Крім того, не всі компоненти мають датчики, які можна використовувати для прогнозного обслуговування. Багато частин не мають датчиків, а інші частини можуть мати датчики, які не слугують для цілей прогнозованого технічного обслуговування [24].

1.4. Опис поломок часто заповнюється персоналом вручну. Інколи наприкінці зміни, або після того, як поломка була виправлена чи машина було зупинена [25]. В одному й тому ж списку можуть бути як поломки, які можна прогнозувати, так і регулярні дії, наприклад, заміна лампочок, але класифікуються вони однаково, наприклад «зупинка машини №1».

2. Чисті дані не завжди показові або з ними легко працювати.

2.1. Заміна не є еквівалентом поломки. Наприклад, планове технічне обслуговування, таке як заміна масла кожні 3000 миль, призводить до заміни без збоїв. Відмови деталей також призводять до поломки робочих частин. Наприклад,

спущене колесо може спричинити зіткнення, що призведе до заміни інших деталей. Наслідком цього є те, що навіть ідеальний запис про заміну деталей може не забезпечити узгоджену ціль для навчання під час створення моделі машинного навчання.

2.2. Збої високої вартості трапляються рідко. Машинна прогностика відображає медичну прогностику та аналіз виживання, коли події є рідкісними. Хоча це хороша новина для компаній, які працюють з такими машинами, фахівці Data Science можуть зіткнутися з труднощами, коли збір даних навіть за роки дає лише декілька прикладів несправностей, із якими можна працювати. Крім того, для найскладніших машин існує широкий діапазон типів поломок. Це може означати, що значення запобігання одному типу збоїв може бути незначним, але значення значно зростає, коли вдається запобігти більшій кількості збоїв [26].

2.3. На відміну від даних про рідкісні несправності, дані сигналу датчика можуть бути величезними. Датчики вібрації збирають дані багато тисяч разів на секунду. Невібраційні датчики збиратимуть дані раз на секунду або частіше. Це створює навантаження на обчислення під час проведення дослідницького аналізу, і в деяких випадках фахівцям доведеться працювати лише над підсумковими даними, що лежать в основі, а не над самими вихідними даними.

2.4. Багато джерел даних є залежними. У статистичному сенсі всі дані, що надходять від одного обладнання, є залежними. Усі дані, що надходять від груп обладнання в одній географічній зоні, є залежними. Навіть дані, створені різними частинами обладнання, але одним оператором, будуть залежними. Залежності даних впливають і можуть обмежувати підходи до моделювання та перевірки.

2.5. Контекст машини має значення. Машини старіють, працюють у жаркому та холодному кліматі, ходять у тунелі та крізь багнюку та працюють у дуже екстремальних умовах. Кожен із цих режимів роботи може змінювати сигнатури даних, що надходять з машини.

3. Досконалий прогноз сам по собі не перетворюється безпосередньо на цінність.

3.1. Якщо сигнатури збоїв виявляються моделлю, дія на основі прогнозу моделі потребує ручної роботи та логістики. Наприклад, щоб замінити несправну деталь на верстаті, необхідна нова деталь повинна бути в наявності у відповідній майстерні технічного обслуговування. Управління запасами саме по собі створює величезні проблеми. Створення правильного прогнозу та надання його таким чином, щоб забезпечити правильний подальший робочий процес, може бути складним завданням.

3.2. Проблеми прогнозного технічного обслуговування можуть бути «значними» проблемами. Великі суми в доларах, а в деяких випадках і безпека людей, пов'язані з як вжитими, так і невжитими діями. Користувачі прогнозів повинні мати можливість довіряти прогнозам, щоб впевнено застосовувати правильні дії [27].

Дана робота присвячена прогнозному технічному обслуговуванню для технологічних ліній важкої промисловості. Успішні підходи до прогнозного технічного обслуговування та прогнозування протидіятимуть деяким з описаних проблем безпосередньо та обійдуть інші.

1.5 Постановка задачі

Провівши аналіз підходів до прогнозного технічного обслуговування технологічних ліній було зроблено висновок про необхідність розробки інформаційної системи аналізу сенсорів технологічних ліній для виявлення проблемних даних, які впливають на якість прогнозування їх поломок.

Створення інформаційної системи передбачає створення вебзастосунку для аналізу сирих даних показників, створення звіту з проведеного аналізу, зменшення часу на створення нового звіту й відображення вже існуючих звітів. Для цього необхідно необхідно реалізувати наступне:

- завантаження файлів з даними показників сенсорів;
- проведення кореляційного аналізу та розрахунок показників описової і контекстної статистики;
- створення звіту з проведеного аналізу на невикористовуваних даних, з оновленням даних та перегляд і використання існуючих звітів;
- комунікація з клієнтом на базі отриманого аналізу з наданням графічного та табличного формату аналізу і виявлених проблем за даними;
- аргументоване створення конфігурації для наступних кроків роботи за даними: перехід від стандартизованих правил до “dataknowledge based configuration”.

Об’єктом дослідження є процес процес технічного обслуговування технологічних ліній.

Предметом дослідження є програмні засоби, методи та моделі дескриптивного і предикативного аналізу показників сенсорів технологічних ліній.

Мета дослідження – підвищення точності прогнозування поломок технологічних ліній важкої промисловості шляхом створення інформаційної системи аналізу показників сенсорів із використанням методів описової, контекстної статистики та кореляційного аналізу даних.

Досягнення поставленої мети обумовлює необхідність вирішення наступних **завдань**:

- дослідити теоретичні засади прогнозного обслуговування технологічних ліній, проаналізувати існуючі дослідження та рішення технічного обслуговування;
- обґрунтувати вибір моделей і методів прогнозного технічного обслуговування та інструментальних засобів розробки інформаційної системи;
- здійснити моделювання, проектування, програмну реалізацію системи аналізу показників сенсорів технологічних ліній та оцінити її якість.

Висновки до розділу 1

Досліджено теоретичні засади прогнозного обслуговування технологічних ліній. Установлено, що технологічна лінія є сукупністю машин та апаратів, сполучених транспортними пристроями, призначених для виконання певного технологічного процесу. В умовах високих темпів цифровізації суспільства широко розповсюдженим у різних сферах виробництва є впровадження у сферу управління та обслуговування технологічних ліній передових інформаційних технологій, які базуються на застосуванні методів та моделей штучного інтелекту, машинного навчання та Інтернету речей.

Установлено, що технічне обслуговування технологічної лінії є важливим аспектом її надійної та ефективної роботи. Прогресивна стратегія обслуговування технологічних ліній базується на RCM-методології, спрямованій на забезпечення надійності обладнання. Головним принципом цієї методології є недопущення відхилення параметрів стану обладнання технологічних ліній до тих значень, які можуть призвести до її поломки. RCM-методологія включає іноваційні підходи до технічного обслуговування технологічних ліній: 1) прогнозне технічне обслуговування (Predictive Maintenance, PdM) – дозволяє визначити стан технологічної лінії, яка знаходиться в експлуатації та оцінити, коли слід провести технічне обслуговування на основі сукупних даних датчиків і тенденції для прогнозування майбутньої деградації та відмови обладнання; 2) прогнозне технічне обслуговування на основі стану (Condition-based Predictive Maintenance, CBM) – використовує дані, зібрані під час моніторингу, щоб виконати технічне обслуговування саме в той момент, коли це необхідно, і до того, як станеться критичний збій.

Здійснений аналіз існуючих рішень дозволив виявити проблеми технічного обслуговування у важкій промисловості. Моделі, які здійснюють прогнозне технічне обслуговування технологічних ліній, постійно вдосконалюються, їх розробкою та супроводом займаються фахівці відділів Data Science. Однак є

проблема, обумовлена різними технічними причинами, що потребують корегування та оновлення параметрів моделей. З цією метою здійснюється прогнозне технічне обслуговування на основі стану, яке передбачає аналіз даних датчиків із метою виявлення проблемних даних та прогнозування можливих проблем у обслуговуванні технологічних ліній. З урахуванням цієї аналітики відбувається оновлення моделей та забезпечується уникнення збоїв у їх роботі.

Проведений аналіз останніх наукових досліджень та публікацій дозволив виявити ключові проблеми у роботі з даними з даними для прогнозного обслуговування: 1) Проблеми якості даних: шум і неконсистентність у сенсорних даних, пропущені або недостатні дані через проблеми з IoT-системами; 2) проблеми кількості даних: нестача маркованих даних через рідкісні випадки відмов, дисбаланс у даних (домінування нормальної роботи над несправностями). Для покращення точності прогнозів і ефективності систем PdM дослідники пропонують використовувати сучасні підходи, такі як алгоритми заповнення пропущених даних, інженерія особливостей, стандартизація даних і навчання на основі малих вибірок для подолання зазначених викликів.

2 МОДЕЛІ ТА МЕТОДИ ПРОГНОЗНОГО ТЕХНІЧНОГО ОБСЛУГОВУВАННЯ НА ОСНОВІ СТАНУ

Здійснений аналіз проблем прогнозного технічного обслуговування технологічних ліній та наявних досліджень і рішень дозволяє описати основні підходи до аналітики показників датчиків технологічних ліній, які дозволяють реалізувати їх прогностичне технічне обслуговування у важкій промисловості. Робота з даними, а саме підготовка, очищення, аналіз даних може займати більше 70% часу, затраченого на вирішення задачі. Якість роботи моделі залежить від якості даних і налаштувань моделі, які базуються на глибокому розумінні фахівцем специфічності кожного досліджуваного випадку.

Коли йдеться про спілкування з клієнтом, дуже важливо мати змогу ставити конкретні питання, посилаючись на цифрові дані та візуалізації. Це дозволяє отримати всю необхідну інформацію, яка враховує специфіку об'єкту, задачі прогнозування та адаптувати очікування клієнта до реальності його даних. У даному дослідженні для аналізу даних показників датчиків було обрано наступні методи: описова статистика; специфічна контекстна статистика; кореляційний аналіз.

2.1 Виявлення аномалій

Виявлення аномалій (англ. Anomaly detection) допомагає знайти шаблони в конкретних даних, які не відповідають очікуваній поведінці у роботі обладнання технологічних ліній. Ці невідповідності називаються аномаліями, викидами, винятками тощо у різних областях. Аномалія виявляється, коли вхідні дані показують відмінності порівняно зі звичайними умовами роботи обладнання. Виявлення аномалії не означає помилкових даних, оскільки це може бути пов'язано з новою нормальною функцією, до якої алгоритм виявлення аномалії не адаптувався або не пов'язаний з історичними даними [28, 29].

Аномалія може бути викликана кількома факторами; деякі з них пов'язані з помилками в системі збору даних, такими як несправність датчика, низький заряд батареї, помилки під час передачі даних, тоді як інші аномалії можуть бути викликані несправністю промислового обладнання або подіями, такими як зміни у технологічній лінії або лікувальна зупинка. Аномалії, спричинені помилками датчика, можуть призвести до неправильної інтерпретації даних.

Аномалії можна класифікувати на: точкові аномалії, коли одна точка даних значно відрізняється від своїх сусідів; поведінкові або колективні аномалії, коли модель даних відрізняється від очікуваної поведінки; і контекстуальні аномалії, коли шаблон даних можна очікувати, але в іншому контексті. Існує два типи підходів до обробки аномалій даних щодо архітектури рішення: 1) централізовані підходи, коли обчислювальний процес повністю здійснюється на одному обладнанні (наприклад, віддалений сервер); 2) розподілені рішення, де кілька етапів обчислювального процесу здійснюються в різних компонентах (наприклад, периферійні пристрої, хмара тощо).

Що стосується методологій виявлення аномалій, їх можна розділити на дві великі групи: статистичні підходи та підходи на основі машинного навчання (англ. Machine Learning, ML). Статистичні підходи спрямовані на визначення аномалій на основі розподілу змінних під час робочого процесу, тоді як ML належить до сфери штучного інтелекту (AI) і надає методології для обробки даних великої розмірності та вилучення прихованих зв'язків між нелінійними та складними даними.

Підходи до виявлення аномалій на основі статистики відповідають статистичній моделі для очікуваної поведінки заданих точок даних, які можна використовувати для визначення того, чи належить невидима точка даних до прогнозної моделі чи ні. Оцінка викиду на основі гістограми – це підхід, який визначає ступінь викидів шляхом побудови гістограм для кожної функції, а потім для кожної функції вимірюється щільність гістограми [30]. Це було застосовано для виявлення аномалій у нафтовій і газовій промисловості [31].

Застосовують також двоетапне виявлення аномалій. На першому етапі виявлення аналізують два аргументи: ступінь відхилення від норми значення датчика та попередні значення, усереднені для цього датчика. Другий етап розглядає дані від датчиків одного типу в різних місцях і їх коефіцієнт кореляції. Метою багатоетапного виявлення є виявлення локальних аномальних точок у кожному датчику та порівняння результату з корельованими датчиками, щоб визначити, чи викликано аномальне виявлення зовнішніми умовами чи несправністю датчика. Результати показують, що такий метод виявлення аномалій досягає вищої точності порівняно з традиційними методами, хоч і потребує вищих обчислювальних витрат. Цей підхід досяг кращої продуктивності щодо точності виявлення та часу обчислення для великих обсягів даних.

2.2 Описова статистика

Для вибірки значень показників кожного сенсора з урахуванням специфіки технологічної лінії розраховуються описані нижче статистичні показники.

1. Кількість окремих значень (англ. Distinct values number): скільки варіантів значень показників нараховується в даних. Не плутати з унікальними значеннями: кожне зі значень може зустрічатися багато разів. Якщо цих, окремих, значень мало – на тисячі екземплярів лише десяток значень – змінна може бути індикативною, категорійною або такою, що вже пройшла обробку.

2. Відсоток окремих значень (англ. Distinct values percent) – ще один спосіб оцінити різноманіття даних. Якщо в нас є 100 екземплярів, кількість окремих значень дорівнює 2 і відсоток окремих значень теж дорівнює 2, ми дивимося на бінарний сенсор, де по половині значень припадає на кожний варіант.

3. Кількість та відсоток нульових значень показників сенсора (англ. Zero values number, Zero values percent) – з практичного досвіду пороговим значенням було обрано 10%. Сенсори з більшою кількістю нульових значень могли бути потенційно проблемними, адже здебільшого ці сенсори передаються дані, що

заміряють процеси (температуру, швидкість, оберти, тиск тощо) і в реальному житті дуже зрідка мають нульові значення.

4. Кількість та відсоток від'ємних значень показників сенсора (англ. Negative values number, Negative values percent) – сама по собі наявність від'ємних значень є неочікуваною для досліджуваних даних, проте, коли всі дані такі, можна просто взяти їх абсолютні значення (по модулю). Якщо ж частина даних від'ємна, це сигнал, що від клієнта потрібна додаткова інформація.

5. Кількість та відсоток наявних значень показників сенсора (англ. Existing values number, Existing values percent) – для тих даних, де збір даних для всіх сенсорів уніфіковано, підраховується кількість та відсоток наявних значень.

6. Кількість та відсоток пропущених значень (англ. Missing values number, Missing values percent) – для тих даних, де збір даних для всіх сенсорів уніфіковано, підраховується кількість та відсоток відсутніх значень.

7. Загальна кількість значень показників сенсора N (англ. Total number of values) – кількість значень для цього сенсора на всьому проміжку часу.

8. Мінімальне значення показників сенсора X_{\min} (англ. Minimum value) – мінімальне значення даних сенсора на обраному проміжку часу.

9. Медіана (англ. Median) – медіанне значення показників сенсора на обраному проміжку часу.

10. Середнє арифметичне (англ. Average tag value) – середнє арифметичне значення показників сенсора на обраному проміжку часу.

11. Максимальне значення X_{\max} (англ. Maximum value) – максимальне значення показників сенсора на обраному проміжку часу.

12. Процентилі (X_5 , X_{25} , X_{75} , X_{95}) – значення, які розділяють ранжовані значення показників сенсора на групи із вказаною вірогідністю (5%, 25%, 75%, 95%).

13. Міжквартильний розмах показників сенсора IQR – різниця між X_{75} -м та X_{25} -м центилями:

$$IPQ = X_{75} - X_{25}. \quad (2.1)$$

14. Розмах значень показників сенсора (англ. Range of values) – різниця між найбільшим та найменшим значенням показників сенсора:

$$R = X_{\max} - X_{\min}. \quad (2.2)$$

15. Середнє абсолютне відхилення показників сенсора (англ. Mean absolute deviation, MAD) – міра мінливості, яка характеризує розсіяння значень показників сенсора навколо середнього арифметичного значення:

$$MAD = \frac{1}{N} \sum_{i=1}^N |x_i - \bar{x}|, \quad (2.3)$$

де \bar{x} – середнє арифметичне значення, N – кількість значень показників сенсора, x_i – i -те значення показника сенсора.

16. Дисперсія (англ. Variance) – міра розсіяння значень показників відносно середнього значення вибірки:

$$\sigma^2 = \frac{(x_i - \bar{x})^2}{N - 1}, \quad (2.4)$$

де \bar{x} – середнє арифметичне значення, N – кількість значень показників сенсора, x_i – i -те значення показника сенсора.

17. Стандартне (середньоквадратичне) відхилення (англ. Standard deviation) – корінь квадратний із дисперсії:

$$\sigma = \sqrt{\frac{(x_i - \bar{x})^2}{N - 1}}. \quad (2.5)$$

18. Коефіцієнт варіації (англ. Coefficient of variation) – статистичний оцінний показник, що характеризує співвідношення середньоквадратичного відхилення та середнього значення аналізованого показника:

$$CM = \frac{\sigma}{\bar{x}}, \quad (2.6)$$

де \bar{x} – середнє значення показника, σ – середньоквадратичне відхилення.

19. Коефіцієнт асиметрії (англ. Skewness) – числова характеристика, яка характеризує асиметричність розподілу значень показника сенсора:

$$As = \frac{1}{N-1} \sum_{i=1}^N \left(\frac{x_i - \bar{x}}{\sigma} \right)^3, \quad (2.7)$$

де \bar{x} – середнє значення показника, σ – середньоквадратичне відхилення, N – кількість значень показників сенсора, x_i – i -те значення показника сенсора.

20. Коефіцієнт ексцесу (англ. Kurtosis) – характеризує «крутість», тобто, стрімкість підвищення кривої розподілу значень показників сенсора у порівнянні з нормальним розподілом:

$$Ex = \frac{1}{N-1} \sum_{i=1}^N \left(\frac{x_i - \bar{x}}{\sigma} \right)^4 - 3, \quad (2.8)$$

де \bar{x} – середнє значення показника, σ – середньоквадратичне відхилення, N – кількість значень показників сенсора, x_i – i -те значення показника сенсора.

21. Графічна візуалізація надається для розподілу значень сенсора (англ. Tags Values Distribution) – поруч з графіком надається і таблиця значень і кількості входжень кожного для 20 найуживаніших значень.

2.3 Специфічна контекстна статистика

Дані на підприємствах збираються та накопичуються у різний спосіб. В одному й тому ж цеху на певному обладнанні можуть стояти автоматизовані системи збору даних, а на іншому – певні сенсори підключені окремо, якимсь обладнання може мати вбудовані системи від виробника.

Культура зберігання та накопичування даних ще знаходиться у стані формування, і на великих, старих підприємствах це особливо помітно. Обладнання працює роками, регулярно відбуваються планові роботи, при цьому дані збираються відносно нещодавно, історія поломок і робіт збирається в ручному режимі, журнали або електронні таблиці заповнюються в кінці дня, зміни або після закінчення робіт.

Дані можуть збиратися нерівномірно: частина пишеться весь час, інші – тільки під час зміни значень, після технічних зупинок частина сенсорів підключається автоматично при запуску обладнання, інші – треба підключити і ввімкнути запис в базу даних «вручну».

Задля збереження фінансових та машинних ресурсів на багатьох підприємствах відбувається автоматичне архівування даних через певних проміжок часу. Таким чином дані з частотою в кілька хвилин через півроку перетворюються на дані, де значення зберігаються у вигляді «середнє за 4 – 12 годин». Це звільняє дуже багато місця на серверах, проте інформацію втрачено – наприклад, кілька показників температури на добу нічого не скажуть про патерни роботи обладнання в штатному режимі, на початку поломки, перед зупинкою обладнання і на початку нового циклу після ремонту.

Чим більше підприємство, тим більш розгалужена ієрархія і збільшений вплив людського фактору. Щоб підвищити ефективність своєї роботи аналітики, фахівці Data Science, мають знайти можливість ставити максимально

конкретизовані питання, щоб менеджмент підприємства-клієнта міг перенаправити запит до компетентних фахівців.

З метою якнайглибше вивчити особливості та якість історичних даних, переконатися, які дані та в якому форматі є сенс отримувати в режимі живого потоку, здійснюється збір великої кількості специфічної (контекстної) статистики. Цей аналіз відбувається кілька разів, поки інформація з сенсорів не очищена до того стану, коли можна уніфікувати інформацію з різних джерел і переходити до наступних кроків: пошук аномалій та навчання моделей.

Здійснимо перелік специфічних (контекстних) статистичних даних на етапі аналізу історичних даних для кожного сенсора.

1. *Часові проміжки між сусідніми даними в часовому ряді* (англ. Time intervals), подаються у табличному форматі та у вигляді графіків:

– кількість окремих значень часових проміжків (Time interval distinct values number) – кількість варіантів тривалості часових інтервалів між сусідніми значеннями (з'ясовуємо, уніфіковані чи ні);

– статистика тривалості інтервалів: мінімальна, центилі (25% та 75%), медіанна, середня, максимальна, стандартне відхилення (Min, 25%, Median, Mean, 75%, Max, StD).

2. *Пропуски даних* (англ. Gaps), подаються у табличному форматі та у вигляді графіків:

– найдовша тривалість пропущених даних (англ. Longest gap) – на етапі створення звіту фахівець визначає, яка тривалість часу між сусідніми даними вважається пропуском даних (за замовчуванням – 6 годин), все, що менше, вважається інтервалом, все, що дорівнює чи більше – пробілом. У процесі роботи зі звітом цю межу можна змінювати. Варто зазначити, що ці пропуски можуть бути ознакою того, що дані в цей час не змінювалися (і не надходили згідно налаштувань збоку клієнта), або, навпаки, що в цей час дійсно дані не надходили, або зі сторони клієнта було надано часткові дані, або відбулося архівування старих даних;

– дати початку та завершення найдовшого пропуску даних (англ. Longest gap start, Longest gap end) – якщо для більшості важливих сенсорів інформація не надходила протягом одного й того ж тривалого періоду, можна просто змістити часові рамки проекту на кінець цього періоду і працювати з рештою даних;

– кількість періодів пропусків (англ. Number of gaps) – кількість окремих проміжків часу, триваліших за встановлену межу, без даних;

– загальна тривалість часу без даних в годинах та відсотках (Gaps from total time (%), Total time gaps).

3. *Завмерлі дані* (англ. Freezes), подаються у табличному форматі та у вигляді графіків:

– найдовша тривалість завмерлих даних (Longest freeze) – на етапі створення звіту фахівець визначає, яка тривалість часу надходження незмінюваних даних вважається «завмерлими даними». У процесі роботи з таблицями та візуалізаціями цю межу можна змінювати;

– дати початку та завершення найдовшого періоду завмерлих даних (англ. Longest freeze start, Longest freeze end) – в табличних даних та на візуалізації можна побачити, наскільки значущими є ці збої в якості даних на загальній тривалості даних;

– кількість періодів завмерлих даних (англ. Number of freezes) – кількість окремих проміжків часу завмерлих даних, триваліших за встановлену межу;

– загальна тривалість часу завмерлих даних в годинах та відсотках (англ. Total time freeze (hrs), Freezes from total duration (%)).

2.4 Кореляційний аналіз показників сенсорів

Кореляційний аналіз є сукупністю методів для виявлення статистичної залежності даними сенсорів, які дають можливість отримати інформацію про ймовірність появи певних значень одного сенсора під впливом зміни значень інших

сенсорів. У якості числової характеристики ймовірного зв'язку величин використовують коефіцієнти кореляції.

Для визначення тісноти та напрямку лінійного зв'язку показників значень двох сенсорів x_i та x_j розраховують коефіцієнт кореляції Пірсона (англ. Pearson's correlation coefficient):

$$r = \frac{\text{COV}(x_i, x_j)}{\sigma_{x_i} \cdot \sigma_{x_j}}, \quad (2.9)$$

де σ_{x_i} та σ_{x_j} – стандартні відхилення показників сенсорів x_i і x_j ,

$\text{COV}(x_i, x_j)$ – коваріація показників сенсорів x_i і x_j .

Коваріація є сумісною варіацією показників значень двох сенсорів x_i та x_j , і розраховується за формулою:

$$\text{COV}(x_i, x_j) = \sum_{k=1}^n (x_{ik} - \bar{x}_i) \cdot (x_{jk} - \bar{x}_j), \quad (2.10)$$

де x_{ik} – k -те значення показника сенсора x_i , x_{jk} – k -те значення показника сенсора x_j ,

\bar{x}_i та \bar{x}_j – середні значення показників сенсорів x_i і x_j .

Коефіцієнт кореляції набуває значень у інтервалі $[-1; 1]$. Від'ємні значення коефіцієнта свідчать про наявність оберненого зв'язку між змінними: збільшення значень однієї змінної супроводжується зменшенням значень іншої змінної. Додатні значення коефіцієнта кореляції відповідають прямому зв'язку між змінними: збільшення значень однієї змінної супроводжується збільшенням значень іншої ознаки.

