

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
(повна назва)

Кафедра Штучного інтелекту  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

рівень вищої освіти другий (магістерський)

Дослідження методів машинного навчання для технічного  
обслуговування в Індустрії 4.0  
(тема)

Виконав:  
здобувач другого року навчання,  
групи СШМ-23-1

Єгор Гордієнко  
(власне ім'я, прізвище)

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

Тип програми освітньо-наукова  
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту  
(повна назва освітньої програми)

Керівник проф. Ваган Терзіян  
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ \_\_\_\_\_  
(підпис)

Олег ЗОЛОТУХІН  
(власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ Комп'ютерних наук \_\_\_\_\_

Кафедра \_\_\_\_\_ Штучного інтелекту \_\_\_\_\_

Рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_

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

Тип програми \_\_\_\_\_ освітньо-наукова \_\_\_\_\_  
(освітньо-професійна або освітньо-наукова)

Освітня програма \_\_\_\_\_ Системи штучного інтелекту \_\_\_\_\_  
(повна назва)

ЗАТВЕРДЖУЮ:  
Зав. кафедри \_\_\_\_\_  
(підпис)  
«\_\_\_\_\_» \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві \_\_\_\_\_ Гордієнку Єгору Дмитровичу \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи \_\_\_\_\_ Дослідження методів машинного навчання для технічного обслуговування в Індустрії 4.0 \_\_\_\_\_

затверджена наказом університету від 21 квітня 2025 р. № 295Ст

2. Термін подання студентом роботи до екзаменаційної комісії 10 червня 2025 р.

3. Вихідні дані до роботи \_\_\_\_\_ Дослідження можливостей наявних програмних середовищ розробки всіх етапів реалізації програмного комплексу, програмна реалізація верхнього рівня у SCADA-системі, програмна реалізація нижнього рівня та комунікаційного каналу зв'язку, алгоритм обробки даних \_\_\_\_\_

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1) Аналіз предметної галузі \_\_\_\_\_


2) Аналіз технічних засобів та методів \_\_\_\_\_

3) Опис програмної реалізації \_\_\_\_\_

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	21.04.2025	виконано
2	Аналіз предметної галузі	22.04.2025–24.04.2025	виконано
3	Постановка задачі	24.04.2025–27.04.2025	виконано
4	Дослідження методів реалізації	27.04.2025–30.04.2025	виконано
5	Проектування структури комплексу	01.05.2025–04.05.2025	виконано
6	Розробка нижнього рівня реалізації	05.05.2025–09.05.2025	виконано
7	Розробка верхнього рівня реалізації	09.05.2025–15.05.2025	виконано
8	Налагодження каналу зв'язку	15.05.2025–19.05.2025	виконано
9	Створення бази даних	19.05.2025–22.05.2025	виконано
10	Впровадження обраного алгоритму	22.05.2025–27.05.2025	виконано
11	Структуризація програмного комплексу	27.05.2025–02.06.2025	виконано
12	Оформлення пояснювальної записки	02.06.2025–04.06.2025	виконано
13	Надання роботи на відгук та рецензування	05.06.2025	виконано
14	Перевірка пояснювальної записки на антиплагіат	06.06.2025	виконано
15	Попередній захист роботи з керівником	09.06.2025	виконано
16	Захист перед ЕК	10.06.2025	виконано

Дата видачі завдання 21 квітня 2025 р.

Здобувач   
(підпис)

Керівник роботи \_\_\_\_\_ проф. Ваган Терзіян  
(підпис) (посада, власне ім'я, прізвище)

## РЕФЕРАТ

Пояснювальна записка: 95 с., 38 рис., 6 табл., 2 дод., 15 джерел.

ВЕРХНІЙ РІВЕНЬ, КЛІЄНТ-СЕРВЕРНИЙ ПІДХІД, МАШИННЕ НАВЧАННЯ, НАВЧАННЯ З УЧИТЕЛЕМ, НИЖНІЙ РІВЕНЬ, ПРОМИСЛОВЕ ВИРОБНИЦТВО, ТЕХНІЧНЕ ОБСЛУГОВУВАННЯ, ШТУЧНИЙ ІНТЕЛЕКТ.

Об'єкт дослідження – впровадження елементів машинного навчання у технічне обслуговування обладнання на промисловому виробництві.

Предметом дослідження у кваліфікаційній роботі є програмний комплекс для роботи обладнання з елементами машинного навчання.

Мета роботи – створення алгоритмів машинного навчання для коректного процесу проходження технічного обслуговування, на основі даних, що надходять з польових пристроїв.

Методи дослідження – ґрунтовне порівняння та аналіз існуючих методів машинного навчання, дослідження можливостей впровадження алгоритмів машинного навчання.

## **ABSTRACT**

Master's thesis contains: 95 pp., 38 fig., 6 tabl., 2 ann., 15 references.

ARTIFICIAL INTELLIGENCE, CLIENT-SERVER APPROACH, INDUSTRIAL PRODUCTION, LOW LEVEL, MACHINE LEARNING, SUPERVISED LEARNING, TECHNICAL MAINTENANCE, TOP LEVEL.

The object of the research is the implementation of machine learning elements in the maintenance of equipment in industrial production.

The subject of the research in the qualification work is a software package for the operation of equipment with machine learning elements.

The purpose of the work is the creation of machine learning algorithms for the correct process of maintenance, based on data received from field devices.