Кореляційний аналіз дозволяє визначати тісноту та характер зв'язку між досліджуваними величинами й кількісно оцінити ступінь не випадковості їх сумісних змін. Візуалізується кореляція в такі способи:

– теплова карта – щоб побачити загальну ситуацію з кількістю пар сенсорів з високим коефіцієнтом кореляції (мапу не повинна бути ані занадто «блідою», ані занадто темною) (рис. 2.1);

– групи сенсорів з високим коефіцієнтом кореляції, виведені у табличному вигляді, розмір групи – визначається, скільки сенсорів мають залежні сенсори, і скільки залежних сенсорів у кожній групі;

– візуалізація поведінки сенсорів у групах – графічне зображення часового ряду даних перерахованих в групі сенсорів (рис. 2.2).

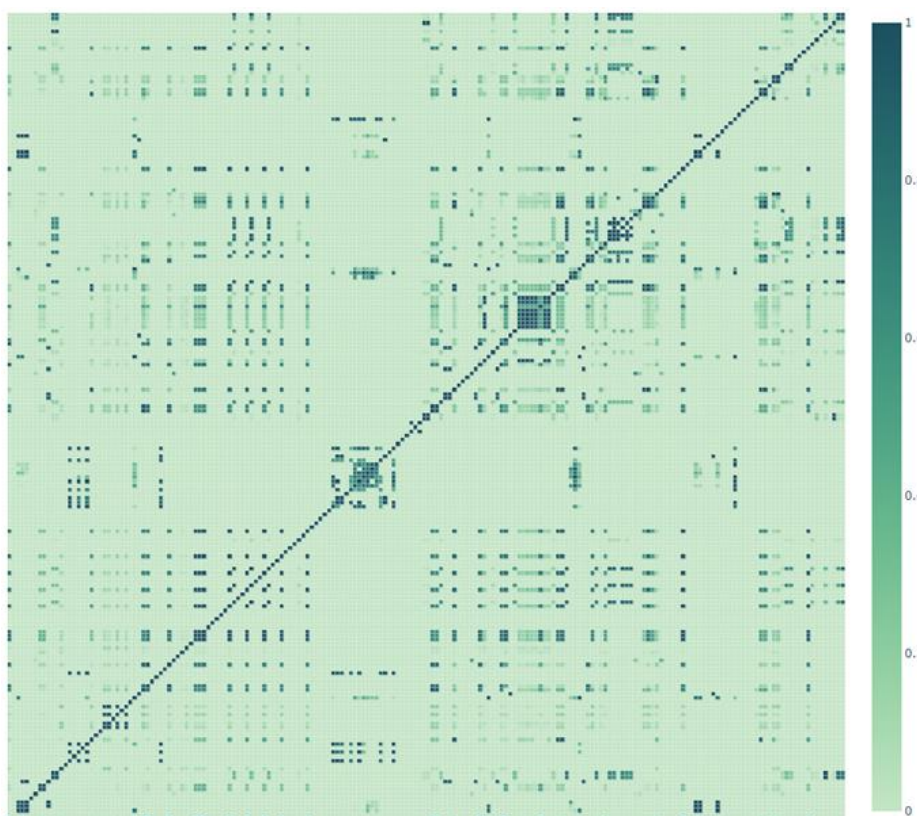


Рисунок 2.1 – Теплова карта коефіцієнтів кореляції сенсорів

На наведеному рисунку 2.1 кореляції не обчислюються для сенсорів, якщо назви їхніх часових позначок у необроблених даних відрізняються більше ніж удвічі. Високий коефіцієнт кореляції може бути зумовлений як дійсно залежністю одного механізму від іншого, так і дублюванням даних: про частину процесів може збиратися більше інформації, з сенсорів, які розташовані близько один до одного.

Наприклад, на одному вузлі можуть збирати RPM (обороти за хвилину), швидкість, вібрації, рівень шуму, температуру, тиск для ротора (надається виробником обладнання «з коробки») і вібрації, рівень шуму і температуру на зовнішньому кожусі (встановлено підприємством-клієнтом на всіх машинах). Таким чином, при визначенні сумарної сили аномалій, дані для вібрації, шуму та температури матимуть подвійну вагу. Інший приклад: зберігаються сирі дані та дані після препроцесингу для одного й того ж сенсору – туж подвійна вага аномалій перед навчанням машини.



Рисунок 2.2 – Візуалізація поведінки сенсорів у групах

Висновки до розділу 2

Здійснений аналіз дозволив обґрунтувати вибір методів та моделей декриптивної та предиктивної аналітики показників сенсорів технологічних ліній. Для виявлення проблем, які можуть впливати на якість прогнозних моделей, здійснюється аналітика «сирих» поточних даних і даних за минулі періоди, з використанням:

а) описової статистики: здійснюється поглиблений аналіз із виведенням у табличному і графічному вигляді стандартних показників описової статистики та виявлення кількості пропущених, нульових, окремих, від'ємних значень сенсорів, коефіцієнтів ексцесу, варіації й асиметрії;

б) специфічної контекстної статистики – здійснюється розрахунок із виведенням у табличному і графічному вигляді:

– часових проміжків між сусідніми даними у часовому ряді показників сенсорів: кількості окремих значень часових проміжків (уніфіковані чи ні) та статистичних показників тривалості інтервалів;

– пропусків даних: визначення кількості та тривалості пропусків, початку та завершення найдовшого пропуску даних;

– завмерлих даних: визначення кількості та тривалості завмерлих даних, початку та завершення найдовшого періоду завмерлих даних;

в) кореляційного аналізу: розрахунок коефіцієнтів кореляції для сенсорів із врахуванням можливого дублювання даних і відображенням теплової карти та виявлення груп сенсорів із високим рівнем кореляції і візуалізація поведінки сенсорів у групах у вигляді часових рядів.

3 ТЕХНОЛОГІЇ І ЗАСОБИ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Для розробки системи було обрано стек технологій, що забезпечує ефективну обробку, зберігання та візуалізацію даних, а також можливість реалізації статистичних методів і масштабовану обробку великих обсягів даних. Основу стеку складає мова програмування Python, яка завдяки своїй гнучкості та широкій екосистемі бібліотек дозволяє реалізувати різні аспекти роботи з даними.

3.1 Мова програмування та бібліотеки Python

Python – основна мова програмування, використана для розробки системи. Вона має потужні інструменти для обробки даних, створення візуалізацій та підтримки роботи з базами даних. Python також підтримує багатопоточність та багатопроцесорну обробку, що є важливим для роботи з великими обсягами даних.

Python надає багато переваг при розробці:

- розробники можуть легко читати та розуміти програми на Python, оскільки мова має базовий синтаксис, схожий на синтаксис англійської;
- Python допомагає розробникам бути більш продуктивними, оскільки вони можуть писати програми на Python, використовуючи менше рядків коду, ніж іншими мовами;
- Python має велику стандартну бібліотеку, що містить багаторазові коди практично для будь-якого завдання. В результаті розробникам не потрібно писати код із нуля;
- розробники можуть легко поєднувати Python з іншими популярними мовами програмування: Java, C та C++;
- активна спільнота Python складається з мільйонів розробників з усього світу. У разі виникнення проблем співтовариство допоможе в їх вирішенні;
- Python можна переносити на різні операційні системи: Windows, MacOS, Linux і Unix [32].

Для роботи з даними та візуалізації результатів їх аналізу обрано Python-бібліотеки Pandas, NumPy, Plotly та Pandas-profiling [33-36]. Як інструмент для паралельної обробки даних обрано бібліотеку Python multiprocessing [37].

Pandas – бібліотека для обробки та аналізу даних, яка забезпечує роботу з таблицями та підтримує різні типи даних. Pandas використовувалася для завантаження, обробки та підготовки даних до візуалізації та подальшого аналізу.

NumPy – бібліотека для наукових обчислень, що забезпечує ефективну роботу з великими масивами числових даних та підтримує різні математичні операції. NumPy активно використовувався для обробки числових даних, що є важливим для підвищення продуктивності системи.

Plotly – бібліотека для створення інтерактивних графіків і візуалізацій. Plotly використовується у поєднанні з Dash для побудови інтерактивних графіків, які допомагають користувачам системи краще розуміти та аналізувати результати обробки даних.

Pandas-profiling – бібліотека для автоматизованого створення звітів про дані. Цей інструмент дозволив реалізувати базові статистичні методи аналізу, такі як описові статистики, виявлення пропущених даних, аномалій та інші.

Для обробки великих обсягів даних використовувалася бібліотека Python multiprocessing, яка дозволяє виконувати паралельні обчислення. Застосування багатопроцесорної обробки на внутрішніх серверах забезпечує оптимізацію продуктивності та знижує час виконання завдань, що є важливим для роботи з великими наборами даних.

3.2 Фреймворки Flask і Plotly Dash

Для реалізації API та побудови інтерфейсу системи було обрано мікрофреймворк *Flask*, призначений для розробки вебзастосунків на мові Python [38]. Його легкість і гнучкість дозволяють легко інтегрувати інші бібліотеки та інструменти, а також підтримують швидкий розвиток проекту та масштабування

до складних проєктів. Як мікрофрейсворк, він не має у комплекті набору інструментів та бібліотек. Однак програміст сам має можливість їх установити у залежності від поставлених задач.

Flask має багато переваг, які виділяють його серед інших фреймворків: простий синтаксис, зручні шаблони – можна швидко створювати прототипи вебзастосунків, купа інструментів для гнучкого налаштування сайтів під будь-які потреби. Під Flask написані сотні розширень та плагінів, які додають додаткові можливості – різні види автентифікації, управління базами даних та роботу з формами. І все це – безкоштовно і з відкритим кодом.

Flask Framework використовує Jinja2 – застосунок для обробки шаблонів, і Werkzeug – інструмент для роботи з WSGI (стандартом взаємодії між програмою Python, яка виконується на стороні сервера, і самим веб-сервером). Для створення ізольованого середовища Python використовується модуль Virtualenv.

Вебфреймворк Flask підтримує:

- модульне тестування – перевірку на коректність окремих модулів вихідного коду програми;
- Google App Engine — потужна платформа для створення та запуску програми в інфраструктурі Google;
- RESTfull-архітектуру - відповідний набір архітектурних принципів для створення гнучкої та масштабованої мережі.

Flask захищений від XSS-атак, коли зловмисник впроваджує на сторінку шкідливий скрипт, і працює з безпечними cookie. В інструменті Werkzeug передбачено простий веб-сервер.

Фреймворк Plotly Dash використовують при створення інтерактивних вебзастосунків для аналітики даних. Аналітичний програмний Python-фреймворк дозволяє швидко створювати інформаційні панелі – дашборди для браузера з використанням технологій Data Mining та штучного інтелекту [39].

Plotly Dash призначений для роботи з моделями Python, дозволяючи дослідницьким та аналітичним групам зосереджуватися на даних та моделях,

одночасно створюючи та розгортаючи готові до роботи програми й інформаційні панелі. Система Dash добре підходить для створення застосунків візуалізації даних на базі нетривіальних алгоритмів з високо настроюваними інтерфейсами користувача і чистою мовою Python.

Завдання, які зазвичай вимагають звернення до команди програмістів, застосовуючи програмний пакет Plotly Dash можна зробити з мінімальним зверненням до програмування. Він дозволяє командам дослідження даних створювати, проектувати, розгортати та безпечно керувати даними застосунків, які відповідають поставленим цілям. Компанії можуть прискорити та підвищити ефективність реалізації своїх ініціатив у галузі даних, аналітики та штучного інтелекту, не звертаючись до неспецифічних мов програмування та технологій JavaScript, CSS, CronJobs або DevOps.

Використання Dash дозволило розробити інтуїтивно зрозумілі та візуально привабливі інтерфейси для представлення даних користувачам. Завдяки інтеграції з бібліотекою Plotly, Dash забезпечує можливість динамічної побудови графіків та інтерактивних візуалізацій.

3.3 СУБД та сервіси для доступу й аналізу даних

Для зберігання даних та аналізу і доступу до них було використано об'єктно-реляційну СУБД PostgreSQL, хмарне сховище Amazon S3, інтерактивний сервіс SQL-запитів Amazon Athena і бібліотеку для роботи з реляційними базами даних на Python SQLAlchemy.

Об'єктно-реляційна система управління базами даних PostgreSQL використовується для зберігання структурованих даних. PostgreSQL надає надійний механізм для роботи з реляційними даними, а також забезпечує підтримку складних SQL-запитів, необхідних для обробки й аналізу даних.

PostgreSQL – це одна з найсучасніших систем управління базами даних яка підтримує об'єкти користувача і їх поведінку. Основні відмінності СУБД PostgreSQL:

- підтримує посилальну та транзакційну цілісність збереженої інформації;
- має безліч додаткових функцій, які зберігають тільки коректні data-файли;
- має розширену ємність за рахунок внутрішньої структури;
- відповідає стандарту ANSI/ISO SQL:2008.

Розмір бази, як і кількість рядків та індексів у таблиці, не обмежені. Максимальний розмір таблиці може становити до 32 ТБ. Що є корисним при роботі з Big Data наборами даних [40].

Postgres є гнучка та надійна СУБД, яка може працювати зі складними структурами та типами даних: багатовимірними масивами, json, uuid, бітовими рядками, мережевими адресами. У PostgreSQL можна користуватися вбудованими типами даних – бітовими рядками, двійковими даними, інтервалом часу – або створювати свої: базові, складові, перераховані дані, діапазон.

Amazon Athena – інтерактивний сервіс для виконання SQL-запитів по відношенню до даних, що зберігаються у хмарному сховищі Amazon S3. Використання Amazon Athena дозволяє ефективно аналізувати великі обсяги даних без необхідності попереднього переносу їх до локальної бази даних, що оптимізує використання ресурсів і прискорює обробку запитів.

Amazon Athena є безсерверним інтерактивним аналітичним сервісом, побудований на базі фреймворків з відкритим вихідним кодом і підтримує формати відкритих таблиць та файлів. Athena надає спрощений та гнучкий спосіб аналізу даних там, де вони знаходяться. Сервіс можна використовувати для обробки журналів, аналізу даних та інтерактивних запитів до них [41]. Athena масштабується автоматично і виконує запити паралельно, тому результати повертаються дуже швидко, навіть за умови виконання складних запитів на великих обсягах даних.

Amazon Athena забезпечує високу доступність, виконуючи запити за допомогою обчислювальних ресурсів, розподілених між кількома об'єктами, та автоматичної маршрутизації запитів у разі недоступності певних об'єктів. Як основне сховище Athena використовує Amazon S3, що гарантує високу доступність і надійність даних.

Amazon S3 – хмарне сховище для зберігання даних. Amazon S3 забезпечує надійне зберігання великих обсягів даних, а також підтримку інтеграції з іншими сервісами Amazon, такими як Amazon Athena, що дозволяє здійснювати аналіз даних безпосередньо у сховищі. Стійка інфраструктура Amazon S3 забезпечує надійне зберігання об'єктів. Резервні копії даних розподіляються між кількома об'єктами та кількома пристроями на кожному об'єкті [42].

У Amazon Athena можна керувати доступом до даних за допомогою політик AWS Identity and Access Management (IAM), списків контролю доступу або політик використання кошиків Amazon S3. Завдяки політикам IAM можна надавати користувачам IAM засоби точного керування кошиками S3. Контролюючи доступ до даних S3, можна заборонити користувачам надсилати запити до цього сховища через Athena. Athena також дозволяє виконувати запити до зашифрованих даних, що зберігаються в Amazon S3, та записувати зашифровані результати назад у кошик S3. Підтримується як шифрування за сервера, і шифрування за клієнта.

SQLAlchemy – бібліотека для роботи з реляційними базами даних на Python. SQLAlchemy надає об'єктно-реляційне відображення (ORM), що спрощує роботу з базами даних та забезпечує зручні інтерфейси. SQLAlchemy дозволяє при роботі з базами даних замість SQL-запитів використовувати звичні для розробника об'єкти та їх методи, що досить зручно. Бібліотека підтримує різні СУБД, управляти якими можна, використовуючи одні й ті ж універсальні методи [43].

Ще одна особливість SQLAlchemy в тому, що є можливість замінити базу даних у будь-який момент, наприклад, з SQLite на PostgreSQL. Код програми залишиться тим самим і продовжить працювати. Розробнику не доведеться нічого переписувати, крім тих місць, де вказано, яка база даних використовується. Все

завдяки тому, що бібліотека складається з двох компонентів – SQLAlchemy Core та SQLAlchemy ORM.

SQLAlchemy Core – це абстракція над SQL. Компонент вміє спілкуватися з SQL і перетворювати Python-код на зрозумілі для баз даних запити за допомогою SQL Expression Language. При цьому він розуміє, з яким видом сховища даних взаємодіє. SQLAlchemy ORM – зручний для розробника інтерфейс управління базами даних. Компонент дозволяє працювати з ними та комітити запити прямо в Python-коді – без необхідності переписувати все на мову SQL.

3.4 Опис технологій back-end і front-end

Для створення вебзастосунку системи збору та аналізу даних було використано кілька ключових технологій на бекенді. Основна мова програмування Python використовувалася для обробки даних і реалізації аналітичних функцій. Завдяки її бібліотекам та екосистемі Python є чудовим вибором для роботи з великими обсягами даних і статистичним аналізом. Бібліотеки Pandas та NumPy використовуються для обробки і маніпулювання даними. Pandas дозволяє зручно працювати з табличними даними, а NumPy забезпечує вискоелективні операції з числовими масивами. Бібліотека Pandas-profiling використовується для автоматичного створення звітів про базові статистичні характеристики даних. Це зручно для швидкого огляду основних метрик.

Fastparquet застосовується для роботи з Parquet-файлами, що оптимізує зберігання великих обсягів даних, зокрема для створення і зберігання інтервальних файлів та даних про пропуски та завмерлі дані. Формат Parquet – це поширене сховище двійкових даних, яке використовується, зокрема, у сфері Hadoop/big-data. Він забезпечує ряд переваг, що стосуються обробки великих даних, у тому числі:

- стовпчате сховище, читання тільки даних, які потрібні у даних час;
- ефективна бінарна упаковка;
- вибір алгоритмів стиснення та кодування;

- розбиття даних на файли, що дозволяє проводити паралельну обробку;
- діапазон логічних типів;
- статистика, що зберігається в метаданих, дозволяє пропускати непотрібні фрагменти;
- поділ даних із використанням структури каталогів.

Оскільки він був розроблений як частина екосистеми Hadoop, еталонна реалізація Parquet написана Java. Мета цього пакету – надати продуктивну бібліотеку для читання та запису файлів Parquet з Python. Це зробить формат Parquet ідеальним механізмом зберігання робочих процесів великих даних на основі Python. Таблична природа Parquet добре підходить для об'єктів data-frame Pandas.

Бібліотека fastparquet прискорює введення-виведення Pandas у 5 разів. Це високопродуктивна реалізація формату Parquet на Python, призначена для безперебійної роботи з кадрами даних Pandas. Вона забезпечує швидку продуктивність читання та запису, ефективне стиснення та підтримку широкого спектру типів даних.

Модуль Multiprocessing у Python надає можливості для роботи з багатопроцесорним програмуванням, дозволяючи створювати та керувати процесами, обмінюватися даними між процесами, використовувати пули процесів та інші механізми для паралельного виконання завдань.

Деякі ключові функції та класи модуля multiprocessing:

- Process: клас для створення та управління процесами, дозволяє запускати функції у новому процесі;
- Pool: клас, що надає пул процесів для виконання завдань паралельно, дозволяє керувати пулом процесів та розподіляти завдання між процесами;
- Queue: клас для обміну даними між процесами через чергу, забезпечуючи безпеку при роботі з даними, що поділяються;
- Manager: клас для управління об'єктами, що розділяються між процесами,

дозволяє створювати списки, словники, черги та інші;

- Lock, Event, Condition, Semaphore: класи для синхронізації процесів та запобігання змаганням (race conditions) при доступі до загальних ресурсів;
- Pipe: механізм обміну даними між двома процесами через двосторонній канал;
- `cpu_count()`: функція повернення кількості доступних у системі процесорів (ядер).

Використання модуля Multiprocessing дозволяє ефективно використовувати ресурси багатопроцесорної системи, прискорити виконання завдань та реалізувати паралельне виконання обчислень. Що суттєво пришвидшує процеси обробки великих наборів даних. Це рішення особливо корисне під час завантаження й обробки даних для кожного сенсора одночасно.

SQLAlchemy використовується як ORM для спрощення роботи з реляційними базами даних. Це допомагає інтегрувати різні бази даних у проєкт.

Описані технології разом створюють ефективний і гнучкий бекенд, здатний обробляти великі обсяги даних у паралельному режимі, проводити аналіз і створювати звіти.

Фронтенд вебзастосунку використовує бібліотеки Dash, Dash Bootstrap Components і Plotly для створення інтерактивного інтерфейсу, що дозволяє користувачам переглядати, налаштовувати й візуалізувати результати аналітики.

Основою для побудови інтерфейсу є бібліотека Dash – бібліотека для створення вебзастосунку на Python. Dash дозволяє швидко побудувати інтерактивний інтерфейс, який обробляє користувацькі дії і забезпечує відображення аналітичних даних. У системі аналізу даних Dash надає базові компоненти та механізми callback для забезпечення інтерактивності інтерфейсу. Застосування колбеків дає можливість у потрібні моменти часу у відповідь на дії користувача частину коду передати до іншого коду як аргумент. Що дозволяє реалізувати функції зворотного синхронного чи асинхронного виклику.

Для застосування стандартних Bootstrap-стилів і створення організованої структури інтерфейсу використовується бібліотека Dash Bootstrap Components (DBC). Компоненти, такі як Card, Row, Col, використовуються для відображення аналітичних блоків, таблиць і фільтрів, що допомагає створити зручний і сучасний вигляд інтерфейсу.

Для створення зручних інструментів керування – календарів, фільтрів, перемикачів – використано бібліотеку Dash Mantine Components (dmc). Компоненти Checkbox, Slider, Switch, ActionIcon, Button, Modal, Accordion, DateTimePicker та інші використовуються для створення зручної інтерактивності, вміщення великої кількості інформації в кожний екран прокрутки.

Для візуалізації аналітичних даних обрано бібліотеку Plotly. Бібліотека використовується для побудови інтерактивних графіків і візуалізації даних. У системі аналізу даних Plotly використовується для представлення аналітичних результатів, таких як гістограми, кореляційні карти, часові ряди з позначенням пропусків і періодів завмерлих даних. Ця бібліотека забезпечує користувачів візуальним відображенням результатів обробки, полегшуючи їх аналіз.

Висновки до розділу 3

Для створення інформаційної системи було обрано мову програмування Python, фреймворки Flask та Plotly Dash. Для роботи з таблицями, даними та візуалізації застосовано Python-бібліотеки Pandas, NumPy, Plotly. Реалізація статистичних методів здійснювалася з використанням Pandas-profiling. Python multiprocessing було застосовано для паралельної обробки великих обсягів даних на внутрішніх серверах. Створення фронтенду інформаційної системи аналізу показників сенсорів за допомогою Dash дозволяє забезпечити інтерактивність і зручність роботи з аналітичними результатами.

Для зберігання даних використано об'єктно-реляційну СУБД PostgreSQL та інтерактивний сервіс запитів Amazon Athena для аналізу даних у сховищі Amazon

S3 за допомогою SQL. Python-бібліотека SQLAlchemy використана для спрощення роботи з реляційними базами даних.

Використання вищезазначеного стеку технологій дозволяє забезпечити надійну обробку, зберігання та візуалізацію даних. Інструменти Python, такі як Flask, Dash, Pandas та SQLAlchemy, забезпечують високу продуктивність системи, а хмарні сервіси Amazon S3 та Athena допомагають масштабувати обробку даних, роблячи систему придатною для роботи з великими обсягами даних та забезпечуючи ефективність її функціонування. Використання Plotly для візуалізації та Dash Bootstrap Components для структурування інтерфейсу робить систему гнучкою та інтуїтивно зрозумілою для користувачів, полегшуючи роботу з великими наборами даних і дозволяючи швидко отримувати візуальні аналітичні інсайти.

4 РОЗРОБКА ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ АНАЛІЗУ ПОКАЗНИКІВ СЕНСОРІВ

4.1 Алгоритмізація логіки вебзастосунку

Для вебзастосунку інформаційної системи розроблено діаграму діяльності для графічного відображення алгоритму аналізу показників сенсорів технологічних ліній (рис. 4.1).

В технічне завдання для створення системи аналізу показників сенсорів технологічних ліній входили такі задачі:

- створити інтерактивний інструмент для уніфікованого аналізу сирих даних;
- знизити навантаження на основну базу даних;
- максимально зменшити час на створення нового звіту та відображення вже існуючих.

Відповідно до цього було втілено такі рішення:

- на етапі створення конфігурації звіту опис даних береться через запит “get” до початку завантаження даних;
- у базі даних системи зберігаються завантажені дані, які оновлюються та доповнюються у разі необхідності;
- у директорії звіту зберігаються результати розрахунків і дані, необхідні для створення візуалізацій: дані зберігаються в форматах .pcl та .parq, що дозволило пришвидшити роботу вебзастосунку та зменшити обсяги використовуваної пам’яті;
- всі створені звіти доступні всім користувачам системи;
- після створення конфігурації звіту система пропонує список вже існуючих звітів для цього набору даних.

Світло-зелена заливка на рисунку 4.1 позначає операції з основною базою даних, світло-блакитна – операції з базою даних застосунку.

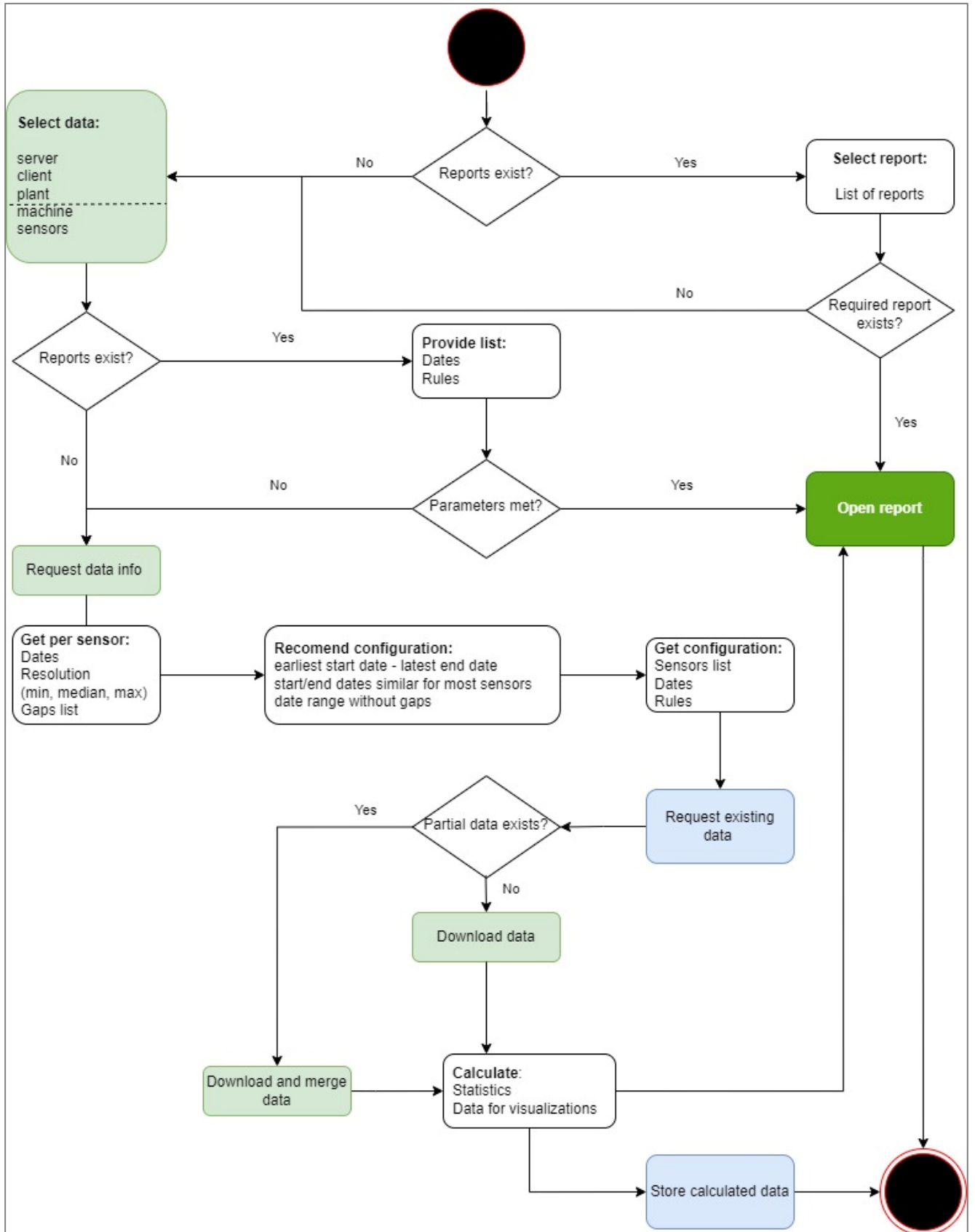


Рисунок 4.1 – Діаграма діяльності аналізу показників сенсорів

4.2 Моделювання інформаційної системи

На етапі моделювання інформаційної системи аналізу показників сенсорів технологічних ліній було розроблено діаграму сценаріїв використання системи для пояснення роботи системи та формування функціональних вимог до системи (рис. 4.2).

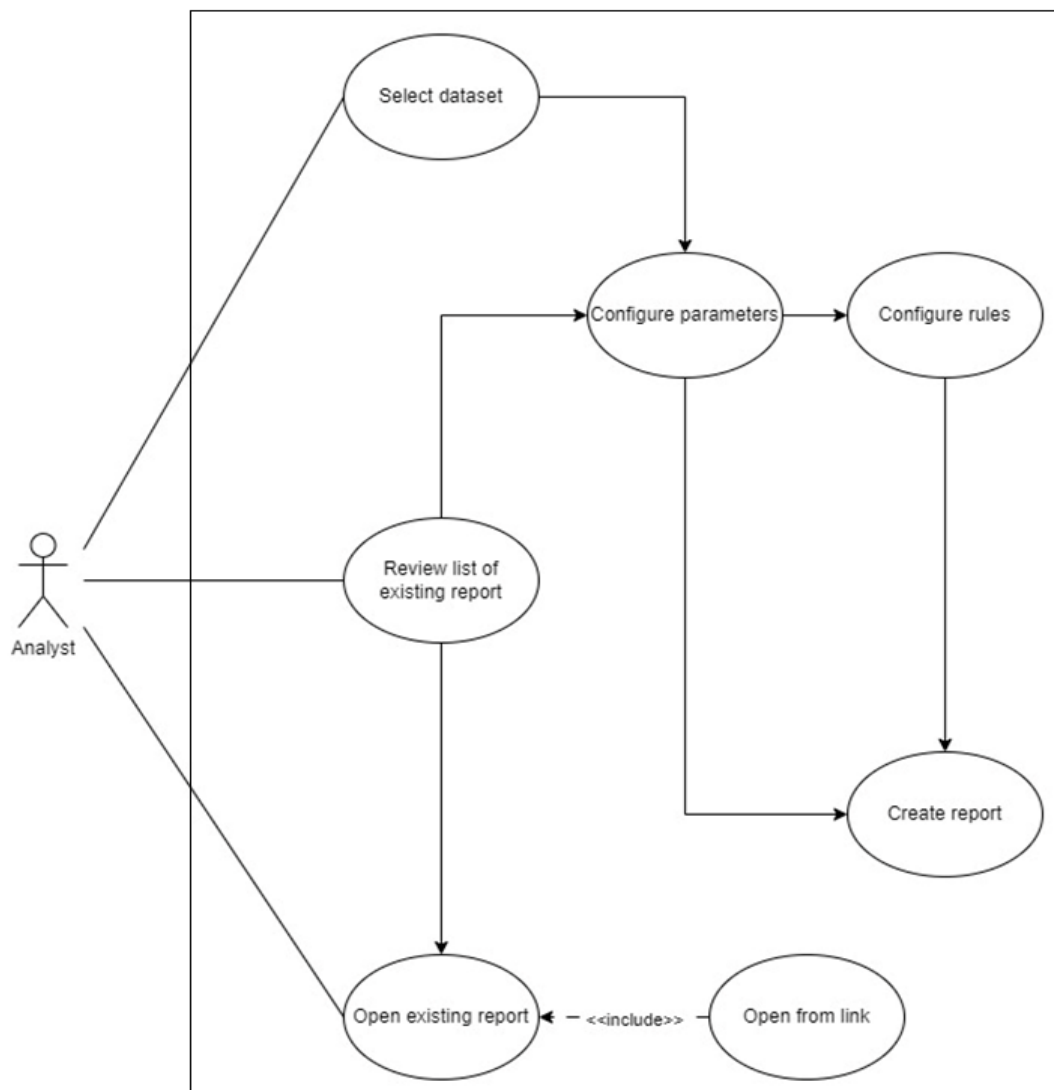


Рисунок 4.2 – Діаграма прецедентів

Взаємодія користувача з системою передбачає три сценарії використання:

- Use case 1: створення звіту на невикористовуваних даних;
- Use case 2: створення звіту з оновленням даних;

– Use case 3: використання існуючого звіту.

Use case 1 – створення звіту на невикористовуваних даних. Користувач відкриває веб-застосунок системи аналізу даних показників сенсорів, обирає файл з набором даних, часовий проміжок, конфігурує правила і генерує звіт. Необхідні дані завантажуються на сервер аналітичної системи, проводяться необхідні розрахунки. Вихідні (сирі) дані та результати аналізу зберігаються на сервері аналітичної системи, в локальній базі даних, і можуть бути використані будь-яким користувачем системи. Готовий звіт додається в таблицю на сторінці вибору звіту для перегляду і автоматично відкривається в новому вікні браузера.

Use case 2 – створення звіту з оновленням даних. Користувач відкриває веб-застосунок системи аналізу даних показників сенсорів, переглядає існуючі звіти, де наявний аналіз потрібного йому набору даних (клієнт, підприємство, машина, певні сенсори), але треба змінити перелік сенсорів, дати, правила. Користувач обирає потрібні йому налаштування, правила і генерує звіт. Система перевіряє, які дані вже є, складає перелік відсутніх проміжків, сенсорів, довантажує необхідне, поєднує набори, обраховує статистику і кореляції на оновлених даних, зберігає вихідні та обраховані дані. Готовий звіт додається в таблицю на сторінці вибору звіту для перегляду і автоматично відкривається в новому вікні браузера.

Use case 3 – використання існуючого звіту. Цей сценарій може використовуватися або коли користувач знає про існування потрібного звіту (створив його сам, отримав посилання від колеги, знайшов в переліку існуючих звітів тощо), або коли на етапі конфігурації звіту система повідомила, що для цих даних вже існують звіти і надала перелік з описами та посиланнями.

При роботі за цим сценарієм, система не проводить жодних розрахунків. Користувач відкриває готовий звіт і працює з інтерактивними дашбордами, завантажує локально таблиці, графіки тощо.

Під час моделювання інформаційної системи також було розроблено схему, яка відображає послідовність роботи з базами даних. Для роботи системи аналізу показників сенсорів технологічних ліній при виконанні операцій з даними

передбачено використання трьох баз даних та сервер аналітичної системи. Дані та метадані отримуються з хмарного середовища AWS, а після обробки зберігаються в СУБД PostgreSQL та на сервері вебзастосунку у виділеному просторі (рис. 4.3).

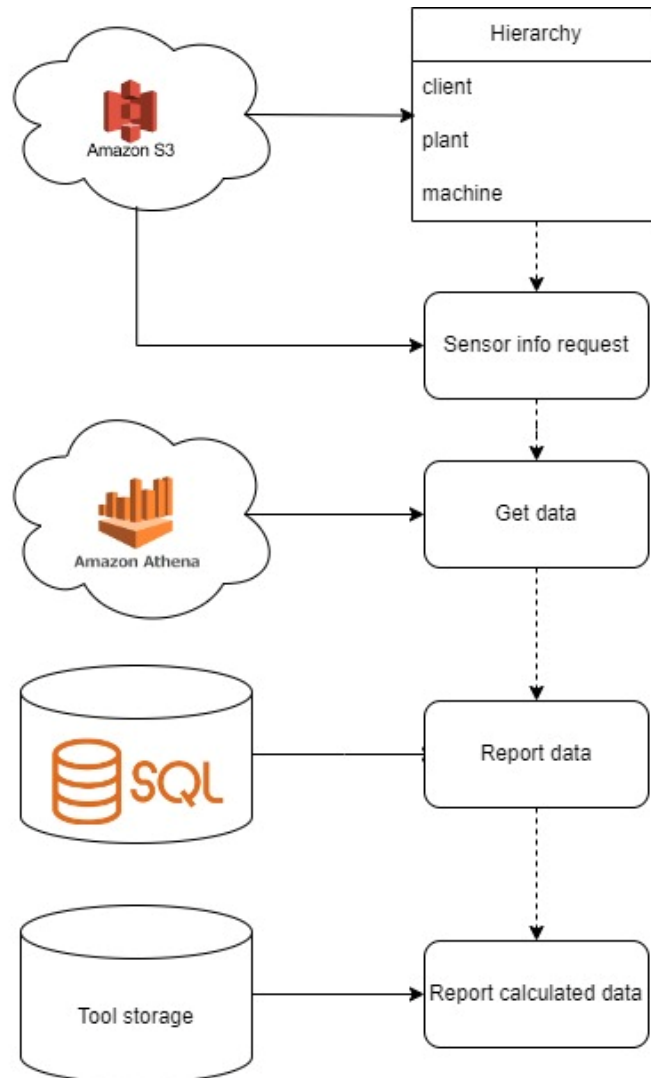


Рисунок 4.3 – Схема використання даних

На першому етапі створення звіту користувач обирає набір даних, для цього необхідно отримати інформацію про наявні у сховищі набори даних. Ця інформація отримується з хмарного об'єктного сховища Amazon S3 Bucket, яке зберігає дані великих обсягів у вигляді хронологічної послідовності показників сенсорів – часових рядів.

На наступному кроці система отримує метадані про обрані сенсори. Ця інформація зберігається в форматі JSON на Amazon S3. Метадані – це інформація про дані, яка включає в себе: першу та останню дати часового ряду, список пропусків даних (за замовчуванням пропуском вважається проміжок в 6 та більше годин і це реєструється на етапі отримання даних від клієнта), мінімальний, максимальний та медіану часових проміжків в часовому ряді (теж реєструється на етапі отримання даних). Обрані користувачем дані завантажуються і зберігаються в СУБД PostgreSQL та на сервері вебзастосунку у виділеному просторі.

Також, на етапі конфігурації звіту користувач може завдати правила для певних сенсорів. Наприклад, значення «`sensor_1 >= 5`» або «`sensor_1 >= 5 & sensor_2 == 0`». Файл конфігурації включає: ієрархічну структуру набору даних; дати початку та кінця часового ряду; мінімальну тривалість проміжку, який вважається пропуском даних; мінімальну тривалість проміжку, який вважається завмерлими даними; правила. Цей файл зберігається в форматі JSON на сервері аналітичної системи в піддиректорії звіту.

Після проведення аналізу даних всі розраховані показники, необхідні для візуалізацій та таблиць, зберігаються в форматах `.pkl` та `.parq` на сервері аналітичної системи в піддиректорії звіту.

Для роботи користувача зі звітом, система потребує тільки локально збережених даних, які використовуються у максимально швидких для Python форматах. За рахунок цього досягнуто високого рівня зручності для користувача, мінімізовано навантаження на головні сервери і витрати на звернення до AWS.

4.3 Клас `RD_analysis`: логіка роботи та основні етапи процесу

Клас `RD_analysis` (або `Raw Data Analysis`) відповідає за обробку даних із сенсорів, підготовку статистичних показників, збереження результатів і організацію простору для роботи з аналітичними даними. Загальна логіка роботи класу є наступною:

- *організація простору для даних*: створення необхідних директорій та файлів для зберігання мета-інформації;
- *підготовка списків сенсорів для завантаження та обробки*: організація структур для кожного сенсора залежно від наявних даних, визначення необхідності завантаження додаткових даних або оновлення інформації;
- *автоматизація обробки даних*: завантаження даних для кожного сенсора, обчислення статистичних показників, виявлення пропусків і завмерлих даних, збереження оброблених даних і мета-інформації.

Основні методи класу дозволяють здійснити завантаження, обробку, підготовку мета-даних та їх збереження у відповідній структурі (рис 4.4).

```

self.check_dirs()

self.sensors_range = self.arrange_sensors_lists()

if self.auto_flow:
    self.parallelized_download()
    self.parallelized_interval_count()
    self.parallelized_gaps()
    self.parallelized_freezes()
    self.parallelized_pd_profiling()
    self.get_correlation_groups()

```

Рисунок 4.4 – Методи класу RD_analysis

Змінна `auto_flow`, встановлена на значення `True` за замовчуванням, дозволяє автоматично запустити всі основні методи класу. Нижче наведено детальний опис основних методів і логіку виконання коду.

1. Перший етап – створення/перевірка простору для даних: метод `check_dirs()`. На рисунку 4.5 наведено фрагмент коду функції `check_dirs()`, а саме – організації робочого простору і оновлення історії завантажень на рівні машини.

Метод `check_dirs()` забезпечує створення та перевірку структури директорій для зберігання даних на сервері. У межах цього методу виконується:

- створення піддиректорій для даних: для кожного проєкту та набору сенсорів створюється унікальна структура директорій, що зберігає результати обчислень та статистичних показників. Така організація дозволяє легше керувати окремими звітами та обробляти їх незалежно один від одного;
- створення файлів для відстеження історії завантажень: це необхідно для розуміння, які дані вже завантажені, а які потрібно додатково отримати, що оптимізує використання ресурсів сервера;
- створення файлів для відстеження пропусків та завмерлих даних: ці файли допомагають зафіксувати періоди, коли дані відсутні або залишаються незмінними, що дозволяє провести більш точний аналіз якості даних.

```

19     class RD_analysis:
117         def check_dirs(self): 1usage
133         for m in self.config['machines']:
134             machine_dir = os.path.join(self.path, m)
135             if machine_dir not in machine_dirs:
136                 os.mkdir(machine_dir)
137             if os.path.join(machine_dir, 'raw_data') not in machine_dirs:
138                 os.mkdir(os.path.join(machine_dir, 'raw_data'))
139             if os.path.join(machine_dir, 'resampled_data') not in machine_dirs:
140                 os.mkdir(os.path.join(machine_dir, 'resampled_data'))
141             if not os.path.exists(os.path.join(self.path, m, f"{m}_info.csv")):
142                 df = pd.DataFrame(
143                     # sensor name; date of last download;
144                     # flag: all sensor data is downloaded; db first and last timestamps
145                     columns=['sensor', 'date downloaded', 'all data', 'db start', 'db end',
146                             # config dates; first and last timestamps of downloaded data
147                             'config start', 'config end', 'sensor start', 'sensor end',
148                             # number of raw records, median interval, number of resampled records;
149                             # flag to update data
150                             'records raw', 'interval', 'records resampled', 'refresh'])
151                 df.to_csv(os.path.join(self.path, m, f"{m}_info.csv"), index=False)
152             if not os.path.exists(os.path.join(self.path, m, f"{m}_downloads.csv")):
153                 df = pd.DataFrame(
154                     columns=['sensor', 'db start', 'db end', 'config start', 'config end',
155                             'start flag', 'end flag', 'all data'])
156                 df.to_csv(os.path.join(self.path, m, f"{m}_downloads.csv"), index=False)
157             if not os.path.exists(os.path.join(self.path, m,
158                 f"{m}_{self.dates_str}_gaps_freezes.csv")):
159                 df_g_f = pd.DataFrame(columns=['sensor', 'freezes', 'gaps'])
160                 df_g_f.to_csv(os.path.join(self.path, m,

```

Рисунок 4.5 – Фрагмент коду методу check_dirs()

Таким чином, метод `check_dirs()` формує простір для подальшого завантаження та обробки даних, що є важливим кроком перед основними обчисленнями.

2. Другий етап – підготовка списків сенсорів для завантаження та обробки: метод `arrange_sensors_lists()`. Після організації простору для даних, метод `arrange_sensors_lists()` відповідає за структурування та підготовку списків сенсорів, з якими буде проводитися обробка.

Цей метод є ключовим етапом у підготовці до обробки даних у класі `RD_analysis`. Він виконує кілька важливих функцій, серед яких зчитування мета-інформації, аналіз необхідності завантаження даних, оновлення мета-даних і підготовка списків для обробки пропусків та завмерлих даних (фрізів). Логіка роботи методу складається з перерахованих нижче основних кроків:

- *зчитування мета-інформації набору даних*: метод зчитує мета-дані для всіх сенсорів у конфігурації, включаючи інформацію про наявність даних, періоди активності сенсорів, а також історію попередніх завантажень. Ця інформація є основою для подальшого визначення необхідності завантаження додаткових даних;

- *визначення необхідності завантаження додаткових даних*: на основі мета-даних і історії завантажень для кожного сенсора визначається, чи потрібно завантажувати додаткові дані для заданого проміжку часу, або ж дані вже наявні. Якщо дані вже були завантажені раніше, завантаження можна пропустити, але ці дані можуть потребувати оновлення в мета-інформації для подальших розрахунків;

- *оновлення мета-даних*: для кожного сенсора, залежно від наявності даних, оновлюються проміжки часу, що вже були оброблені. Метод додає нові інтервали для завантаження та зберігає інформацію про інтервали у мета-даних, це дозволяє створити повну картину про наявність і доступність даних для аналізу;

- *підготовка списків для обробки пропусків і завмерлих даних (фрізів)*: окрім підготовки списків на завантаження, метод також формує окремі списки для обробки пропусків і завмерлих даних. Навіть якщо для певного сенсора дані вже були завантажені, вони можуть потребувати обчислення пропусків і фрізів, тому

списки для завантаження та для обробки можуть не співпадати. Це розділення дозволяє оптимізувати роботу, виконуючи обробку вже наявних даних без повторного завантаження.

На рисунку 4.6 наведено фрагмент коду методу `arrange_sensors_lists()`, а саме – оновлення значень атрибутів об’єкту класу, які зберігають списки, і збереження необхідних службових файлів.

```

164 def arrange_sensors_lists(self): usage
188     # select only sensors with data in a config range
189     sensors_with_data = machine_df[
190         (machine_df['db start'] < self.range_end) &
191         (self.range_start < machine_df['db end'])].index.to_list()
192     sensors = [t for t in sensors_with_data if t in self.config['sensors_list'][m]]
193     # crop machine_df to only relevant sensors
194     machine_df = machine_df.loc[sensors]
195
196     # download history
197     self.sensors_to_download[m], df_info_sensors = \
198         self.filter_dates_with_history(m, sensors, machine_df)
199     # arrange lists for processing: add sensors from config which are not in resulting files
200     self.arrange_lists_for_stat_files(m, sensors)
201
202     df_info = df_info.append(machine_df.loc[[t for t in df_info_sensors if t not in df_info.index]])
203     df_info.sort_index(inplace=True)
204     sensors_range[m] = df_info
205     df_info.to_csv(os.path.join(self.path, m, f"{m}_info.csv"))
206     return sensors_range

```

Рисунок 4.6 – Фрагмент коду методу `arrange_sensors_lists()`

Таким чином, метод `arrange_sensors_lists()` виконує комплексну підготовчу роботу, формуючи всі необхідні списки для подальших обчислень і зберігання результатів. Такий підхід забезпечує ефективне використання ресурсів, оптимізуючи завантаження та обробку великих обсягів даних для кожного сенсора.

3. Третій етап – завантаження даних, обрахунок та збереження результатів. Основна обробка даних у класі `RD_analysis` виконується методами, які реалізують паралельні процеси для завантаження і підготовки даних кожного сенсора. Це дозволяє значно оптимізувати роботу, зменшуючи час обробки. Основні методи обробки описано у таблиці 4.1.

Таблиця 4.1 – Методи обробки даних та збереження результатів

Метод	Опис
<code>parallelized_download()</code>	<i>Ініціює паралельне завантаження даних для кожного сенсора, зчитує інформацію про кожен сенсор, визначає періоди, для яких потрібно завантажити дані, і зберігає їх у відповідних директоріях</i>
<code>parallelized_gaps()</code>	<i>Виявляє пропуски у даних, які перевищують певний часовий інтервал. Ці пропуски можуть вказувати на відсутність вимірювань або проблеми з передачею даних</i>
<code>parallelized_freezes()</code>	<i>Виявляє завмерлі дані, коли значення сенсора залишаються незмінними протягом тривалого часу, що може свідчити про технічні проблеми або аномалії в роботі сенсора</i>
<code>parallelized_pd_profiling()</code>	<i>Створює базові статистичні звіти для кожного сенсора за допомогою <code>pandas_profiling</code>, які включають інформацію про середнє значення, медіану, розподіл даних тощо, що допомагає оцінити якість і властивості даних</i>
<code>parallelized_correlations()</code>	<i>Обчислює кореляційні залежності між різними сенсорами, що дозволяє створити групи висококорельованих сенсорів і допомагає виявити взаємозалежностей між різними параметрами і оптимізувати модель</i>
<code>parallelized_resampling()</code>	<i>Для великих часових проміжків автоматично виконує ресемплінг (усереднення) даних, що дозволяє зменшити їх розмір і спростити обчислення</i>

Розглянемо приклад використання паралельних процесів на прикладі завантаження даних. Метод `parallelized_download()` викликає для кожного сенсора в списку сенсорів для завантаження метод `get_sensor_data_by_query()`. На рисунку

4.7 наведено фрагмент коду з результатом отроботки `get_sensor_data_by_query()` у випадку наявності чи відсутності даних.

```

371 def get_sensor_data_by_query(self, machine: str, sensor: dict): 1 usage
396     if updated:
397         return (sensor['sensor'], datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
398               df_parts.index[0], df_parts.index[-1],
399               len(df_parts.loc[self.range_start:self.range_end, :]), history_dict)
400     else:
401         self.empty_sensors[machine].append(sensor['sensor'])
402         return (sensor['sensor'], datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
403               None, None,
404               len(df_parts.loc[self.range_start:self.range_end, :]), history_dict)
405
406 def parallelized_download(self): 1 usage
407     for m in self.config['machines']:
408         if self.sensors_to_download[m]:
409             history = pd.read_csv(os.path.join(self.path, m, f"{m}_downloads.csv"))
410             print('start pool')
411             pool = multiprocessing.Pool(self.cores_num)
412             result = pool.map(partial(self.get_sensor_data_by_query, *args: m), self.sensors_to_download[m])
413             pool.close()
414             pool.join()
415             for r in result:
416                 if r[4]:
417                     history = pd.concat(objs=[history, pd.DataFrame.from_dict(r[5])], ignore_index=True)
418                     history.loc[history['sensor'] == r[0], 'db start'] = \
419                         self.sensors_range[m].loc[r[0], 'db start']
420                     history.loc[history['sensor'] == r[0], 'db end'] = \
421                         self.sensors_range[m].loc[r[0], 'db end']
422                     self.sensors_range[m].loc[r[0], 'date downloaded'] = r[1]
423                     self.sensors_range[m].loc[r[0], 'sensor start'] = r[2]

```

Рисунок 4.7 – Фрагмент коду методу `get_sensor_data_by_query()`

Метод повертає назву сенсора, дату закінчення завантаження, граничні значення часових проміжків завантажених даних, кількість наявних значень у часовому ряді, словник з оновленою історією завантажень. Коли всі процеси завантаження відпрацювали, метод `parallelized_download()` розкладає отримані кортежі результатів по відповідним атрибутам об'єкту класу та директоріям проекту.

Описана вище реалізація має суттєві переваги. Завдяки паралельному процесингу, всі методи класу `RD_analysis` працюють у багатозадачному режимі, що значно підвищує продуктивність обробки. Клас є громіздким, проте це дозволяє виконувати всі етапи аналізу в одному потоці, не перезавантажуючи сервер. Це

особливо корисно у випадках, коли необхідно регулярно оновлювати дані або створювати звіти для різних проміжків часу. Кожен звіт розглядається як окремий проєкт зі своєю директорією, що містить результати обчислень, необхідні для створення таблиць і візуалізацій. Такий підхід дозволяє гнучко керувати обчисленнями та адаптуватися до змін у конфігурації (зміна дат або сенсорів тощо).

Таким чином, методи класу `RD_analysis` забезпечують повний цикл обробки даних, починаючи від організації простору для даних і закінчуючи створенням статистичних звітів та збереженням результатів обчислень.

4.4 Скрипт для зв'язку між **back-end** і **front-end**

Скрипт `create_report.py()` забезпечує інтеграцію між обробкою даних на бекенді та аналітичним інтерфейсом на фронтенді, використовуючи об'єкт `project_data`, який є екземпляром класу `RD_analysis`. Ця зв'язка дозволяє підготувати та передати аналітичні результати, які були оброблені і збережені під час виконання `RD_analysis`, у зручному для візуалізації форматі.

Опишемо логіку роботи зв'язки.

1. *Ініціалізація `project_data`*: скрипт ініціалізує об'єкт `project_data` на основі класу `RD_analysis`, використовуючи конфігураційний файл проєкту. Цей файл містить інформацію про налаштування для аналізу, як-от список сенсорів, часові інтервали та інші параметри.

2. *Зчитування збережених даних*: функція `calculations_to_project()` у скрипті зчитує результати, які були попередньо оброблені і збережені за допомогою `RD_analysis`. Ці дані включають:

- таблиці основних статистичних показників для сенсорів;
- інформацію про пропуски та завмерлі дані (фрізи);
- кореляційні матриці та статистику, потрібні для аналітичних звітів.

У наведеному фрагменті коду функції `calculations_to_project()` можна побачити, як система крок за кроком перевіряє наявність обрахованих значень за

кожним розділом звіту і додає їх до змінної `summary` (рис. 4.8). Після цього обраховуються значення за якими будуть візуально позначатися найбільш проблематичні сенсори в звіті.

```

49 def calculations_to_project(project_data, config, pr_path): 1 usage
64
65 # basic statistics
66 b_s_file = path.join(project_data.path, f"{m}__{project_data.dates_str}__rsm__basic_stats.pkl"
67     if project_data.resampled \
68     else path.join(project_data.path, f"{m}__{project_data.dates_str}__raw__basic_stats.pkl")
69 > if path.exists(b_s_file):...
85
86 # correlation
87 > if path.exists(path.join(project_data.path, f"{m}__{project_data.dates_str}__correlations.parq
127
128 # intervals, freezes, gaps and summaries
129 suffixes = ['intervals.parq', 'freezes.parq', 'gaps.parq']
130 for s in suffixes:
131 >     if path.exists(path.join(project_data.path, f"{m}__{project_data.dates_str}__{s}")):...
147
148 # predefined values
149 if m in list(config["values_data"]):
150     predefined = config["values_data"][m]
151     summary = add_predefined_summary(summary, predefined, m, project_data.path)
152     print(f"add_predefined_summary | {m}")
153
154 tt = pd.DataFrame(summary.loc[summary['to_summary'] > 0, 'to_summary'].value_counts())
155 summary_th = tt['to_summary'].loc[tt['to_summary'] <= tt['to_summary'].median()].index[0]
156 if len(summary.loc[summary['to_summary'] > summary_th]) > 0:
157     summary = summary.loc[summary['to_summary'] > summary_th]
158 else:
159     summary = summary.loc[summary['to_summary'] >= summary_th]
160 summary.reset_index(inplace=True)
161 summary.rename(columns={"index": "Sensor"}, inplace=True)
162 summary.sort_values(by='to_summary', ascending=False, inplace=True)
163 summary['Predefined conditions'] = summary['Predefined conditions'].astype(str)

```

Рисунок 4.8 – Фрагмент коду методу `calculations_to_project()`

3. *Форматування даних для фронтенду*: дані, зчитані `calculations_to_project()`, зберігаються в структурованих форматах (як Parquet та Pickle). Це дозволяє ефективно передавати їх для візуалізації у веб-інтерфейсі. Всі потрібні таблиці та аналітичні метрики форматуються так, щоб їх можна було легко інтегрувати у компоненти Dash для фронтенду.

4. *Відображення у веб-інтерфейсі*: після зчитування та форматування дані використовуються для побудови інтерактивних таблиць і графіків у веб-інтерфейсі.

Це дозволяє користувачам переглядати статистичні звіти та аналітичні висновки, що були оброблені `RD_analysis`, у зручному та наочному вигляді.

Переваги такої зв'язки є очевидними. Скрипт `create_report.py()` створює ефективний зв'язок між бекендом і фронтендом, автоматично зчитуючи готові аналітичні дані, підготовлені `RD_analysis`, і забезпечуючи їхнє відображення в інтерфейсі. Це значно підвищує швидкість завантаження звітів, адже обчислень не проводиться – вже оброблені дані швидко подаються у зручному форматі. Такий підхід покращує юзабіліті системи, забезпечуючи користувачів інтерактивними звітами з мінімальними затримками, що робить роботу з аналітичними даними швидкою і простою.

4.5 Структура інтерфейсу, налаштування системи для проведення аналізу

Для створення інтерактивних елементів інтерфейсу вебзастосунку системи аналізу показників сенсорів та організації їх структури було використано Dash, Dash Bootstrap Components, Dash Mantine Components і Plotly.

Структуру інтерфейсу, розміщення компонентів і фільтрів описують файли з суфіксом *layout*. В інтерфейсі вебзастосунку представлені такі основні елементи:

- *фільтри та параметри вибору*: компоненти для вибору машин, сенсорів, діапазону дат і розміру пропусків – це дозволяє користувачам налаштувати параметри аналізу, щоб отримати потрібну вибірку даних;

- *таблиці з аналітичними результатами*: використовується `dash_table.DataTable` для створення інтерактивних таблиць із результатами аналізу, такими як зведення за сенсорами, пропуски, завмерлі дані та інші показники;

- *інформаційні картки*: картки Bootstrap з інструментами керування Mantine відображають інформацію про кластер, клієнта, сайт, а також часовий діапазон, що використовується в аналізі – це забезпечує користувачам зручний доступ до загальної інформації про параметри дослідження.

У файлах з суфіксом *callbacks* реалізовані функції виклику коду у певні моменти часу для забезпечення інтерактивності та обробки дій користувача:

– *оновлення фільтрів*: колбеки автоматично змінюють значення у випадючих списках, залежно від вибору користувача (наприклад, зміна списку сенсорів після вибору машини);

– *завантаження й відображення результатів обробки*: колбеки зчитують попередньо збережені на сервері результати аналітики (наприклад, таблиці з пропусками, завмерлими даними або кореляціями), які потім відображаються у вигляді графіків і таблиць;

– *повідомлення та валідація*: вбудовані повідомлення й алерти надають користувачам підказки для налаштування параметрів і попереджають про можливі помилки або особливості аналізу.

Інтерактивні елементи інтерфейсу реалізовано з використанням наступних компонент:

– *DateRangePicker* і *випадючі списки Dropdown* використовуються для вибору дат, машин, сенсорів та інших параметрів, що дозволяє налаштувати аналіз, виходячи з конкретних потреб і періоду;

– *алерти (англ. alert) та сповіщення*: сповіщення про певну подію, наприклад, про тривалий часовий діапазон або виявлення пропусків, допомагають користувачам ефективно працювати з інтерфейсом, попереджаючи про особливості аналітики і допомагаючи правильно налаштувати параметри.

Робота з аналітичною системою починається з вибору сценарію: користувач може обрати існуючий звіт (проект) або створити новий. Сторінка інструменту завантажується з двома вкладками: «Select project» («Обрати проект») та «Create project» («Створити проект») (рис. 4.9). Якщо хоча б один звіт існує, активною буде вкладка вибору. На ній розташовано таблицю з переліком існуючих звітів (рис. 4.10). Таблицю можна сортувати, фільтрувати, видаляти рядки, видаляючи і звіти (дані при цьому зберігаються, створення нового звіту на цих даних

відбудеться дуже швидко). Відкрити звіт можна, клікнувши на іконку в останньому стовбці таблиці.



Рисунок 4.9 – Вкладки для вибору сценарію роботи

Description	Server	Client	Plant	Machines and sensors	Start	End	Created	Open
filter data...								
x				: 220 / 220	2022-05-18 23:30:33	2022-09-28 11:56:01	28 12:39:10	
x				: 220 / 220	2022-05-18 23:30:33	2022-09-28 09:56:33	28 12:31:38	
x				: 220 / 220	2020-01-01 00:00:00	2022-07-07 00:00:00	22 13:32:34	
x				: 87 / 87	2021-09-22 21:02:44	2021-10-14 01:44:00	22 09:25:51	

Рисунок 4.10 – Вибір існуючого звіту

Якщо потрібного звіту в переліку немає, або користувач не хоче його шукати, він переходить на вкладку створення проекту. Новий проєкт створюється за 4 кроки: 1) вибір набору даних до рівня заводу (рис. 4.11), 2) вибір машин та сенсорів (рис. 4.12), 3) вибір часового проміжку (рис. 4.13), 4) визначення додаткових правил (рис. 4.14)

Step 1. Select data, submit and review the number of machines, sensors, and dates range.

Server: Select...

Client: Select...

Plant: Select...

Refresh plant info (for streaming or changed raw data).

Submit plant

Рисунок 4.11 – Вибір набору даних

Step 2. Select machines, sensors and minimum gap size. Submit and review gaps distribution over data.

Machines

Sensors to include

Minimum gap

* Remove or add machines and sensors, change gap size, resubmit to refresh plots and dates suggestions.

Рисунок 4.12 – Вибір машин та сенсорів

Step 3. Select dates range from suggested or any dates from calendar below.

Suggested dates are based on data range of all non empty and not excluded sensors

Suggested start dates

Suggested end dates

Selected sensors dates

Рисунок 4.13 – Вибір часового проміжку

Step 4. Create additional rules.

You can review statistics on normal and abnormal values of the selected sensors.

Create rules for sensor's behavior like: 'machine | sensor name' + 'sign' + 'value'.

Predefined value range	Sensor name	Sign	Value
	<input type="text" value="Select..."/>	<input type="text" value="Sele..."/>	<input type="text"/>

Optional. Provide short description for current RDA.

Description

Рисунок 4.14 – Визначення додаткових правил

Користувач проходить перші кроки – вибір набору даних і встановлення розміру часового проміжку, який вважатиметься пропуском, до бази даних аналітичної системи відправляється запит на існуючі звіти для цього набору даних.

На рисунку 4.15 курсор вказує на перемикач «Refresh plant info» («Оновити інформацію по підприємству»). Якщо дані знайдено в базі даних проекту (навіть за відсутності звіту), цей перемикач стане активним. У випадку, якщо дані транслюються в режимі реального часу, або були перезаписані, це дозволить працювати з актуальними даними.

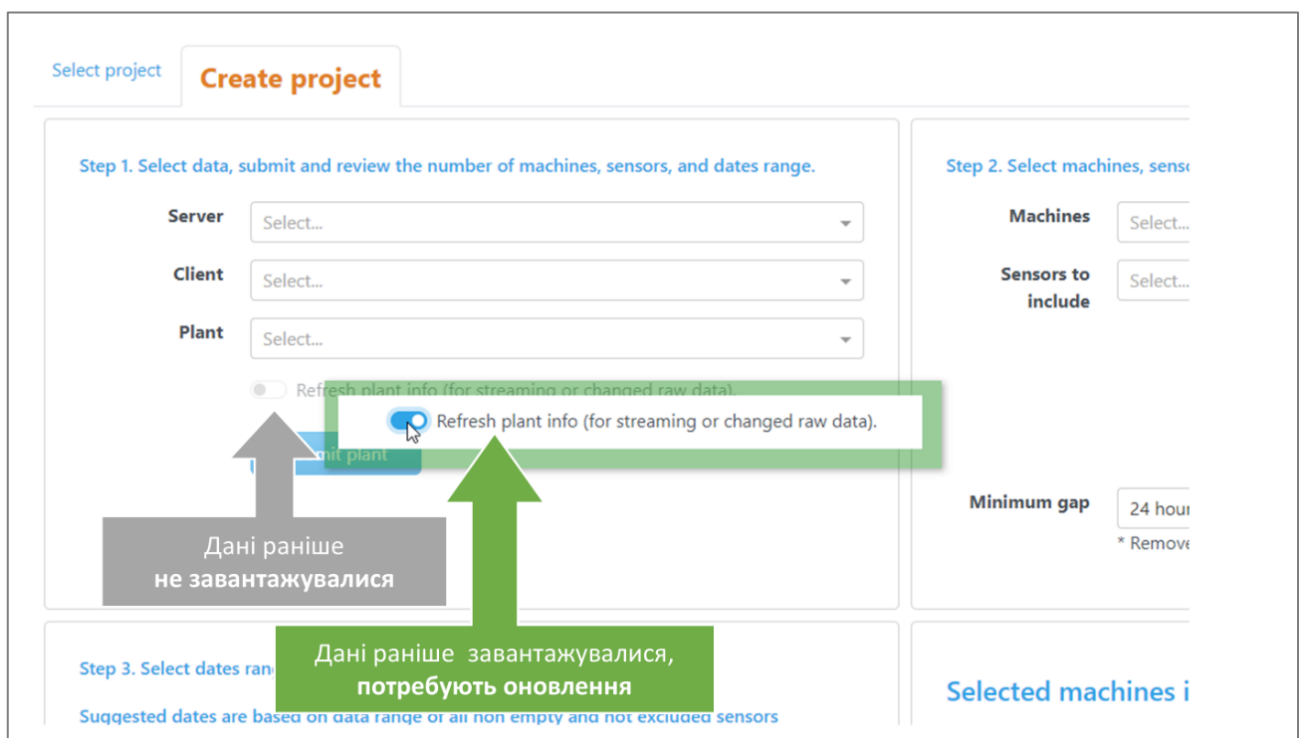


Рисунок 4.15 – Перемикач оновлення даних

Якщо користувач завдав параметри набору даних, звіти за яким вже існують, система запропонує йому з ними ознайомитися (рис. 4.16).

Одним із пунктів технічного завдання було оптимізувати кількість запитів до баз даних. Ця перевірка спрямована саме на досягнення мети оптимізації кількості запитів і навантаження на сервери. Крім того, переглянувши існуючі звіти,

користувач може точніше обрати часовий проміжок для свого звіту, якщо шукає відповідь на якісь конкретні питання.

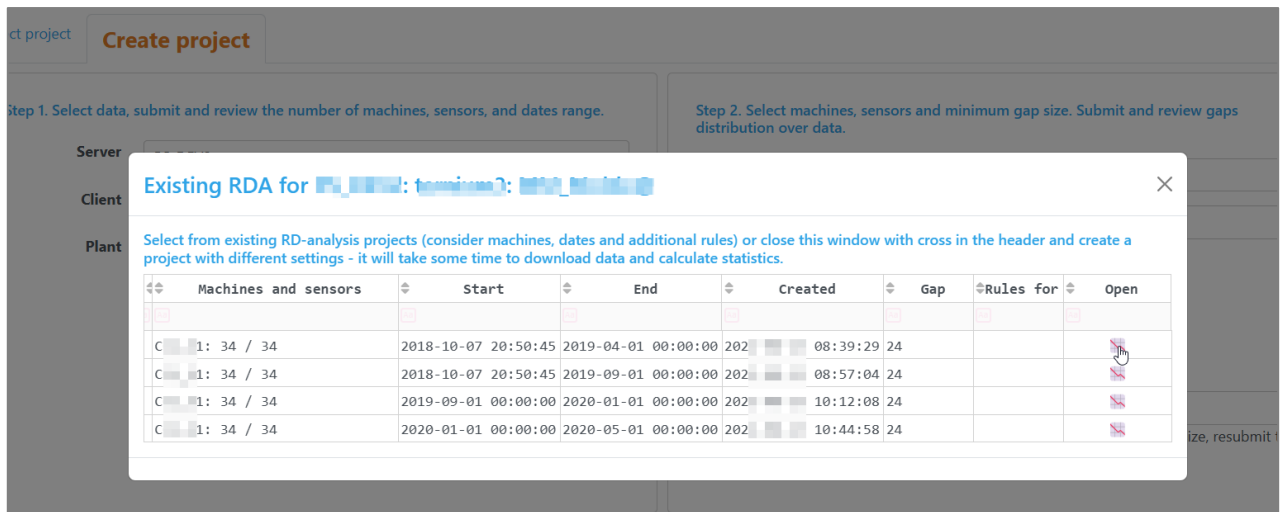


Рисунок 4.16 – Перелік існуючих звітів для обраного набору даних

Так, наприклад, можна зробити, щоб порівняти характеристики даних тренувального періоду, періоду валідації та тестового періоду. І потім або змінити межі, або розуміти заздалегідь, якої якості модель можна очікувати. Цією опцією активно користуються працівники департаменту наукових досліджень, підбираючи саме такі якості наборів даних, які їм потрібні для перевірки тих чи інших гіпотез і можливостей алгоритмів.

Таким чином, технічна реалізація перших кроків налаштування звіту не тільки надає можливість вибору набору даних, а і слугує меті оптимізації витрат часу персоналу та оптимізації навантажень на сервери і полегшує роботу дослідників та науковців.

Якщо в переліку немає потрібного звіту або система не запропонувала вибір, тобто таких звітів немає, користувач переходить до другого кроку налаштувань – вибору машин та сенсорів для кожної.

Одночасно для аналізу в списку «Machines» («Машини») можна обрати будь-яку кількість машин, але аналітика буде показуватися для кожної окремої машини. На наступних кроках прогностичного обслуговування – обробка даних, тренування

2024 р.

моделі, прогнозування, інтерпретація прогнозу – робота йде над кожною машиною окремо, крос-сенсорні частини алгоритму теж працюють в межах однієї машини, тому і аналітична система дотримується тієї ж логіки.

Більше того, один великий вузол може бути розділений на кілька «машин» на етапі формування набору даних в базі даних. Так, насоси, ротор та передавальний вал фактично є єдиним механізмом, але може бути представлений мінімум трьома різними гілками на рівні ієрархії зберігання даних в базі даних. Насос на початку циклу роботи видає пікові вібрації, показники сенсорів дуже відрізняються від «нормальної поведінки», проте це не є аномалією – це правильна та очікувана поведінка машини, тому на подальших кроках робота з даними на початку робочих циклів буде налаштована окремо для кожного насосу. Решта даних обробляється без цих «сліпих зон».

Користувач обирає машину, і в наступний випадаючий список «Sensors to include» («Додати сенсори») за замовчуванням додаються всі сенсори цієї машини (рис. 4.17).

The screenshot displays a web application interface for configuring a project. It is divided into three main steps:

- Step 1: Select data, submit and review the number of machines, sensors, and dates range.** This section includes dropdown menus for 'Server', 'Client', and 'Plant', along with a 'Submit plant' button.
- Step 2: Select machines, sensors and minimum gap size. Submit and review gaps distribution over data.** This section features a 'Machines' dropdown, a 'Sensors to include' list, and a 'Minimum gap' input field set to '24 hours'.
- Step 3: Select dates range from suggested or any dates from calendar below.** This section includes 'Suggested start dates' and 'Suggested end dates' with radio button options, a date range input field, and a 'Submit sensors' button.

Below the date selection, there is a calendar for June 2018. To the right, a table titled 'Selected machines info' provides details for three machines:

Machine name	Empty sensors number	Non-empty sensors number	Start date	End date	Latest sensor start	Earliest sensor end
	0	4	2016-12-27 15:45:13	2019-11-06 00:00:00	2016-12-27 15:54:51	2019-09-25 00:12:17
	0	8	2017-01-01 03:00:00	2019-12-05 20:45:00	2018-06-06 14:30:00	2019-12-05 20:45:00
PS1P	0	4	2016-12-27 15:41:39	2019-11-06 00:00:00	2016-12-27 15:42:36	2019-09-25 00:17:14

At the bottom right, there is a section for 'Selected sensors timeline with gaps'.

Рисунок 4.17 – Списки машин, сенсорів та вибір дат для аналізу

Для всіх обраних сенсорів ще на етапі конфігурації звіту надаються візуалізації пропусків даних, таблиці з датами наявних даних для кожного сенсору. Користувач може видаляти та додавати знов сенсори, налаштовуючи свій проект, зважаючи на цю інформацію та на інструкції та пояснення від клієнта. Також користувач обирає межі часового проміжку, для якого потрібно провести аналіз (третій крок конфігурації проєкту). Система автоматично пропонує проміжок від найпізнішого початку до найраннішого кінця даних сенсора, користувач може обрати проміжок з дат, запропонованих системою: найменше перше значення початок, медіана перших значень, найбільше перше значення часових рядів даних всіх сенсорів; найменше, медіана і найбільше останнє значення часових рядів даних всіх сенсорів; або обрати довільні дати та час, використовуючи вбудований календар, або ввівши значення в поле.

За замовчуванням, мінімальний проміжок, який вважається пропущеними даними дорівнює шести годинам. Користувач може обрати інші налаштування проміжку згідно специфіки даних та задачі (рис. 4.18).

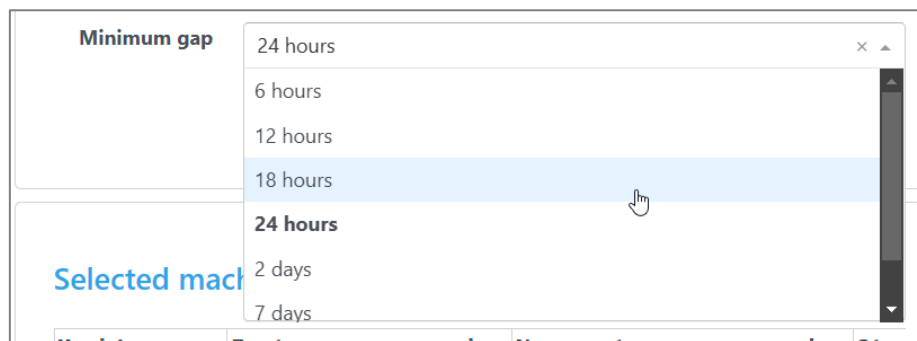


Рисунок 4.18 – Випадаючий список для налаштування мінімального значення проміжку для визначення пропущених даних

На базі ієрархії набору даних, часових рамок та значення для визначення пропусків створюється інформаційний запит до хмарного середовища, де зберігаються всі наявні дані для цього клієнта та заводу. Цей запит значно легший, ніж завантаження самих даних, проте дає достатньо інформації, щоб сформувати

набір даних. Відповідь на цей запит зберігається на внутрішньому сервері системи і використовується для подальших операцій, уникаючи повторних запитів, що, як вже було згадано, було однією з задач при створенні аналітичного інструменту. Запит буде повторено, якщо користувач увімкне перемикач «Refresh plant info» (рис. 4.15)

Для обраних сенсорів для кожної машини надається така інформація (рис. 4.19):

- назва машини (як в базі даних);
- кількість пустих сенсорів;
- кількість сенсорів з даними;
- часовий проміжок, коли для цієї машини є дані мінімум для одного сенсора;
- найпізніша дата початку наявних даних на рівні сенсорів в межах вказаного проміжку;
- найранніша дата закінчення наявних даних на рівні сенсорів в межах вказаного проміжку;
- візуалізація пропусків даних;
- у випадку наявності пропусків у більшості сенсорів, буде надано попередження.

На графіку пропусків окремо кольорами позначено «Late start» («Пізній початок») та «Early end» («Раннє закінчення»).

Формально відсутність даних від початку визначеного проміжку до першого наявного значення часового ряду та від останнього наявного значення до кінця визначеного проміжку є пропусками даних, проте це показує, що в такому вигляді при тренуванні моделі дані цих сенсорів відсутні під час тренувального або тестового періодів і це може вплинути на якість роботи моделі (рис. 4.20).

Кафедра інтелектуальних інформаційних систем
Інформаційна система аналізу показників сенсорів технологічних ліній

Step 3. Select dates range from suggested or any dates from calendar below.

Suggested dates are based on data range of all non empty and not excluded sensors

Suggested start dates

2016-12-27 15:41:39 (earliest start)

2016-12-29 21:27:25.500000 (median start)

2018-06-06 14:30:00 (latest start)

Suggested end dates

2019-02-09 10:40:06 (earliest end)

2019-10-31 02:31:07 (median end)

2019-12-05 20:45:00 (latest end)

2018-06-06 14:30:00 – 2019-02-09 10:40:06

Selected sensors dates

Motor

Export

Sensor	Data			Gaps		
	Start	End	days	No	days	%
filter d						
x	2016-12-27 15:54	2019-09-25 08:12	1001.68	191	246.44	24.6
x	2016-12-27 15:45	2019-02-09 10:40	773.79	176	230.4	29.78
x	2016-12-27 15:45	2019-09-25 08:12	1001.69	184	238.03	23.76
x	2016-12-27 15:54	2019-09-25 08:12	1001.68	192	243.95	24.35

Process

Pump

а) обрана машина, дати сенсорів

Selected machines info

Machine name	Empty sensors number	Non-empty sensors number	Start date	End date	Latest sensor start	Earliest sensor end
Motor	0	4	2016-12-27 15:45:13	2019-11-06 00:00:00	2016-12-27 15:54:51	2019-09-25 08:12:17
Process	0	8	2017-01-01 03:00:00	2019-12-05 20:45:00	2018-06-06 14:30:00	2019-12-05 20:45:00
Pump	0	4	2016-12-27 15:41:39	2019-11-06 00:00:00	2016-12-27 15:42:36	2019-09-25 08:17:14

Selected se Data Alert!

Motor

There are gaps in more than 50% of the data in the Motor machine. There are gaps in more than 50% of the data in the Pump machine. Please select different data set or remove gappy tags.

Sensors

Motor

Process

Pump

Jul 2018 Aug 2018 Sep 2018 Oct 2018 Nov 2018 Dec 2018 Jan 2019 Feb 2019

Type

Gap

Late Start

б) попередження про відсутність даних в більшості сенсорів, візуалізація пропусків даних

Рисунок 4.19 – Інформаційні таблиці про набір даних

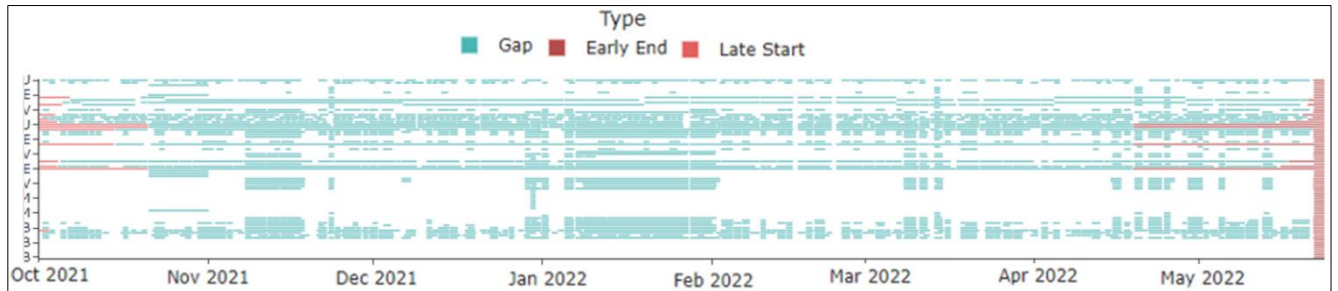


Рисунок 4.20 – Візуалізація пропущених даних з виділенням пізнього початку та раннього закінчення

Останній, четвертий, крок конфігурації параметрів для аналізу – формулювання правил (рис. 4.21).

Sensors values

	Sensor name	Sign	Value
×	[blurred]	>	5

Step 4. Create additional rules.

You can review statistics on normal and abnormal values of the selected sensors.

Create rules for sensor's behavior like: 'machine | sensor name' + 'sign' + 'value'.

Predefined value range

×

Sensor name

×

Sign

▾

Sele.↕

>

<

>=

<=

Value

Add rule

Optional. Provide short description for current RDA.

Description

description

Рисунок 4.21 – Формулювання правил

Цей крок не є обов'язковим. Зазвичай він використовується, якщо від клієнта є певна інформація про індикативні сенсори, критичні значення для певних сенсорів тощо. Також на цьому кроці можна додати опис аналізу. Це зручно, коли з одними й

тими даними працюють фахівці з різних відділів. Якщо аналітик найчастіше захоче побачити якнайдовший період з максимумом доступної інформації, досліджувач може формувати досить обмежений набір для вирішення якоїсь конкретної задачі. Наприклад, позбутися всіх бінарних сенсорів, взяти тільки «вимірювані дані процесів» (measuring process data) і розробляти свій спосіб визначення багаторежимного способу роботи, якщо про це не було повідомлено клієнтом.

Ці правила не вплинуть на статистичні розрахунки чи кореляційний аналіз. У звіті буде вказано, який відсоток значень сенсора відповідає правилу. Якщо клієнт повідомив, що машина працює в кількох режимах, при цьому для першого режиму індикативний сенсор має значення 0, для другого – 1, аналітик зможе побачити наскільки рівномірно розподілені режими в наборі. У випадку, коли один із режимів позначений лише у невеликому відсотку часового ряду, фахівець матиме інформацію, що на таких обраних межах часового проміжку неможливо якісно натренувати модель для кожного режиму. У випадку інформації про критичні значення, навпаки – переважний відсоток таких значень повідомить аналітику, що обрано період аномальної роботи машини, отже навчити модель визначати нормальний режим буде проблематично – більшість даних не може бути аномалією.

Коли користувача влаштовує набір даних, за потреби додано правила та опис проєкту, можна підтверджувати налаштування та починати аналіз. В залежності від розміру набору даних завантаження та аналіз даних може зайняти від кількох хвилин до години.

4.6 Аналіз та виявленням проблемних даних

Після здійснення налаштувань для аналізу по завершенню розрахунків у новому вікні відкривається аналітичний звіт. Якщо користувач закрав браузер під час розрахунків, дані будуть оброблені, звіт сформовано і додано до переліку в таблиці готових звітів. Усі таблиці та візуалізації звіту є інтерактивними: таблиці

можна сортувати і фільтрувати, візуалізації – масштабувати і по вертикалі, і по горизонталі, відповідно, роздивляйтесь результати докладніше. Усі таблиці звіту можна завантажити у форматі .csv, всі візуалізації – в форматі .png.

Сторінка звіту включає п'ять вкладок, які відповідають підрозділам аналітичного звіту (рис. 4.22):

- Підсумки (Summary)
- Основна статистика (Sensors Basic Stats)
- Аналіз завмерлих даних (Freeze Sensors Analysis)
- Кореляційний аналіз (Corelation Analysis)
- Аналіз пропусків та часових інтервалів (Gaps and Time Interval Analysis)

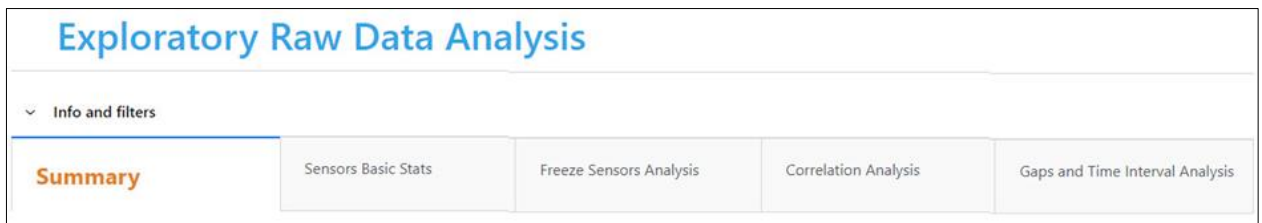


Рисунок 4.22 – Вкладки аналітичного звіту

Картка керування даними є загальною для всіх вкладок (рис. 4.23).

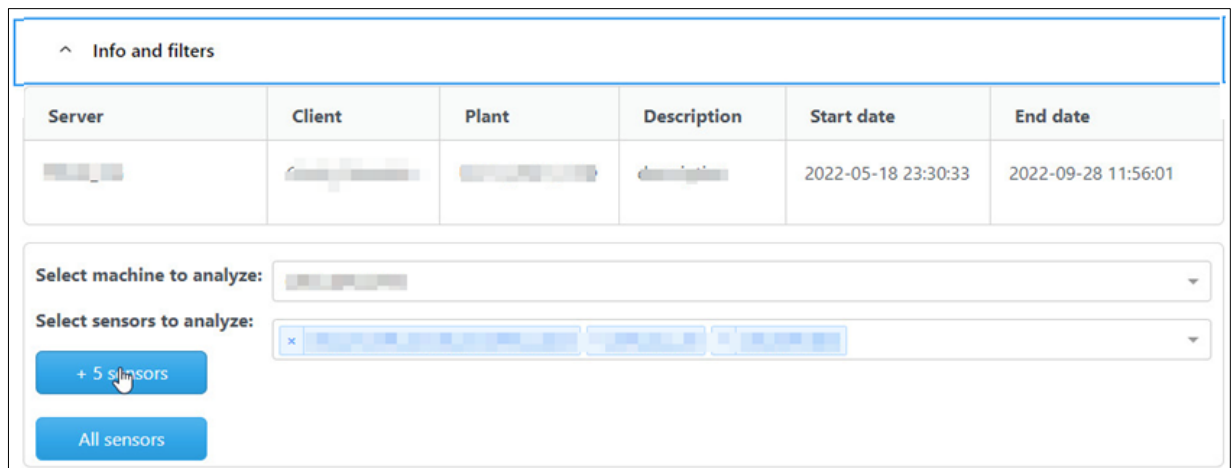


Рисунок 4.23 – Картка керування даними

За замовчуванням картка керування даними згорнута, обрано першу машину в переліку та сенсори, які було автоматично визначено як потенційно проблематичні. Змінюючи перелік сенсорів у цій картці, користувач налаштовує всі візуалізації та таблиці крім таблиці «Всі сенсори машини» («All machine sensors») на вкладці *Підсумки*.

Вкладка Підсумки надає інформацію у вигляді двох таблиць та текстового опису. Опишемо їх більш детально.

1. Таблиця «Автоматично відібрані сенсори» («Automatically selected Sensors») включає (рис. 4.23):

- тип та кількість окремих значень;
- відсоток нульових та пропущених значень;
- коефіцієнти варіації та ексцесу;
- кількість окремих значень тривалості часових проміжків;
- тривалість найдовшого періоду пропущених даних (в годинах);
- тривалість найдовшого періоду завмерлих даних (в годинах);
- кількість сенсорів із високим значенням коефіцієнту кореляції;
- відповідність правилам.

The table below includes sensors with some problematic statistical characteristics. Problematic values are highlighted, hover on column header for more explanations.											
Freeze Sensors Analysis						Correlation Analysis			Gaps and Time Interval Analysis		
Automatically selected sensors ⓘ											
Export											
⌵Sensor	⌵Distinct value No	⌵Zero values %	⌵Missing values %	⌵Coefficient of variation	⌵Kurtosis	⌵Type	⌵Time interval distinct values No	⌵Longest gap (hrs)	⌵Longest freeze (hrs)	⌵Highly correlated sensors No	⌵Predefined conditions
filter data...											
x [redacted]	5	92.33	0	3.47	8.28	Numeric	1	76.92	143.88	0	0
x [redacted]	723	92.16	0	3.5	10.37	Numeric	1	76.92	143.28	2	0
x [redacted]	1736	92.59	0	4	25.25	Numeric	1	76.92	143.88	0	0

Рисунок 4.23 – Таблиця «Автоматично відібрані сенсори»

При наведенні вказівника миші на іконку інформації біля назви таблиці (літера «і» в кружечку), з'являється текстовий поп-ап із інформацією про таблицю

і логіку позначення значень кольором. Кольором позначено проблематичні значення: замала кількість окремих значень, зависокі значення коефіцієнтів варіації та ексцесу або окремих значень тривалості інтервалів (нерегулярне надходження даних), пропусків або завмерлих даних.

Переглянувши цю таблицю, аналітик бачить ті сенсори, які одночасно мають велику кількість характеристик, що є незручними для тренування моделі. Наприклад, сенсор із довгими періодами пропусків або завмерлих даних і з низькою кількістю окремих значень – скоріш за все має недостатньо даних, щоб можна було вивчати його поведінку і визначати аномалії; сенсор з високими коефіцієнтами, що показують наявність викидів та з великим відсотком нульових значень – це сенсор з поведінкою, яку ми називаємо «спіку», в такому випадку буває корисним спитати в клієнта, чи не є це індикативним сенсором, який насправді позначає перемикання механізмів, а не передає чисельні характеристики процесу роботи механізмів.

2. Таблиця «*Всі сенсори машини*» («*All machine sensors*») надає загальне уявлення про характеристики даних кожного сенсора у вибірці (рис. 4.24):

- *дані*: тип, початок, кінець, тривалість (днів);
- *пропуски*: найдовший пропуск (в годинах), загальна тривалість (в годинах), відсоток від загальної тривалості даних;
- *завмерлі дані*: найдовший період завмерлих даних (в годинах), загальна тривалість (в годинах), відсоток від загальної тривалості даних;
- *значення*: відсоток окремих значень, відсоток негативних значень;
- *інтервали*: мінімальний, середнє арифметичне, медіана, стандартне відхилення, максимальний.
- *кількість сенсорів з високим коефіцієнтом кореляції*.

Серед числових даних окремо виділено типи констант та бінарних сенсорів. Формально вони теж є чисельними, але для зручності користування формулювання спрощено.

All machine Sensors									
Export									
		Data					Gaps		
	Sensor	Type	Start	End	Duration (days)	Longest (hrs)	Total (hrs)	% of Duration	
	filter data...								
x	○	Numeric	2020-12-21	2022-09-28	645	39.08	63.35	0.41	
x	○	Numeric	2022-05-04	2022-09-28	147	39.08	63.36	1.79	
x	○	Constant 0	2022-05-04	2022-09-06	125	1224.29	3176.42	105.25	
x	○	Numeric	2022-05-04	2022-09-28	147	39	63.27	1.79	
x	○	Constant 0	2022-05-04	2022-09-06	125	1224.29	3176.42	105.25	
x	○	Numeric	2022-05-04	2022-09-28	147	38.99	63.26	1.79	
x	○	Constant 0	2022-05-04	2022-09-06	125	1224.29	3176.42	105.25	
x	○	Binary	2020-12-21	2022-09-28	645	344.84	2982.91	19.25	
x	○	Binary	2022-05-04	2022-09-21	140	1208.34	3137.6	93	
x	○	Numeric	2020-01-01	2022-09-28	1001	39.08	63.33	0.26	
x	○	Numeric	2022-05-04	2022-09-28	147	38.99	89.23	2.52	
x	○	Constant 0	2022-05-04	2022-09-06	125	1224.29	3176.42	105.25	
x	○	Constant 0	2022-05-04	2022-09-06	125	1224.29	3176.42	105.25	

a) data, gaps

Freezes			Values %		Intervals					Correlation
Longest (hrs)	Total (hrs)	% of Duration	Distinct	Negative	Min	Mean	Median	StD	Max	Tags #
			0.02	0	0.5	0.5	0.5	0	0.5	
1.03	1.03	0.03	0.02	0	0.5	0.5	0.5	0	0.5	
2190.47	2190.47	72.58	25	0						
1.02	1.02	0.03	0.03	0	0.5	0.5	0.5	0	0.5	3
2190.47	2190.47	72.58	25	0						
4.01	5.09	0.14	0.02	0	0.5	0.5	0.5	0	0.5	
2190.47	2190.47	72.58	25	0						
322.36	520.29	3.36	3.17	0	76.5	732.4	793.47	410.15	1329.57	
2643.91	2666.78	79.05	22.22	0	478.55	925.24	925.24	631.72	1371.93	2
			5.93	1.55	1	1	1	0	1	
4.65	7.33	0.21	3.43	0	0.5	0.5	0.5	0	0.5	5
2190.47	2190.47	72.58	25	0						
2190.47	2190.47	72.58	25	0						

б) freezes, values %, intervals, correlation

Рисунок 4.24 – Таблиця «Всі сенсори машини»

Система виокремлює такі *типи констант*: Constant0, Constant1, Constant – відповідно, це ті сенсори, які мають одне окреме значення на всьому досліджуваному проміжку і воно дорівнює нулю, одиниці або будь якому іншому значенню, що не є нулем бо одиницею. Для констант не обраховується статистика, оскільки ці сенсори не мають жодної цінності при навчанні моделі. Вони позначені в звіті саме з цією метою – вказати на сміттєві дані.

Бінарні сенсори – такі сенсори, що мають 2 окремих значення. Система виокремлює такі типи бінарних сенсорів: 1) сенсори, що мають 2 окремих значення, і це нуль та одиниця, 2) сенсори, що мають 2 окремих значення, і хоча б одне з них не нуль і не одиниця. Бінарні сенсори в більшості своїй є індикативними, для них обраховуються всі показники в звіті. У випадку, якщо їхня кількість викликає сумніви, аналітику треба спілкуватися з клієнтом, щоб з'ясувати, які з цих даних є релевантними для тих типів подій, які треба прогнозувати.

Відсоток загальної тривалості пропущених або завмерлих даних може бути вказаний, як більший за 100%, і це зроблено навмисно. Якщо перший такий період почався до визначених меж для звіту і після початкової дати звіту тривалість більша за встановлену величину, система рахує повну довжину періоду. Такий же принцип і з пропуском/завмерлими даними наприкінці звіту. Це зроблено через те, що такі періоди без якісних даних є критичними для тренування моделі, і «загубити» тиждень неякісних даних, врахувавши тільки мінімальний проміжок, що потрапив в обраний інтервал – це викривити характеристики даних. Якщо аналітик хоче дослідити період поведінки без цього впливу, він обирає такі часові межі звіту, які відкидають довгі втрати даних цілком.

Характеристики часових інтервалів у ряді необхідні для наступного кроку роботи з даними – препроцесінгу. Аналітику треба обрати таке значення, яке буде допустимим згладжуванням для всіх обраних сенсорів. Якщо сирі дані надаються з частотою в кілька секунд чи хвилин, буде помилкою згладжувати резолюцію до години, і навпаки, дуже розряджені дані не варто перетворювати в дані з великою резолюцією. У цій таблиці можна побачити значення для всіх сенсорів одночасно і

вчасно помітити можливе джерело проблем через некоректно вибрану резолюцію оброблених даних або те, що набір даних має сенсори з частотою як в секунди, так і в години.

Якщо наявна велика різниця між мінімальним, медіанними та максимальними значенням тривалості інтервалів, це є показником проблеми з якістю даних.

3. Текстовий блок «Підсумки» («*Summary*») включає в себе переліки сенсорів із показниками, на які варто звернути увагу. Деякі переліки надані в табличному вигляді, деякі – просто текстові списки сенсорів. Підсумковий блок включає наступні розділи:

- таблиця бінарних сенсорів 0/1, що вказує кількість і відсоток для кожного значення;
- таблиця бінарних сенсорів зі значенням не 1 чи 0, що вказує значення, їхні кількість і відсоток;
- список небінарних сенсорів з малою кількістю окремих значень (більше 2 та менше 10);
- сенсори з великою кількістю нульових значень (більше 10%);
- список сенсорів із пропущеними значеннями (для даних з уніфікованими інтервалами);
- список сенсорів з негативними значеннями;
- список сенсорів з високим коефіцієнтом варіації (більше 1);
- список сенсорів з високим шансом наявності викидів (коефіцієнт ексцесу більше 5);
- список категорійних сенсорів;
- список сенсорів без даних;
- кореляційні групи (вказано кількість);
- часові інтервали (вказано кількість сенсорів, що мають непостійні часові інтервали, і максимальна кількість варіантів тривалості для одного сенсора);

- завмерлі дані (вказано кількість сенсорів з завмерлими даними, медіанна кількість і медіана тривалості таких періодів для одного сенсора);
- пропущенні дані (вказано кількість сенсорів з пропущеними даними, медіанна кількість і медіана тривалості таких періодів для одного сенсора).

Приклад текстового блока наведено у додатку А. Для прикладу було обрано проблемний набір даних із великою кількістю всіх перерахованих проблем. По всіх проблемних сенсорах можна отримати інформацію в табличному вигляді, завантаживши таблиці, наведені у додатку. В цьому блоці назви сенсорів навмисно здебільшого переведені в формат текстового рядка.

Вкладка Основна статистика включає в себе дві таблиці: Basic Statistics і Basic Descriptive Statistics та блок із графічними візуалізаціями.

Таблиця «Основні статистичні показники сенсорів» («Sensors Basic Statistics») для кожного сенсора надає такі характеристики: кількість та відсоток окремих значень, загальна кількість значень, кількість та відсоток негативних значень, тип даних (рис. 4.25).

Sensor name	Distinct values No	Distinct values (%)	Total No of values	Negative values No	Negative values (%)	Zero values No	Zero values (%)	Type
filter data.								
8	28.57	28	0	0	16	57.14	Numeric	
8	25.81	31	0	0	17	54.84	Numeric	
8	24.24	33	0	0	18	54.55	Numeric	
6	0.02	37636	0	0	37554	99.78	Numeric	
255835	97.92	261257	0	0	12	0	Numeric	
3	0.19	1557	0	0	0	0	Numeric	
2	1.92	104	0	0	55	52.88	Binary	
2	1.57	127	0	0	66	51.97	Binary	
2	1.34	149	0	0	77	51.68	Binary	
1	25	4	0	0	0	0	Constant 1	
421	4.03	10452	0	0	0	0	Numeric	
95372	31.04	307282	0	0	0	0	Numeric	
110720	35.85	308825	0	0	0	0	Numeric	

Рисунок 4.25 – Таблиця Basic Statistics вкладки «Основна статистика»

Блок із графічними візуалізаціями «Розподіл значень сенсорів» («Sensors Values Distribution») містить гістограми розподілу значень для кожного сенсора надано та таблицю значень і кількості входжень кожного для 20 найуживаніших

значень (рис. 4.26). Для цієї візуалізації можна налаштувати кількість бінів та відфільтрувати сенсори з малою кількістю унікальних значень.

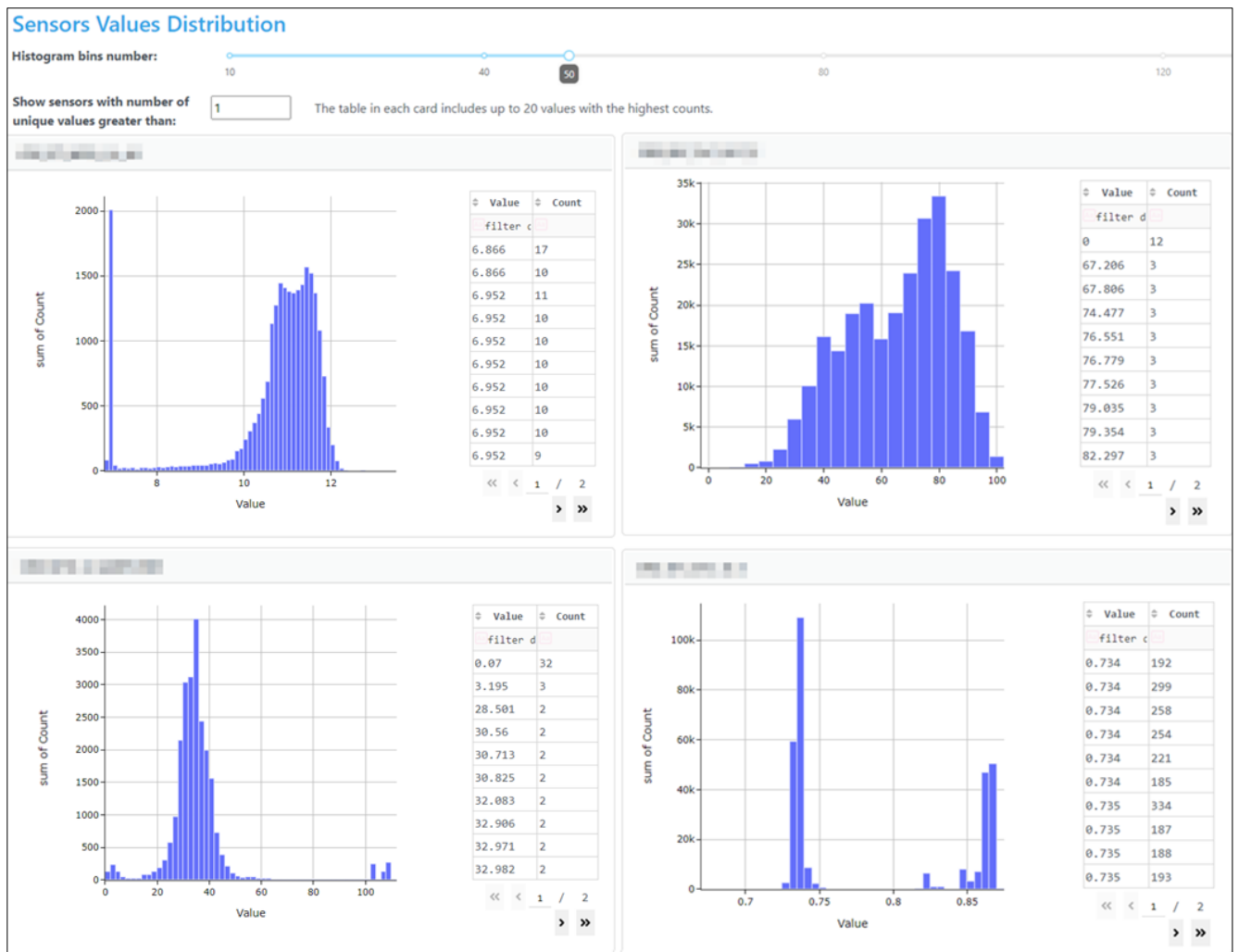


Рисунок 4.26 – Візуалізація гістограм розподілу даних сенсорів

Таблиця «Описова статистика сенсорів» («*Sensors Basic Statistics*») для кожного сенсора надає такі характеристики: значення (рис. 4.27): мінімальне, перцентилі (5, 25, 75, 95), медіана, середнє, максимальнє; стандартне відхилення; середнє абсолютне відхилення; коефіцієнт варіації; міжквартильний розмах; різниця значень; коефіцієнт ексцесу; коефіцієнт асиметрії.

Кафедра інтелектуальних інформаційних систем
Інформаційна система аналізу показників сенсорів технологічних ліній

Sensor name	Minimum value	5%	25%	Median	Average sensor value	75%	95%	Maximum value	Standard deviation	MAD	CV	IQR	Range of values	Kurtosis	Skewness
filter data.															
	0	0	0	0	4.61	4.5	21.4	41	9.13	0	1.98	4.5	41	10.05	3.06
	0	0	0	0	4.55	5	18.5	41	8.72	0	1.92	5	41	10.98	3.15
	0	0	0	0	4.48	6	17.6	41	8.65	0	1.93	6	41	11.19	3.2
	0	0	0	0	0.02	0	0	8	0.37	0	21.51	0	8	463.38	21.55
	0	34.47	51.8	68.82	65.64	80.02	90.99	100	17.93	13.8	0.27	28.2	100	-0.79	-0.39
	1	1	1	2	1.74	2	4	4	0.95	1	0.54	1	3	0.99	1.36
	0	0	0	0	0.47	1	1	1	0.5	0	1.06	1	1	-2.03	0.12
	0	0	0	0	0.48	1	1	1	0.5	0	1.04	1	1	-2.03	0.08
	0	0	0	0	0.48	1	1	1	0.5	0	1.04	1	1	-2.02	0.07
	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
	0.49	0.72	0.72	0.73	0.78	0.85	0.86	0.86	0.06	0.01	0.08	0.13	0.37	-1.43	0.25
	0.67	0.73	0.73	0.74	0.79	0.86	0.87	0.87	0.06	0	0.08	0.13	0.2	-1.79	0.42
	0.67	0.73	0.73	0.74	0.79	0.86	0.87	0.87	0.06	0	0.08	0.13	0.2	-1.78	0.42

Рисунок 4.27 – Таблиця Basic Descriptive Statistics вкладки «Основна статистика»

Вкладка Аналіз замерлих даних містить інформаційну таблицю для всіх сенсорів із замерлими даними (рис. 4.28), візуалізацію шуканих періодів на часовій прямій (рис. 4.29) та блок карток із переліком таких періодів для кожного сенсора (рис. 4.30).

Останній сенсор на візуалізації має періоди замерлих даних довгої тривалості. В назві сенсора присутній префікс «status». Це цінна знахідка – можна порівняти наявний журнал подій зі статусом, який наочно стабільний при нормальній роботі механізму і має змінювану поведінку в короткі періоди часу. Інформаційна таблиця теж показує, що тривалість періодів замерлих даних для цього сенсора незрівнянно більша за тривалість незмінюваних даних для всіх інших сенсорів.

Sensor	Longest freeze start	Longest freeze end	Longest freeze duration (hrs)	Number of freezes	Total time freeze (hrs)	Freezes from total duration (%)
filter data...						
	2018-07-10T15:15:00	2018-07-10T20:45:00	5.5	30	61.5	0.24
	2018-06-06T17:00:00	2018-06-07T00:45:00	7.75	3	23	0.09
	2018-06-07T02:45:00	2018-06-07T18:15:00	15.5	3	31	0.24
	2018-12-12T06:30:00	2018-12-13T18:45:00	36.25	6	69.25	0.27
	2018-06-17T14:30:00	2018-06-18T06:00:00	15.5	24	193	0.75
	2018-12-19T04:45:00	2018-12-19T17:30:00	12.75	14	59.5	0.23
status_	2018-06-13T21:45:00	2018-08-03T02:15:00	1204.5	27	5933.5	44.05

Рисунок 4.28 – Інформаційна таблиця для всіх сенсорів із замерлими даними вкладки «Аналіз замерлих даних»

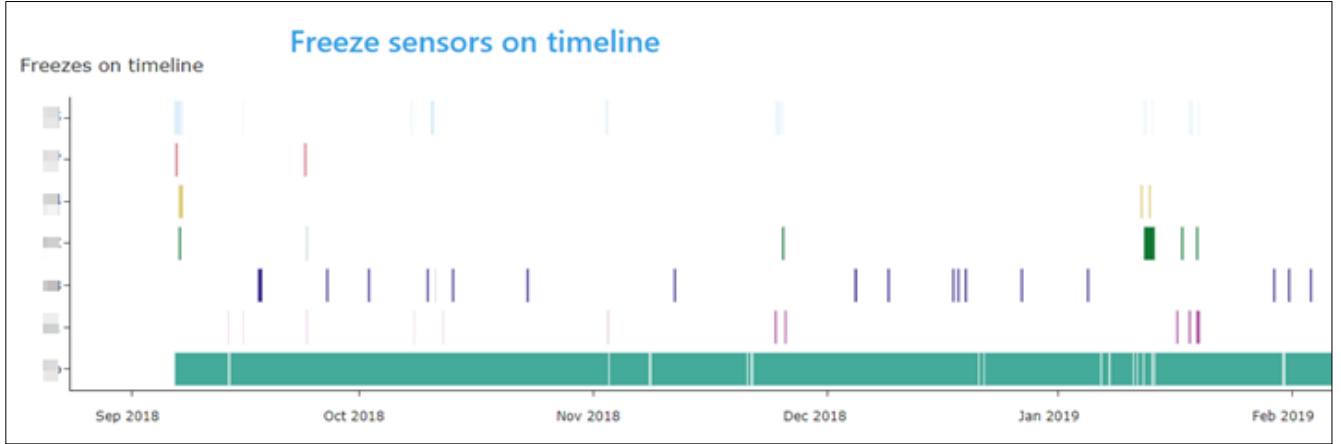


Рисунок 4.29 – Візуалізація шуканих періодів на часовій прямій вкладки «Аналіз завмерлих даних»

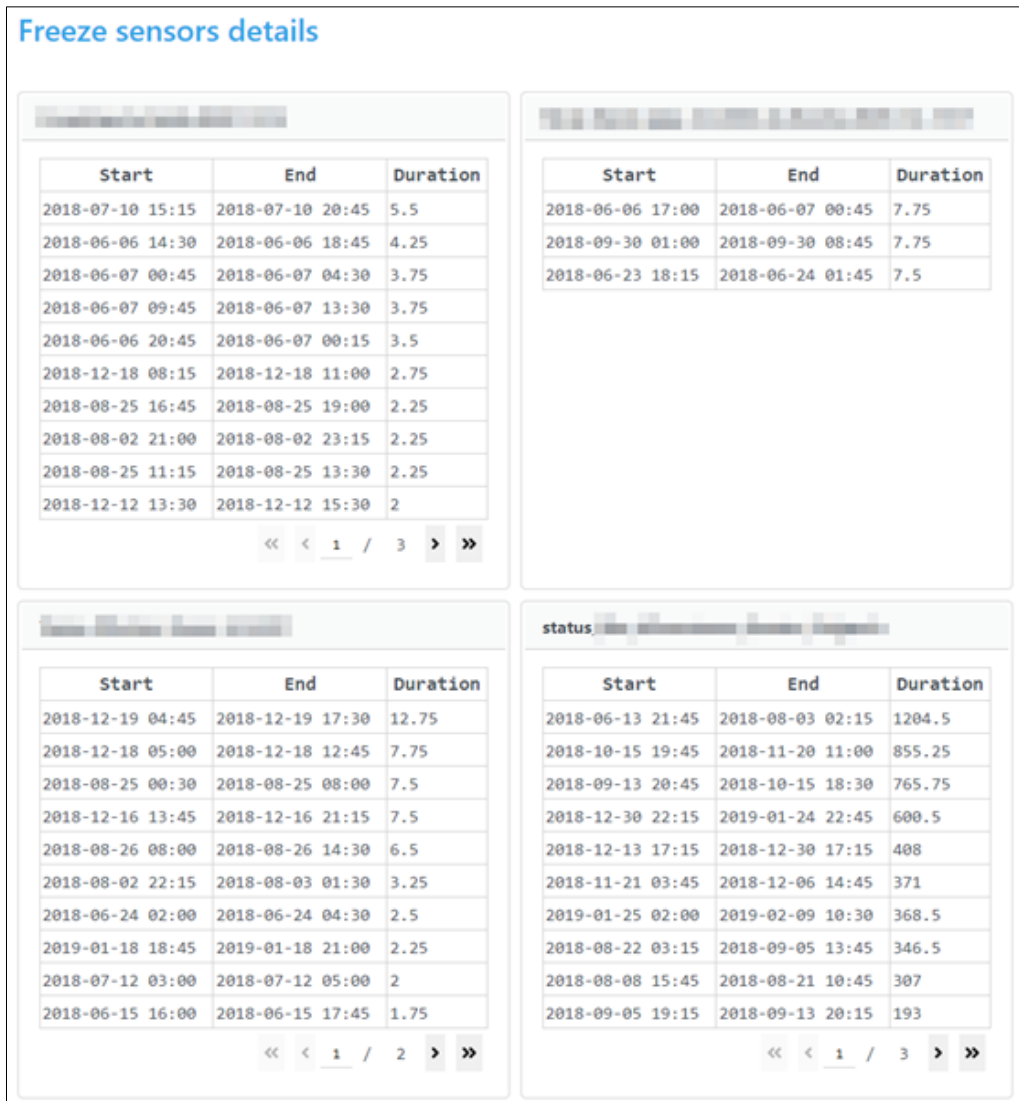


Рисунок 4.30 – Картки із переліком шуканих періодів для кожного сенсора вкладки «Аналіз завмерлих даних»

Вкладка Кореляційний аналіз надає інформацію про групи сенсорів із високим коефіцієнтом кореляції, візуалізуючи їх за допомогою теплової карти (рис. 4.31), двох інформаційних таблиць і візуалізації поведінки сенсорів групи.

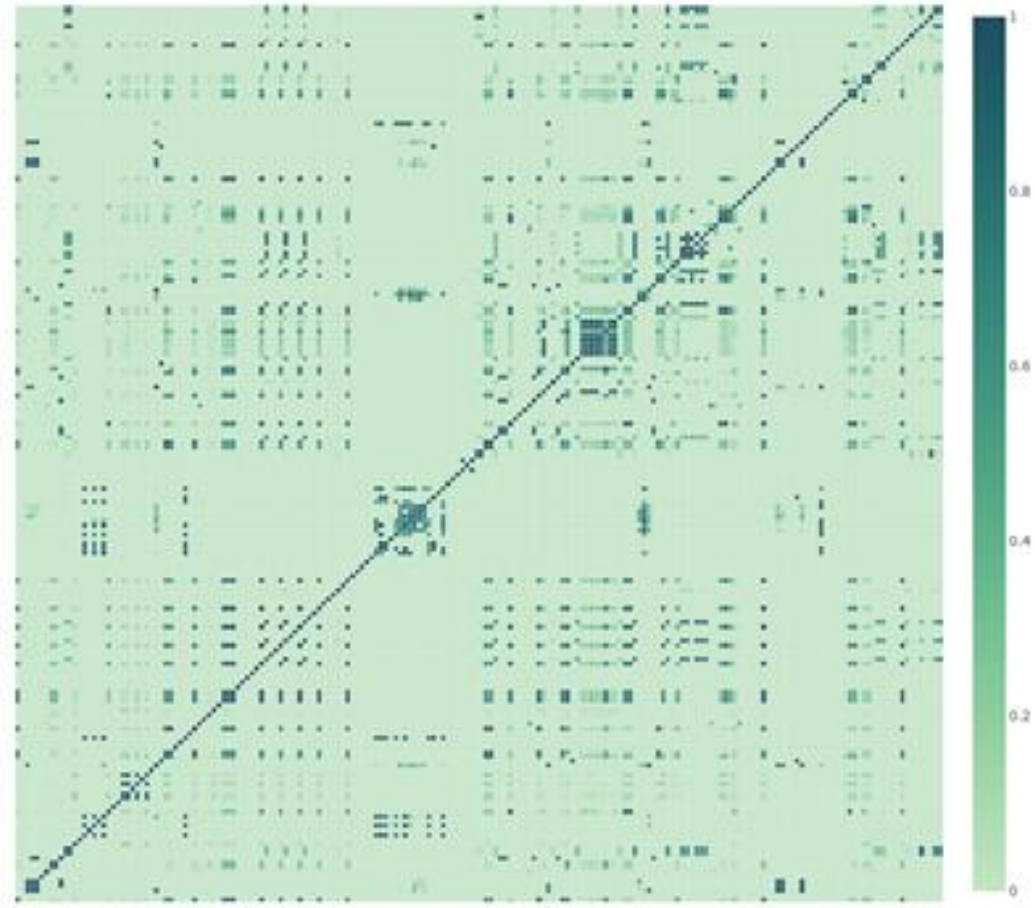


Рисунок 4.31 – Теплова карта коефіцієнтів кореляції

Кореляції не обраховуються для сенсорів, кількість значень яких відрізняється більше, ніж в двічі, щоб уникнути помилково високих коефіцієнтів кореляції між спокійними даними (з низьким коефіцієнтом відхилення) та відсутніми даними (NaN) та між відсутніми даними для обох сенсорів. Теплова карта за замовчуванням показує всі сенсори машини, але їх можна відфільтрувати тільки до обраних в картці керування даними. Також можна вивести на осі назви сенсорів. Приклад різних варіантів вигляду теплової карти наведено на рисунку 4.32.



Рисунок 4.32 – Варіанти відображення теплової карти

Таблиця «Аналіз поведінкової кореляції сенсорів по групах» містить інформацію про кількість груп та сенсорів у кожній групі (рис. 4.33).

Feature Correlation Sensors Groups Analysis

Export

Group ID	Highly correlated sensor	Group size
filter data...		
1		15
2		13
3		9
4		9
5		7
6		6

Рисунок 4.33 – Фрагмент таблиці «Аналіз поведінкової кореляції сенсорів по групах»

Таблицю відсортовано по спаданню розміру групи. Перші групи, що містять близько та більше 10 сенсорів з коефіцієнтом кореляції близьким до 1 або -1 можуть містити дубльовану інформацію. Аналітик перевірить сенсори, перелічені в цих групах, щоб переконатися, що вага аномалій, визначених цими сенсорами не була штучно завищена.

Аналіз поведінки сенсорів «Найбільш корельовані сенсори» складається з власне графіків, які відображають поведінку сенсорів групи (рис. 4.34), та таблиці, що містить інформацію про коефіцієнти кореляції в групі (рис. 4.35).



Рисунок 4.34 – Візуалізація поведінки сенсорів у групі

Дані з дуже високою резольуцією проходять ресемплінг до частоти в три години – цей графік не має на меті точно зобразити поведінку сенсора в кожний окремий момент часу, він фокусується на зображенні головних патернів, тенденцій одночасної зміни даних різних сенсорів. Для відображення поведінки можна обрати формат маркерів – точки чи лінії.

Sensor name	Sensors correlated	Correlation coefficient
filter data...		
	C06_MMS_2484_B	0.81
	C06_MMS_2484_B	0.82
	C06_MMS_2484_B	0.82
	C06_MMS_2484_B	0.83
	C06_MMS_2484_B	0.99

Рисунок 4.35 – Таблиця з інформацією про коефіцієнти кореляції в групі сенсорів

Вкладка Аналіз пропусків та часових інтервалів включає табличне та графічне відображення особливостей пропущених даних та резольуції наявних даних.

Для опису пропущених даних застосовується таблиця та діаграма Ганта. Діаграма Ганта розміщує проміжки часу, позначені кольором, на числовій прямій. При наведенні курсора на проміжок, надається інформація про тип проміжку, його часові межі та назву сенсора (рис. 4.36).

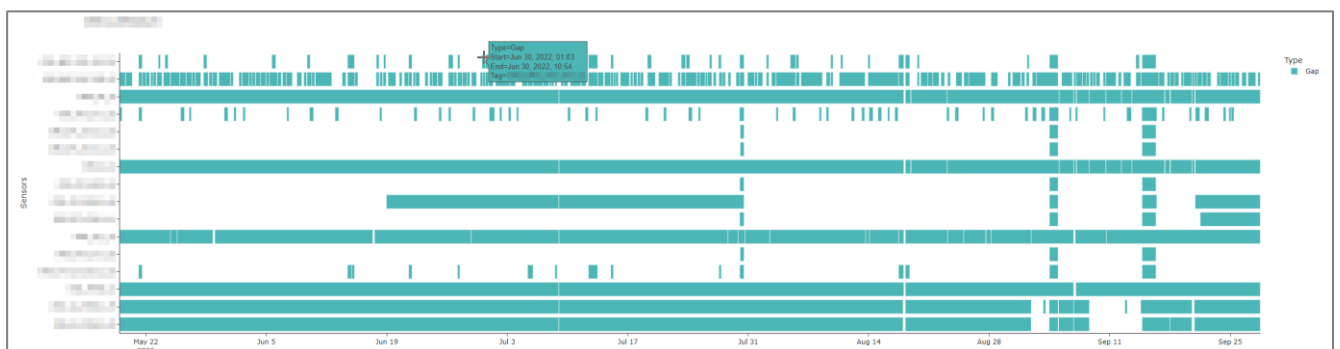


Рисунок 4.36 – Діаграма Ганта для періодів пропущених даних

При наведенні вказівника миші на інформаційну іконку відображується поп-ап виводиться інформація про те, які дані увійшли у таблицю, відфільтровані за замовчуванням проміжки, можливість змінювати інформацію в таблиці, змінюючи фільтри на вкладці (рис. 4.37).

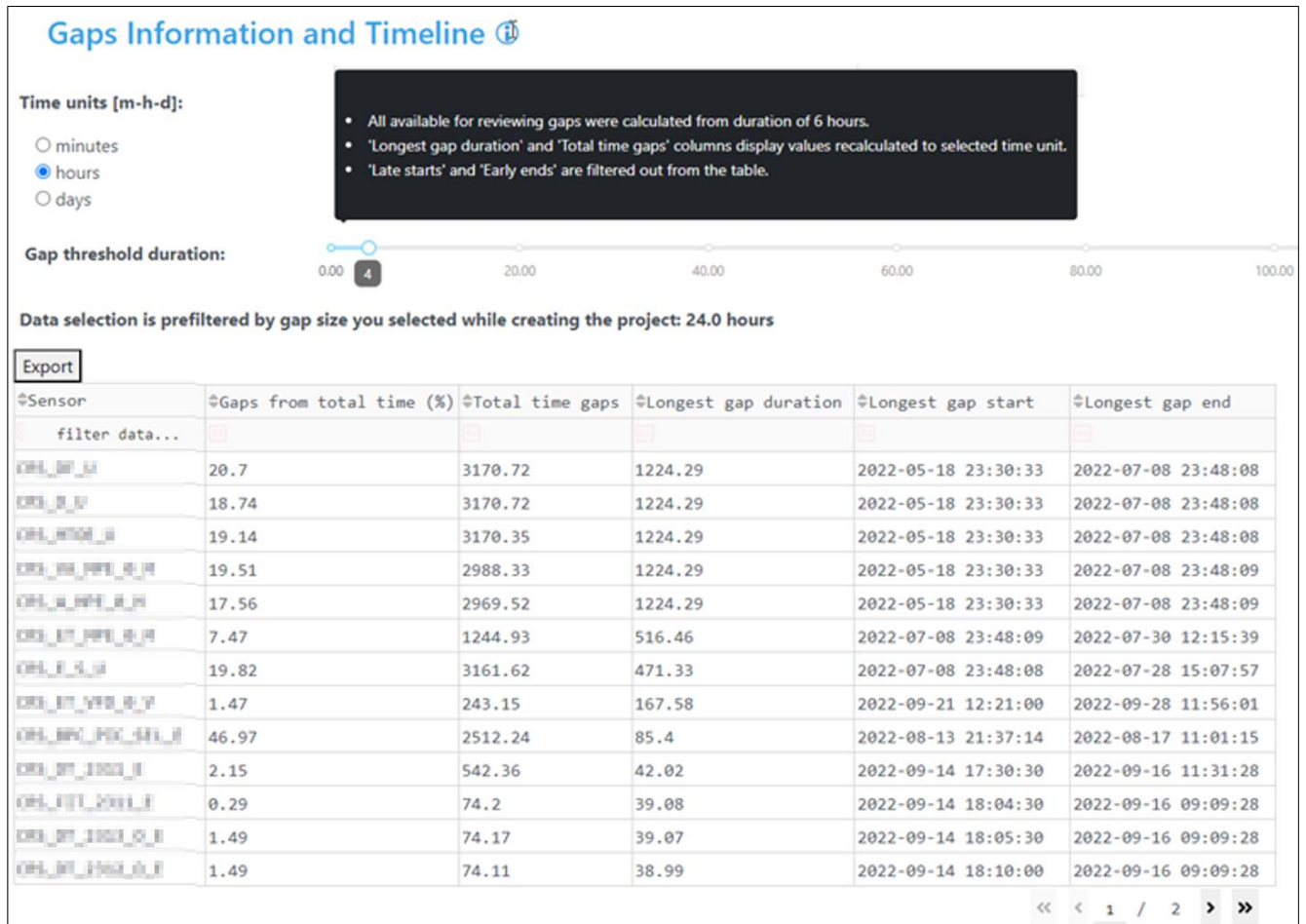


Рисунок 4.37 – Таблиця з інформацією про періоди пропущених даних

Блок, присвячений інформації про часові проміжки, складається з інформаційної таблиці зі статистичними показниками проміжків, діаграми розмаху та точкових діаграм з лініями похибок для кожного сенсора. Інформаційні іконки у заголовку блоку та назві таблиці надають інформацію про спосіб обробки даних та можливість змінювати одиниці виміру часу – дані на графіках та в таблицях відповідно оновляться (рис 4.38).

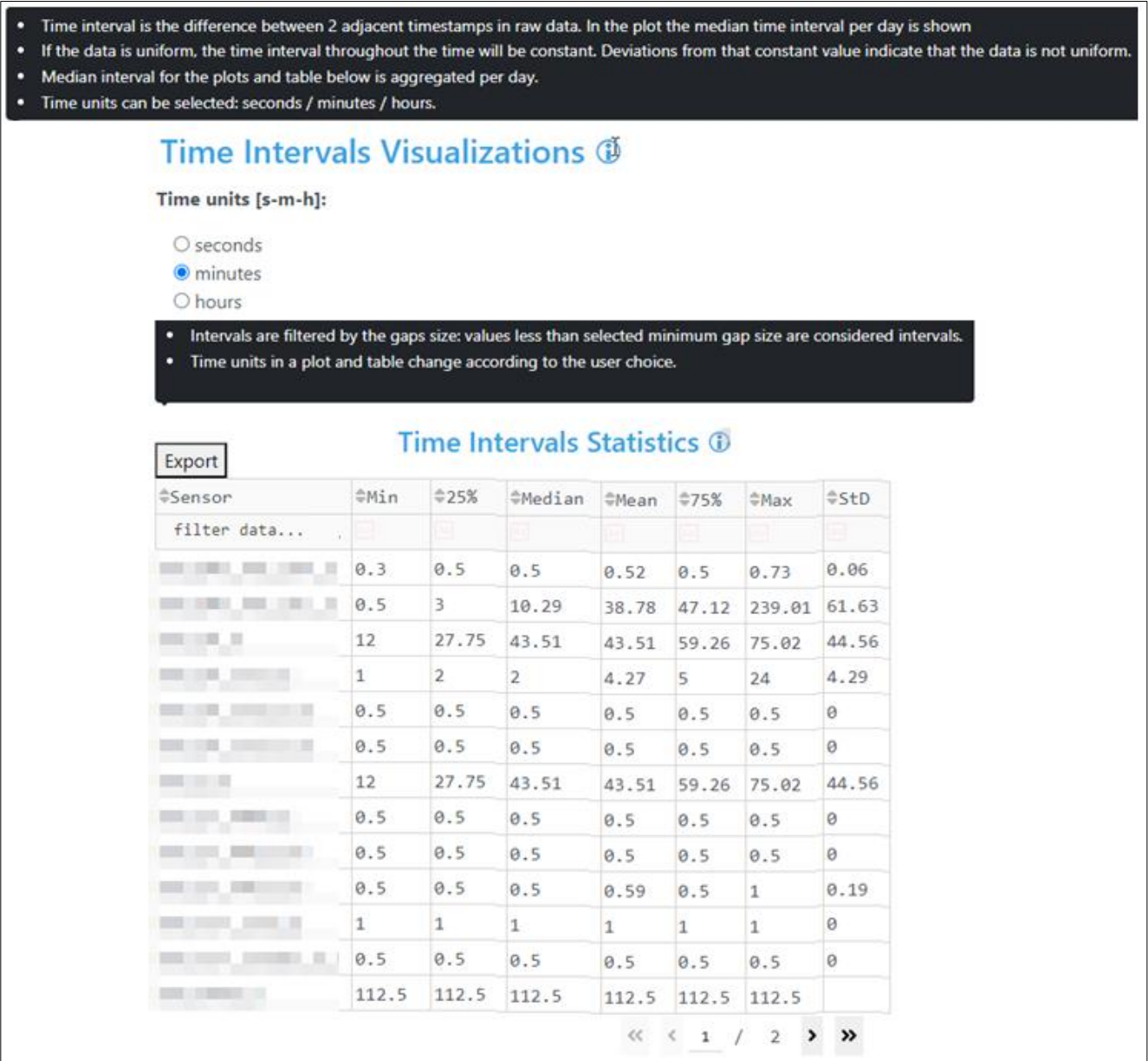


Рисунок 4.38 – Таблиця статистичних показників часових проміжків блоку «Візуалізація інтервалів»

Діаграма розмаху наводить статистичні дані у графічному вигляді (рис 4.39). Передивляючись разом таблицю та діаграму, аналітик визначає, чи є певні «стандартні» значення резольюції цього набору, якими були б ці дані, якби не втрачені дані, які можливі варіанти значень резольюції на подальших кроках роботи.

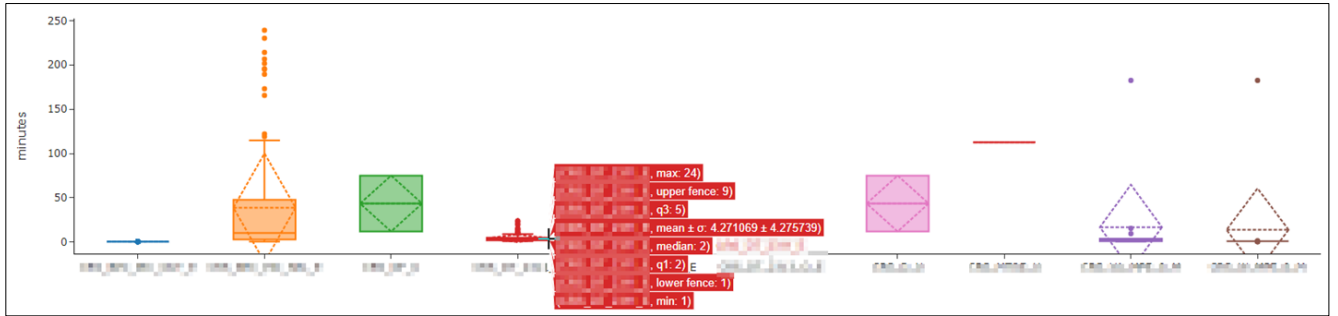


Рисунок 4.39 – Діаграма розмаху значень тривалості часових проміжків

Точкові діаграми з лініями похибок для значень тривалості часових проміжків (рис. 4.40) відображають медіану і похибки тривалості часових проміжків за добу для кожного сенсора.

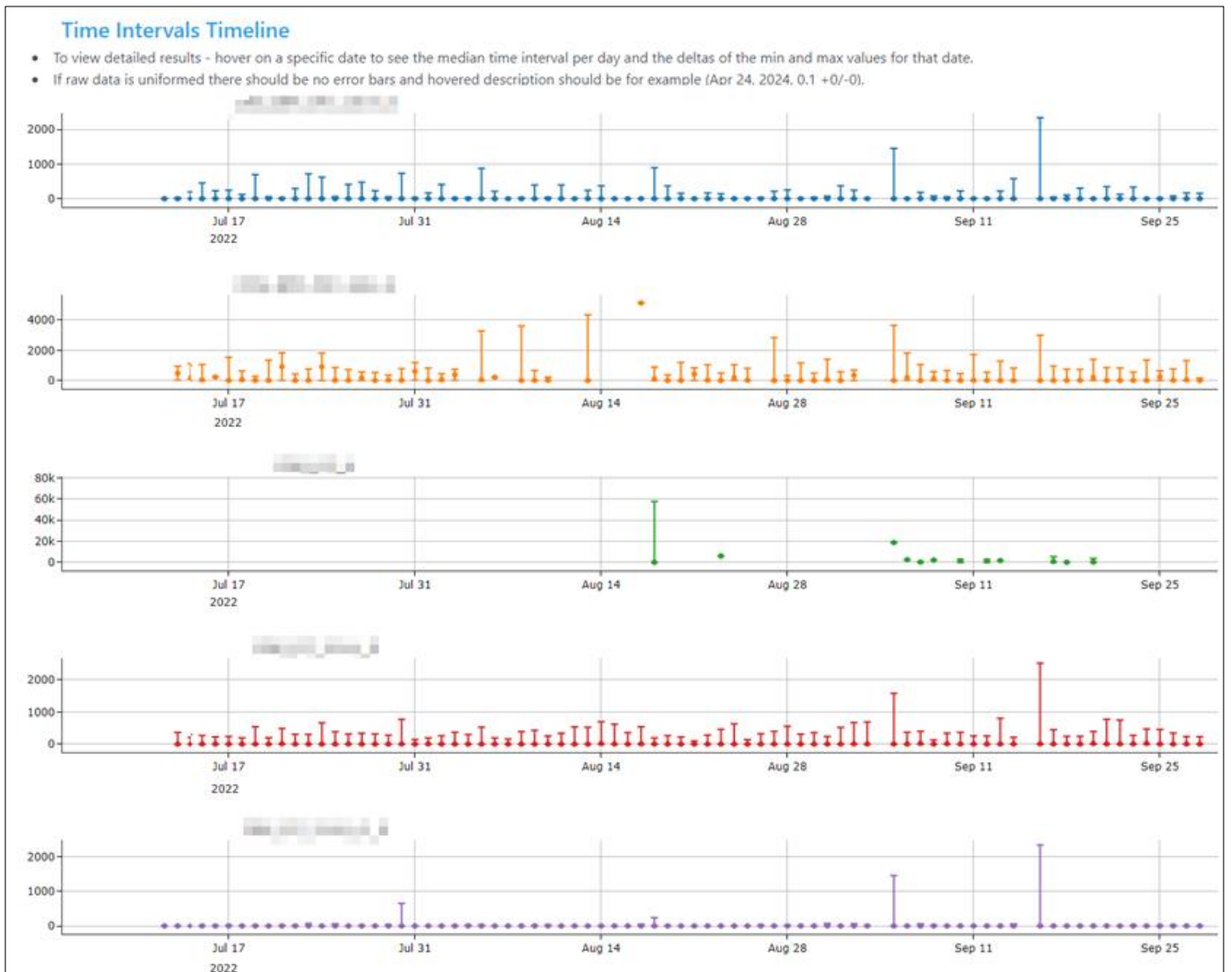


Рисунок 4.40 – Точкові діаграми з лініями похибок для значень тривалості часових проміжків

Дані у точкових діаграмах наводяться в обраних одиницях – секундах, хвилинах або годинах. Крім наочної інформації про частоту сирих даних, ця візуалізація надає інформацію про наявність певних періодів, коли більшість сенсорів надавали дані нерівномірно, та про наявність сенсорів, які більшість часу надають інформацію нестабільно.

Впровадження запропонованого підходу дозволило покращити відбір та підготовку даних для подальшого аналізу. Інформація про роздільність сирих даних стала доступною для всіх сенсорів одночасно, з візуалізацією і можливістю порівняти різні групи сенсорів, визначивши, які значення прийнятні для всіх видів історичних даних. Здійснився перехід до аргументованих рішень для кожного набору даних окремо. Було прискорено та покращено вибір методу заповнення пропусків у даних сенсорів, виявлення та аналіз викидів і аномалій. Аналіз розподілів показників сенсорів дозволяє відкоригувати спосіб навчання прогнозної моделі. Результати аналітики по конкретному виробництву фахівці Data Science отримують досить швидко, що прискорює комунікацію з працівниками виробництва для швидкого вирішення питань стосовно врахування їх специфіки.

4.7 Тестування та оцінка якості інформаційної системи

Результати аналізу використовуються: операційним відділом, відділом роботи з клієнтами, науковим департаментом. Подальші етапи роботи з даними включають: 1) препроцесінг даних (уніфікація частоти даних, заповнення пропусків), 2) визначення аномалій, 3) налаштування та тренування моделей, 4) інтерпретацію отриманих прогнозів подій. Розглянемо використання системи аналізу показників сенсорів різними підрозділами на кожному з етапів.

Операційний відділ для підготовки налаштувань *препроцесінгу* має визначити кінцеву частоту уніфікованих даних та спосіб заповнення пропусків. Отримавши табличний аналіз та візуалізацію часових інтервалів (вкладки Основні статистики

та Аналіз пропусків та часових інтервалів), аналітик у випадку якісних, «простих» даних визначає:

- частоту, прийнятну для обробки без втрати якості;
- тривалість пропусків, які не заповнюються даними (довші за медіану або середнє значення для машини);
- спосіб заповнення пропусків для решти періодів.

У випадку проблемних даних, надсилає переліки, таблиці проблемних даних та посилання на звіт колегам з відділу роботи з клієнтами.

Відділ роботи з клієнтами формулює список запитань для уточнення у клієнта і з'ясовує причину таких нерівномірних даних. У випадку, продемонстрованому на рисунку 4.41, було з'ясовано, що збереження даних на боці клієнта налаштовано таким чином, що дані передаються, тільки коли змінюються. Ці машини в нормі працюють безперервно, з регулярним вимкненнями для профілактичних робіт, які тривають до 3-х годин.

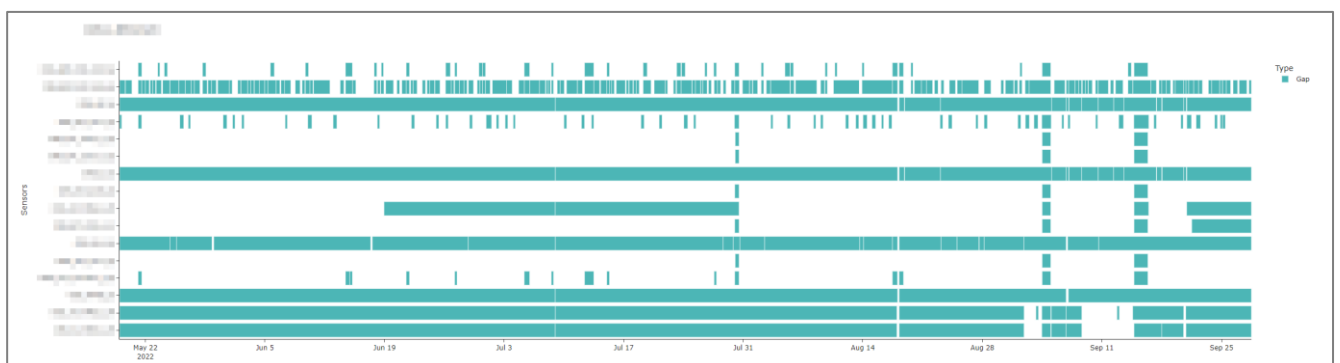


Рисунок 4.41 – Приклад даних з великою кількістю пропущених даних

Таким чином, з'ясувалося, що ті дані, які на перший погляд здавалися «сміттєвими», і були б виключені при тренуванні моделі, насправді є даними, які найкраще підходять для визначення аномалій та тренування моделі, адже більшу частину часу їхня поведінка стабільна, і саме інтенсивність змін значень є приводом для визначення аномалій.

У цьому випадку для значення уніфікованої частоти даних було обрано за медіаною для більшості сенсорів і згладжено від 2,75 хвилини (середнє для 30 секунд та 5 хвилин) до 15 хвилин; пропуски було заповнено останнім значенням перед пропуском для всіх періодів, довгих за 3 години.

Операційний відділ на етапі *визначення аномалій* має вказати, у скількох режимах працює машина, і у випадку «multimode» вказати, скільки режимів, за значеннями яких сенсорів визначаються перемикання.

Гістограми розподілу даних та виокремлення бінарних сенсорів в окремий тип серед числових даних та сенсорів з низькою кількістю окремих значень в окремий перелік дозволяють аргументовано запідозрити наявність такої специфіки, навіть коли вона не була вказана клієнтом. В прикладі, що ми розглядаємо, клієнт не повідомив про багаторежимний випадок, і перші спроби визначення аномалій і тренування моделей на базі цих аномалій були дуже невдалі. Це підприємство в сфері видобутку та логістики нафтопродуктів: машини з сотнями сенсорів і дуже високими вимогами до відсутності хибно негативних прогнозів. Дані було передано в науковий відділ, щоб знайти рішення. Ці дані було використано як тестові при розробці інформаційної системи аналізу показників сенсорів. На рисунку 4.42 наведено приклад такої візуалізації.

У такому випадку знов *відділи операційної діяльності і роботи з клієнтами* працюють разом. Аналітики-операціоністи надсилають колегам з відділу роботи з клієнтами посилення на звіт та списки сенсорів, які виглядають як індикативні, і сенсорів, які передають дані з різних режимів в одному потоці. У цьому випадку після підтвердження клієнтом такого режиму роботи машини і надання необхідної додаткової інформації, було внесено такі зміни для конфігурації:

- встановлено флаг багаторежимної роботи машини;
- позначено сенсори та їхні значення, за якими визначаються перемикання режимів;
- визначено тривалість «сліпих зон» для визначення аномалій на межах періодів перемикання;

- журнал подій розділено на декілька, відповідно до кількості режимів і часу, коли відбувалася подія, позначено як декілька класів подій;
- натреновано кілька моделей;
- отримані кілька потоків об'єднані в один на фінальному етапі – для видачі інформації про статус машини і прогнозів подій клієнтові.

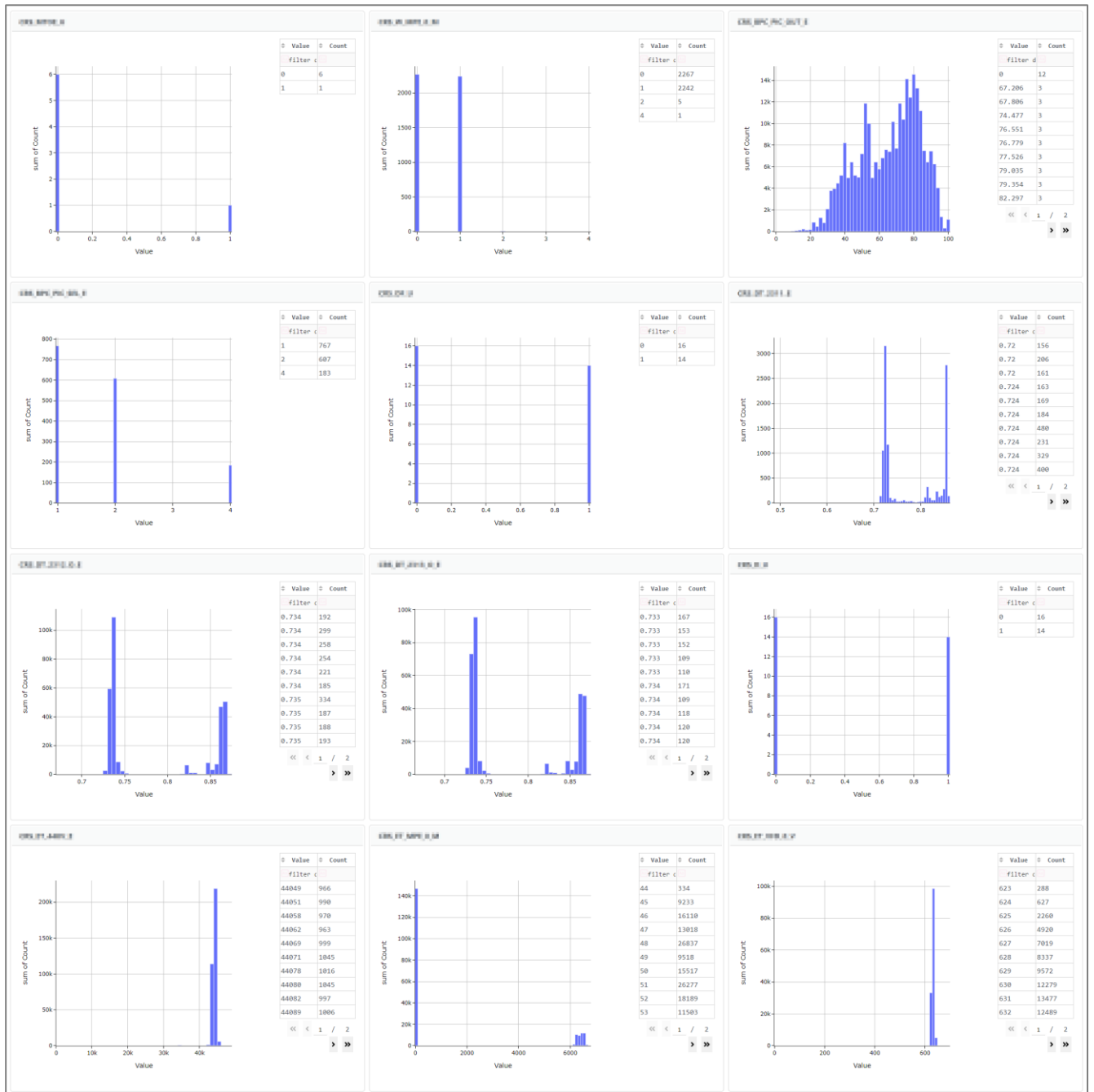


Рисунок 4.42 – Приклад розподілу даних при роботі машини в двох режимах

Це дозволило значно підвищити якість прогнозування і продовжити співпрацю з клієнтом, додавши інші машини в клієнтську базу.

Операційний відділ (за необхідності підключається науковий департамент) тренує моделі на тренувальній частині історичних даних, отримує прогнози для тестової частини історичних даних і обирає найкращу модель.

В описуваному прикладі всі отримані моделі з прийнятною якістю прогнозування мали явні ознаки оверфітінгу. Додатковий аналіз із застосуванням розробленої аналітичної системи даних сенсорів виявив кілька груп високорельованих даних із великою кількістю сенсорів. Як виявилось, дані дублювалися через наявність сенсорів всередині механізмів і на зовнішніх корпусах вузлів та через автоматичну обробку даних на стороні клієнта (деякі сенсори зберігалися в сирому та нормалізованому вигляді, додавалася дані після ресемплінгу (рис. 4.43, 4.44).

Feature Correlation Sensors Groups Analysis

Export

Group ID	Highly correlated sensor	Group size
1	POS_FT_001	48
2	POS_FT_002	34
3	POS_FT_003	21
4	POS_FT_004	4
5	POS_FT_005	3
6	POS_FT_006	3
7	POS_FT_007	2
8	POS_FT_008	1
9	POS_FT_009	1
10	POS_FT_010	1
11	POS_FT_011	1
12	POS_FT_012	1
13	POS_FT_013	1
14	POS_FT_014	1

Рисунок 4.43 – Виявлення завеликих груп сенсорів із високим коефіцієнтом кореляції

Sensor name	Sensors correlated	Correlation coefficient
filter data...		
PSL_FL_001	PSL_FL_001	1
PSL_FL_001	PSL_FL_001	1
PSL_FL_001	PSL_FL_001	1
PSL_FL_001	PSL_FL_001	1
PSL_FL_001	PSL_FL_001	1
PSL_FL_001	PSL_FL_001	1
PSL_FL_001	PSL_FL_001	0.98
PSL_FL_001	PSL_FL_001	0.93
PSL_FL_001	PSL_FL_001	0.92
PSL_FL_001	PSL_FL_001	0.91
PSL_FL_001	PSL_FL_001	0.91

Рисунок 4.44 – Значення коефіцієнтів кореляції в проблемній групі сенсорів

З набору даних було виключено більшість сенсорів, перелічених в знайдених групах. Нові натреновані моделі в звітах інтерпретації прогнозів стали надавати ті сенсори, які не згадувались під час першої ітерації проекту. Це дозволяє зробити висновок, що аномалії меншої сигнатури почали враховуватися алгоритмом, оскільки була прибрана зайва вага для аномалій, що виникла в наслідок дублювання даних. Видалені сенсори не вважаються втраченою інформацією, оскільки це була дубльована інформація, яка спотворювала криві аномалій.

Для дослідження якості розробленої системи аналізу показників сенсорів технологічних ліній було проведено тренування моделей на тестовому наборі юзкейсів, який використовується для обрахунку зміни KPIs для кожної версії алгоритму. Було проведено аналіз даних, внесено зміни в конфігурації на базі отриманих результатів, натреновано і обрано моделі.

Застосування системи дозволяє автоматизувати процеси виявлення проблемних даних. Її впровадження в обслуговування технологічних ліній важкої промисловості супроводжувалося підвищенням якості моделювання і точності прогнозів поломок. Оцінка точності моделей здійснювалася за показником F1-score:

$$F = \frac{2 \cdot PREC \cdot SN}{PREC + SN}, \quad (4.1)$$

де $PREC$ – точність позитивних результатів (англ. Precision),
 SN – частота відклику (англ. Recall).

$PREC$ та SN – розраховуються відповідно за формулами:

$$PREC = \frac{TP}{FP + TP}, \quad (4.2)$$

$$SN = \frac{TP}{TP + FN}, \quad (4.3)$$

де TP (англ. True Positive) – кількість показників, які є критичними, правильно спрогнозованих як критичні;

TN (англ. True Negative) – кількість показників, які не є критичними, правильно спрогнозованих як не критичні;

FP (англ. False Positive) – кількість показників, які є критичними, не віднесені моделлю до таких;

FN (англ. False Negative) – кількість показників, які не є критичними, не віднесені моделлю до таких.

На рисунку 4.45 показано збільшення у середньому на 10% показника F1-score після застосування запропонованого підходу.

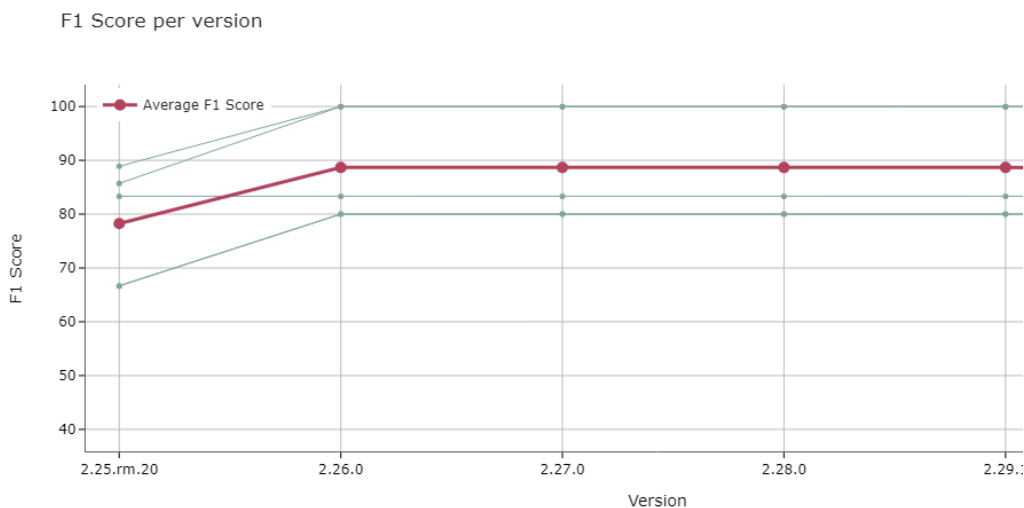


Рисунок 4.45 – Зміна якості прогнозних моделей технічного обслуговування після застосування розробленого аналітичного інструментарію

Іншим результатом стала зміна кількості ітерацій, необхідних для отримання моделі прийнятної якості (рис. 4.46). Порівняльний аналіз отриманих результатів наведено у таблиці 4.2.

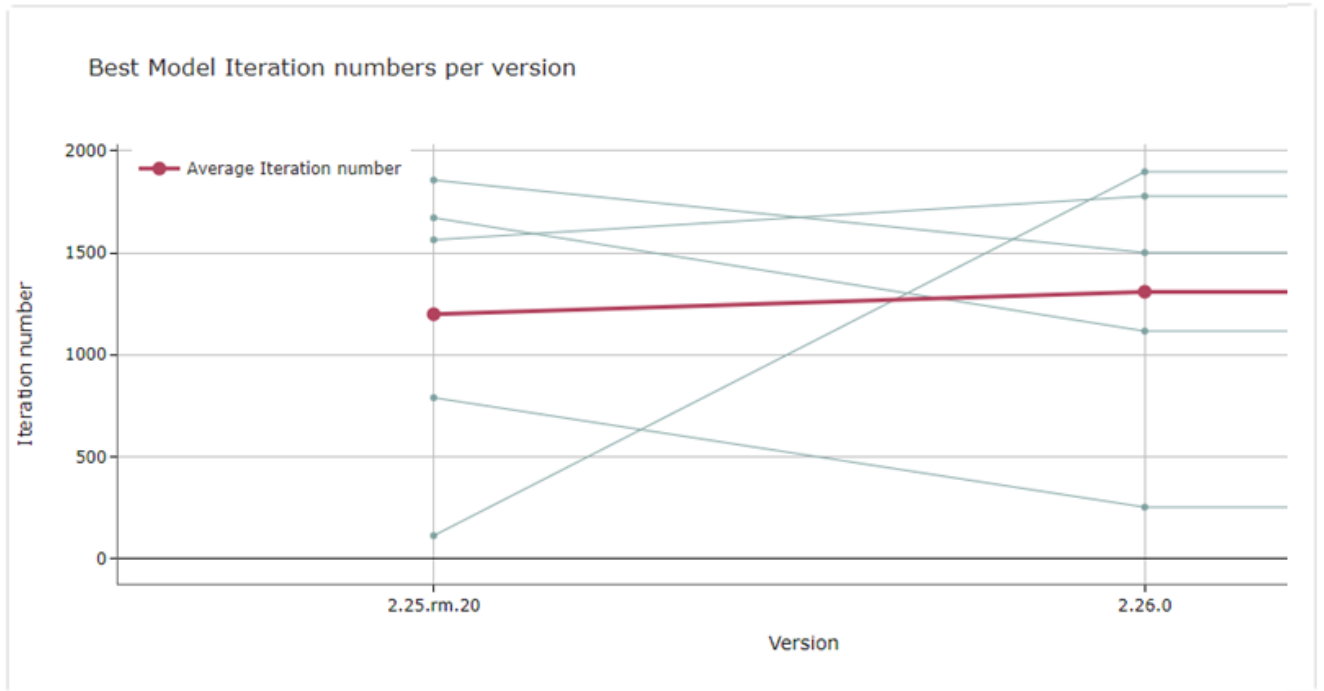


Рисунок 4.46 – Зміна кількості ітерацій тренування прогнозних моделей після застосування розробленого аналітичного інструментарію

Таблиця 4.2 – Порівняння якості прогнозів і швидкості тренування моделей

№ набору даних	До застосування аналітичної системи		Після застосування аналітичної системи		Різниця	
	Значення F1 score	№ ітерації	Значення F1 score	№ ітерації	Значення F1 score	№ ітерації
1	88,89	789	100	252	11,11	-537
2	85,71	1564	100	1778	14,29	214
3	83,33	1857	83,33	1501	0	-356
4	66,67	1672	80	1116	13,33	-556
5	66,67	112	80	1898	13,33	1786

Слід зазначити, що для п'ятого набору даних відбувалася конвергенція моделі після 112 ітерації. Набори даних представляють такі сфери виробництва: 1-3 – гідравлічні насоси, 4 – вітрові генератори, 5 – виплавка чавуну. Всі ці дані є даними з підприємств, і були обрані як тестові юзкейси через складності, з якими ми стикнулися, тренуючи моделі для прогнозування критичних подій на цих підприємствах. Межею прийнятної якості прогнозування є значення F1 score ≥ 80 , кількість ітерації при тренуванні моделей не перевищує 2000.

За результатами аналізу можна побачити, що після переходу від стандартизованих правил до “dataknowledge based configuration” в 4 з 5 випадків якість результатів прогнозування збільшилася: зріст значення показника F1 score – від 11,11 до 14,29, при чому в 2-х випадках модель перетнула межу прийнятної якості. В 3 з 5 випадках зменшилася кількість часу, необхідного для отримання оптимальної моделі, в одному випадку час збільшився, проте це дозволило отримати модель потрібної якості. В іншому випадку, де час навчання моделі збільшився, якість прогнозів моделі досягла 100% точності.

Застосування системи дозволяє автоматизувати процеси виявлення проблемних даних. Її впровадження в обслуговування технологічних ліній важкої промисловості супроводжувалося підвищенням якості моделювання і точності прогнозів поломок.

Таким чином, проведене дослідження дозволило виявити методи аналізу великої кількості показників сенсорів технологічних ліній, які сприяють підвищенню якості моделей прогнозування поломок, зменшуючи час їх виявлення. Впровадження розробленої на основі описаного підходу автоматизованої системи виявлення проблемних даних у показниках сенсорів дозволило підвищити ефективність прогнозного технічного обслуговування технологічних ліній. Що є особливо цінним для підприємств важкої промисловості, де помилка у прогнозуванні може спричинити значні фінансові та екологічні втрати. Створення інформаційних систем, які орієнтовані на предикативне технічне обслуговування конкретних підприємств важкої промисловості із урахуванням їх специфіки,

запобігає виходу з ладу та простою обладнання і подовжує загальний термін служби життєво важливих активів.

Висновки до розділу 4

Здійснено моделювання, розробку та програмну реалізацію інформаційної системи аналізу показників сенсорів технологічних ліній. Описано послідовність етапів роботи з базами даних, які зберігають набори даних з показниками сенсорів. Особливу увагу приділено проектуванню та розробці діаграми діяльності аналізу показників сенсорів та діаграми сценаріїв використання.

Розроблено систему, яка автоматизує аналіз даних сенсорів технологічних ліній, генерує списки потенційних джерел проблем при тренуванні моделей для прогнозування збоїв, включає в себе рекомендаційну систему на етапі формування набору даних для аналізу. Застосування методів описової, контекстної статистики та кореляційного аналізу до даних показників сенсорів технологічних ліній дозволяє виявити:

- пропуски та викиди у даних, які в залежності від специфіки виробництва відповідним чином коригуються;
- аномалії показників сенсорів та кореляцію між ними, що дає змогу відслідковувати зміни корельованих аномалій із часом наближення збою;
- тривалість деградації показників: дозволяє відкоригувати спосіб відбору даних для подальшого аналізу з метою їх раннього виявлення;
- групи сенсорів із високим коефіцієнтом кореляції, поведінка яких є синхронною;
- режими роботи, на яких трапляються збої на основі аналізу графіків розподілу: їх врахування у прогнозній моделі дозволяє вчасно ліквідувати можливий збій.

Інформація про резолюцію сирих даних стала доступною для всіх сенсорів одночасно, з візуалізацією і можливістю порівняти різні групи сенсорів, визначивши, які значення прийнятні для всіх видів історичних даних. Результати аналітики по конкретному виробництву фахівці Data Science отримують досить швидко, що прискорює комунікацію з працівниками виробництва для швидкого вирішення питань стосовно врахування їх специфіки

Було прискорено та покращено вибір методу заповнення пропусків у даних сенсорів, виявлення та аналіз викидів і аномалій. Аналіз розподілів показників сенсорів дозволяє відкоригувати спосіб навчання прогнозної моделі. Дослідження якості системи аналізу показників показало у разі її застосування підвищення точності прогнозних моделей технічного обслуговування технологічних ліній на 10%.

ВИСНОВКИ

Проведене дослідження дозволяє зробити наступні висновки. Прогнозне технічне обслуговування технологічних ліній є важливою складовою надійної та ефективної роботи технологічних ліній важкої промисловості. Проведення якісної аналітики показників сенсорів дозволяє генерувати списки потенційних джерел проблем при тренуванні моделей для прогнозування збоїв та підвищує точність прогнозу поломок технологічних ліній.

Досліджено теоретичні засади прогнозного обслуговування технологічних ліній. Установлено, що технічне обслуговування технологічної лінії є важливим аспектом її надійної та ефективної роботи. Прогресивна стратегія обслуговування технологічних ліній базується на RCM-методології, яка включає іноваційні підходи до технічного обслуговування технологічних ліній: 1) прогнозне технічне обслуговування (Predictive Maintenance, PdM) – дозволяє визначити стан технологічної лінії, яка знаходиться в експлуатації та оцінити, коли слід провести технічне обслуговування на основі сукупних даних датчиків і тенденції для прогнозування майбутньої деградації та відмови обладнання; 2) прогнозне технічне обслуговування на основі стану (Condition-based Predictive Maintenance, CBM) – використовує дані, зібрані під час моніторингу, щоб виконати технічне обслуговування саме в той момент, коли це необхідно, і до того, як станеться критичний збій.

Здійснений аналіз існуючих рішень дозволив виявити проблеми технічного обслуговування у важкій промисловості. Моделі, які здійснюють прогнозне технічне обслуговування технологічних ліній, постійно вдосконалюються, їх розробкою та супроводом займаються фахівці відділів Data Science. Однак є проблема, обумовлена різними технічними причинами, що потребують корегування та оновлення параметрів моделей. З цією метою здійснюється прогнозне технічне обслуговування на основі стану, яке передбачає аналіз даних датчиків із метою виявлення проблемних даних та прогнозування можливих проблем у

обслуговуванні технологічних ліній. З урахуванням цієї аналітики відбувається оновлення моделей та забезпечується уникнення збоїв у їх роботі.

Проведений аналіз останніх наукових досліджень та публікацій дозволив виявити ключові проблеми у роботі з даними для прогнозного обслуговування: проблеми якості даних: шум і неконсистентність у сенсорних даних, пропущені або недостатні дані через проблеми з IoT-системами; проблеми кількості даних: нестача маркованих даних через рідкісні випадки відмов, дисбаланс у даних (домінування нормальної роботи над несправностями). Для покращення точності прогнозів і ефективності систем PdM дослідники пропонують використовувати сучасні підходи, такі як алгоритми заповнення пропущених даних, інженерія особливостей, стандартизація даних і навчання на основі малих вибірок для подолання зазначених викликів.

Здійснений аналіз дозволив обґрунтувати вибір методів та моделей декриптивної та предиктивної аналітики показників сенсорів технологічних ліній. Для виявлення проблем, які можуть впливати на якість прогнозних моделей, здійснюється аналітика «сирих» поточних даних і даних за минулі періоди, з використанням: описової статистики: здійснюється поглиблений аналіз із виведенням у табличному і графічному вигляді стандартних показників описової статистики та виявлення кількості пропущених, нульових, окремих, від'ємних значень сенсорів, коефіцієнтів ексцесу, варіації й асиметрії; специфічної контекстної статистики – здійснюється розрахунок із виведенням у табличному і графічному вигляді: часових проміжків між сусідніми даними у часовому ряді показників сенсорів та статистичних показників тривалості інтервалів; пропусків даних; завмерлих даних; кореляційного аналізу показників сенсорів.

Для створення інформаційної системи було обрано мову програмування Python, фреймворки Flask та Plotly Dash. Для роботи з таблицями, даними та візуалізації застосовано Python-бібліотеки Pandas, NumPy, Plotly. Реалізація статистичних методів здійснювалася з використанням Pandas-profiling. Python multiprocessing було застосовано для паралельної обробки великих обсягів даних на внутрішніх серверах.

Створення фронтенду інформаційної системи аналізу показників сенсорів за допомогою Dash дозволяє забезпечити інтерактивність і зручність роботи з аналітичними результатами. Для зберігання даних використано об'єктно-реляційну СУБД PostgreSQL та інтерактивний сервіс запитів Amazon Athena для аналізу даних у сховищі Amazon S3 за допомогою SQL. Python-бібліотека SQLAlchemy використана для спрощення роботи з реляційними базами даних.

Використання вищезазначеного стеку технологій дозволяє забезпечити надійну обробку, зберігання та візуалізацію даних. Інструменти Python, такі як Flask, Dash, Pandas та SQLAlchemy, забезпечують високу продуктивність системи, а хмарні сервіси Amazon S3 та Athena допомагають масштабувати обробку даних, роблячи систему придатною для роботи з великими обсягами даних та забезпечуючи ефективність її функціонування. Використання Plotly для візуалізації та Dash Bootstrap Components для структурування інтерфейсу робить систему гнучкою та інтуїтивно зрозумілою для користувачів, полегшуючи роботу з великими наборами даних і дозволяючи швидко отримувати візуальні аналітичні інсайти.

Здійснено моделювання, розробку та програмну реалізацію інформаційної системи аналізу показників сенсорів технологічних ліній. Розроблена система автоматизує аналіз даних сенсорів технологічних ліній, генерує списки потенційних джерел проблем при тренуванні моделей для прогнозування збоїв, включає в себе рекомендаційну систему на етапі формування набору даних для аналізу.

Інформація про резолюцію сирих даних стала доступною для всіх сенсорів одночасно, з візуалізацією і можливістю порівняти різні групи сенсорів, визначивши, які значення прийнятні для всіх видів історичних даних. Результати аналітики по конкретному виробництву фахівці Data Science отримують досить швидко, що прискорює комунікацію з працівниками виробництва для швидкого вирішення питань стосовно врахування їх специфіки

Було прискорено та покращено вибір методу заповнення пропусків у даних сенсорів, виявлення та аналіз викидів і аномалій. Аналіз розподілів показників сенсорів дозволяє відкоригувати спосіб навчання прогнозної моделі. Дослідження якості системи аналізу показників показало у разі її застосування підвищення точності прогнозних моделей технічного обслуговування технологічних ліній на 10%.

Поставлені завдання виконано повністю, однак швижкий розвиток технологій Industry 4.0 обумовлює необхідність регулярного вдосконалення функціоналу системи у відповідності з новими підходами у СВРdM.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Автоматизована система керування технологічними процесами: вебсайт. URL: https://bts.net.ua/ua/acy/oborudovanie/acy_tp/ (дата звернення 11.09.2024).
2. Jeschke S., Brecher C., Song H., Rawat D. Industrial Internet of Things. Cybermanufacturing Systems. Springer Cham, 2017. URL: <https://link.springer.com/book/10.1007/978-3-319-42559-7> (дата звернення 11.10.2024).
3. Lughofer E., Kepler J., Mouchaweh M. Predictive Maintenance in Dynamic Systems Advanced Methods, Decision Support Tools and Real-World Applications: Advanced Methods, Decision Support Tools and Real-World Applications. Springer, 2019. DOI:10.1007/978-3-030-05645-2.
4. What is Predictive Maintenance (PdM), and Why is it Important: вебсайт. URL: <https://facilio.com/blog/predictive-maintenance> (дата звернення 15.11.2024).
5. Sharma, J., Mittal, M.L. & Soni, G. Condition-based maintenance using machine learning and role of interpretability: a review. *Int J Syst Assur Eng Manag* 2024. № 15. P. 1345–1360. DOI: 10.1007/s13198-022-01843-7.
6. Horrell M., Reynolds L. and McElhinney A. Data Science in Heavy Industry and the Internet of Things. *Harvard Data Science Review*. 2020. Vol. 2, № 2. DOI: 10.1162/99608f92.834c6595.
7. IoT Predictive Maintenance: Components, Use Cases and Benefits: вебсайт. URL: <https://www.xyte.io/blog/iot-predictive-maintenance> (дата звернення 18.10.2024).
8. Oliynyk K. 5 Use Cases of Predictive Maintenance using IoT: вебсайт. URL: <https://webbylab.com/blog/use-cases-of-iot-based-predictive-maintenance/> (дата звернення 16.10.2024).
9. Lee J., Wu F., Zhao W., Ghaffari M., Liao L., Siegel D. Prognostics and health management design for rotary machinery systems—Reviews, methodology and

applications. *Mechanical Systems and Signal Processing*. 2014. Vol. 42, № 1-2. P. 314-334. DOI: 10.1016/j.ymssp.2013.06.004.

10. Manchadi O., Ezzahraa B., Bassma J. Predictive Maintenance in Healthcare System: A Survey. *IEEE ACCESS*. 2023. Vol. 11. P. 61313-61330. DOI: 10.1109/ACCESS.2023.3287490.

11. Lei Y., Li N., Guo L., Li N., Yan T., Lin J. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing*. 2018. № 104. P. 799–834. DOI: 10.1016/j.ymssp.2017.11.016.

12. Es-sakali N., Cherkaoui M., Mghazli M., Naimi Z. Review of predictive maintenance algorithms applied to HVAC systems. *Energy Reports*. 2022. Vol. 8, № 9. P. 1003-1012. DOI: 10.1016/j.egyр.2022.07.130.

13. Темчур В.С., Баган Т.Г. Методи глибокого навчання моделей для прогнозного обслуговування. *Вчені записки ТНУ ім. В.І. Вернадського. Серія: технічні науки*. 2023. Т. 34(73) № 6. С. 155-162. DOI: 10.32782/2663-5941/2023.6/23.

14. Ucar A., Karakose M., Kırımça N. Artificial Intelligence for Predictive Maintenance Applications: Key Components, Trustworthiness, and Future Trends. *Applied Sciences*. 2024. Vol. 14, № 2. P. 898. DOI: 10.3390/app14020898.

15. Achouch M., Dimitrova M., Ziane K., Sattarpanah Karganroudi S., Dhouib R., Ibrahim H., Adda M. On Predictive Maintenance in Industry 4.0: Overview, Models, and Challenges. *Applied Sciences*. 2022. Vol. 12, № 16. P. 8081. DOI: 10.3390/app12168081.

16. Patel M, Vasa J., Patel B. Predictive Maintenance: A Comprehensive Analysis and Future Outlook. Published in *2nd International Conference on Futuristic Technologies (INCOFT)*, 2024. DOI: 10.1109/INCOFT60753.2023.10425122.

17. Serradilla O., Zugasti E., Zurutuza U. Deep Learning Models for Predictive Maintenance: A Survey. 2020. DOI: 10.48550/arXiv.2010.03207.

18. Keleko A. T., Kamsu-Foguem B., Ngouna R. H. and Tongne A. Artificial Intelligence and Real-Time Predictive Maintenance in Industry 4.0: a bibliometric analysis. *AI and Ethic*. 2022. Vol 2. P. 533-577. DOI: 10.1007/s43681-021-00132-6.

19. Shen K., Ong H., Niyato D. and Yuen C. Predictive Maintenance for Edge-Based Sensor Networks: вебсайт. URL: <https://arxiv.org/pdf/2007.03313> (дата звернення 20.10.2024).
20. Roemer M., Byington C., Kasprzyński G., Vachtsevanos G. An overview of selected prognostic technologies with application to engine health management. Proceedings of the ASME Turbo Expo 2006: Power for Land, Sea, and Air. Vol. 2: Aircraft Engine; Ceramics; Coal, Biomass and Alternative Fuels; Controls, Diagnostics and Instrumentation; Environmental and Regulatory Affairs. 2006. P. 707–715. DOI: 10.1115/GT2006-90677.
21. Rudin C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*. 2019. Vol. 1, № 5. P. 206–215. DOI: 10.1038/s42256-019-0048-x.
22. Blann, D. Maximizing the P-F interval through condition-based maintenance. Maintworld: вебсайт. URL: <https://www.maintworld.com/Applications/Maximizing-the-P-F-Interval-Through-Condition-Based-Maintenance> (дата звернення 21.10.2024).
23. Sikorska J., Hodkiewicz M., Ma L. Prognostic modelling options for remaining useful life estimation by industry. *Mechanical Systems and Signal Processing*. 2011. Vol. 25, № 5. P. 1803–1836. DOI: 10.1016/j.ymssp.2010.11.018.
24. Wheeler K. R., Kurtoglu T., Poll S. D. A survey of health management user objectives related to diagnostic and prognostic metrics. ASME 2009 *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. 2009. P. 1287–1298. DOI: 10.1115/DETC2009-87073.
25. Ambler G., Seaman S., Omar R. Z. An evaluation of penalised survival methods for developing prognostic models with rare events. *Statistics in medicine*. 2012. Vol. 31, № 11–12. P. 1150–1161. DOI: 10.1002/sim.4371.
26. Buyske S., Fagerstrom R., Ying Z. A class of weighted log-rank tests for survival data when the event is rare. *Journal of the American Statistical Association*. 2020. Vol. 95, № 449. P. 249–258. DOI: 10.1080/01621459.2000.10473918.

27. Williams B., Tokar T. A review of inventory management research in major logistics journals: Themes and future directions. *The International Journal of Logistics Management*. 2008. Vol. 19, № 2. P. 212–232. DOI: 10.1108/09574090810895960.
28. Burman P., Chow E., Nolan D. A Cross-validatory method for dependent data. *Biometrika*. 1994. Vol. 81, № 2. P. 351–358. DOI: 10.2307/2336965.
29. Draper D. 20 Bayesian model specification: Heuristics and examples. In P. Damien, P. Dellaportas, N. G. Polson, D. A. Stephens. *Bayesian theory and applications*. Oxford Academic, 2013. P. 409–431. DOI: 10.1093/acprof:oso/9780199695607.003.0020.
30. Chang K., Yoo Y., Baek J-G. Anomaly Detection Using Signal Segmentation and One-Class Classification in Diffusion Process of Semiconductor Manufacturing. *Sensors*. 2021. Vol. 21. № 11. P. 3880. DOI: 10.3390/s21113880.
31. Barbariol T., Feltresi E., Susto GA. Machine learning approaches for anomaly detection in multiphase flow meters. *IFAC-PapersOnLine*. 2019. Vol. 52. № 11. P. 212–217. DOI: 10.1016/j.ifacol.2019.09.143/
32. Python Documentation: вебсайт. URL: <https://www.python.org/doc/versions/> (дата звернення 26.10.2024).
33. Pandas documentation: вебсайт. URL: <https://pandas.pydata.org/docs/> (дата звернення 28.10.2024).
34. NumPy documentation: вебсайт. URL: <https://numpy.org/doc/2.1/> (дата звернення 28.10.2024).
35. Plotly Open Source Graphing Library for Python: вебсайт. URL: <https://plotly.com/python> (дата звернення 29.10.2024).
36. Python pandas-profiling Documentation: вебсайт. URL: <https://pypi.org/project/pandas-profiling> (дата звернення 30.10.2024).
37. Python multiprocessing Documentation: вебсайт. URL: <https://docs.python.org/3/library/multiprocessing.html> (дата звернення 30.10.2024).
38. Flask documentation: вебсайт. URL: <https://flask.palletsprojects.com/en/stable/> (дата звернення 1.11.2024).

39. Dash Python User Guide: вебсайт. URL: <https://dash.plotly.com/> (дата звернення 2.11.2024).

40. PostgreSQL documentation: вебсайт. URL: <https://www.postgresql.org/docs/> (дата звернення 2.11.2024).

41. Amazon Athena Documentation: вебсайт. URL: <https://docs.aws.amazon.com/athena/> (дата звернення 3.11.2024).

42. Amazon Simple Storage Service Documentation: вебсайт. URL: <https://docs.aws.amazon.com/s3/> (дата звернення 4.11.2024).

43. SQLAlchemy 2.0 Documentation: вебсайт. URL: <https://docs.sqlalchemy.org/en/20/> (дата звернення 5.11.2024).

ДОДАТОК А

Текстовий блок із результатами проведеного аналізу

Descriptive machine summary

CRS1_BPC2410 summary

Sensors with low number of unique values:
 32 binary sensor(s) with 0 / 1 values:

Sensor	0	1	% of 0	% of 1
...	55	49	52.88	47.12
...	66	61	51.97	48.03
...	77	72	51.68	48.32
...	114	106	51.82	48.18
...	10558	27484	27.75	72.25
...	31	32	49.21	50.79
...	3	6	33.33	66.67
...	14	10	58.33	41.67
...	6	2	75	25
...	5	1	83.33	16.67

Median percent of zero values for these sensors: 100.0.

Sensors with missing values:
 Not found.

Sensors with negative values:
 20 sensor(s) with negative values:

Median percent of negative values for these sensors: 0.33.

Sensors with high coefficient of variation:
 30 sensor(s) with CV > 1:

Median CV value for these sensors: 1.85.

Sensors with higher chances of outliers (based on kurtosis values):
 27 sensor(s) with kurtosis > 5:

Median kurtosis value for these sensors: 207.42.

Freezes:
 167 sensor(s) with freezes:

Median number of freezes per sensor is 2.
 Median freeze duration per sensor is 142.2 hours.

2 binary sensor(s) with not 0 / 1 values:

Sensor	Value 1	Value 1 (count)	Value 1 (%)	Value 2	Value 2 (count)	Value 2 (%)
...	1	10	58.82	5	7	41.18
...	0	7	77.78	8	2	22.22

50 constant sensor(s) with only '0' values:

1 constant sensor(s) with only '1' values:

Рисунок А.1 – Текстовий блок «Підсумки» (частина 1)

Кафедра інтелектуальних інформаційних систем
 Інформаційна система аналізу показників сенсорів технологічних ліній

7 constant sensor(s) with not '0' or '1' values:

⊕	Sensor	⊕	Value	⊕	Count
	...		363		4
	...		407		4
	...		2134		4
	...		2115		4
	...		2200		4
	...		700		4
	...		40		4

20 sensor(s) with more than 2 and less than 10 unique values:

Median percent of unique values for these sensors: 6.82.

Sensors with zero values:
 98 sensor(s) with more than 10% and less than 100% of zero values:

Categorical sensors:
 Not found.

Correlation groups:
 28 correlation group(s) found.
 Number of correlated sensors in each group varies from 1 to 15.

Time interval:
 143 sensor(s) with not constant time interval:

There are up to 95 different interval values per sensor.

Gaps:
 183 sensor(s) with gaps:

Median number of gaps per sensor is 3.
 Median gap duration per sensor is 75.76 hours.

Рисунок А.2 – Текстовий блок «Підсумки» (частина 2)