Research methods are a thorough comparison and analysis of existing machine learning methods, research into the possibilities of implementing machine learning algorithms.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів .....	7
Вступ.....	8
1 Аналіз предметної галузі .....	10
1.1 Технічне обслуговування, як об'єкт автоматизації .....	10
1.2 Архітектура систем технічного обслуговування в Індустрії 4.0.....	11
1.3 Роль машинного навчання у прогностному обслуговуванні.....	13
1.4 Методи машинного навчання .....	18
1.5 Найбільш уживані алгоритми машинного навчання на практиці.....	21
1.6 Глибинне навчання, як розвиток машинного навчання.....	28
1.7 Аргументація використання машинного навчання .....	29
1.8 Переваги та виклики впровадження ML у промисловості .....	32
1.9 Огляд сучасних ML-підходів у технічному обслуговуванні.....	34
1.10 Перспективи розвитку .....	36
2 Аналіз технічних засобів та методів .....	39
2.1 Опис Freelance Engineering від ABB .....	39
2.2 Встановлення та налаштування Freelance Engineering.....	44
2.3 Опис використання програми UaExpert.....	51
2.4 Опис Ignition від Inductive Automation.....	53
2.5 Встановлення та налаштування Ignition .....	55
2.6 Опис PostgreSQL .....	64
3 Опис програмної реалізації .....	66
3.1 Розробка нижнього рівня.....	66
3.2 Розробка верхнього рівня.....	72
3.3 Впровадження алгоритмів машинного навчання .....	76
Висновки .....	79
Перелік джерел посилання .....	81
Додаток А Програмна реалізація алгоритмів .....	83
Додаток Б Відомість кваліфікаційної роботи.....	95

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

БД – база даних;

ВР – верхній рівень;

НР – нижній рівень;

ПЗ – програмне забезпечення;

ТО – технічне обслуговування;

AI – Artificial Intelligence – штучний інтелект;

DL – Deep Learning – глибинне навчання;

HMI – Human Machine Interface – інженерні рішення, що забезпечують взаємодію оператора з керованими ним машинами;

Industry 4.0 – четверта промислова революція;

ML – Machine Learning – машинне навчання;

OPC – Open Platform Communications – програмна технологія, що надає єдиний інтерфейс для управління об'єктами автоматизації і технологічними процесами;

SCADA – Supervisory Control And Data Acquisition – програмний пакет, призначений для забезпечення роботи в реальному часі систем збору, обробки, відображення та архівування інформації про об'єкт автоматизації;

SQL – Structured Query Language – мова запиту, що використовується реляційними базами даних.

## ВСТУП

Автоматизація є ключовою складовою сучасного ринку та вже давно сформувалася як самостійний і стратегічно важливий напрям, що суттєво впливає на зростання продуктивності праці та ефективність виробничих процесів. З моменту, коли автоматизація перестала бути лише підгалуззю електромеханіки, її розвиток набув цифрового виміру. Особливого значення додавала необхідність передачі даних для реалізації алгоритмів управління технологічними процесами.

Першим етапом розвитку електронних систем керування стало створення централізованих систем, в яких один керуючий вузол відповідав за обробку сигналів від усіх польових пристроїв формував команди для всіх виконавчих механізмів. Однак зі збільшенням складності виробничих систем і кількості завдань управління різко зріс обсяг даних, що потребували обробки, а також ускладнилися алгоритми прийняття рішень. У результаті централізована модель управління втратила ефективність через невідповідність між складністю керованого об'єкта та обмеженими можливостями окремого керуючого пристрою.

У сучасних умовах ключовим фактором розвитку промисловості є здатність оперативно отримувати, аналізувати, зберігати та обробляти великі обсяги інформації. Застосування технологій машинного навчання для автоматизації цих процесів стало основою реалізації концепції Індустрії 4.0. Ця концепція передбачає глибоку цифровізацію виробництва, де всі процеси контролюються та керуються в реальному часі, з урахуванням динамічних змін у внутрішньому та зовнішньому середовищі.

Визначальними характеристиками Індустрії 4.0 є повністю автоматизовані виробництва, використання розумних сенсорних систем, обчислення на периферії, цифрові двійники та штучний інтелект. Одним із найважливіших завдань при впровадженні цієї концепції є забезпечення надійності критично важливих виробничих і технологічних процесів,

зокрема, через впровадження інтелектуальних систем технічного обслуговування.

У межах цієї кваліфікаційної роботи передбачається створення алгоритмів машинного навчання для моніторингу технічного стану обладнання у промисловому середовищі. У ході дослідження розглянуто сучасні підходи до побудови систем прогнозування на виробництві, а також фактори, що впливають на їх ефективність. Особливу увагу приділено демонстрації переваг застосування машинного навчання в аналітиці та прогнозуванні технічного обслуговування, що дозволяє знижувати витрати, запобігати відмовам та підвищувати загальну ефективність виробничих систем.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

### 1.1 Технічне обслуговування, як об'єкт автоматизації

На теперішній час машинне навчання є невід'ємною складовою сучасних систем технічного обслуговування обладнання у промисловості. Його активне застосування дозволяє здійснювати прогнозування стану технологічних процесів, а також моніторинг критичних параметрів у режимі реального часу. Незапланована зупинка виробництва через технічну несправність або аномальну поведінку обладнання може призвести до суттєвих фінансових втрат, зривів логістичних ланцюгів або навіть до аварій, що загрожують безпеці персоналу та навколишньому середовищу.

Сучасні алгоритми обробки даних, у тому числі методи машинного навчання, надають ефективні інструменти для діагностики технічного стану та прогнозного технічного обслуговування. Завдяки використанню статистичного аналізу, нейронних мереж, бустінгу, класифікаторів та інших методів, можливо своєчасно виявляти ознаки деградації або аномальної поведінки агрегатів задовго до настання фактичного збою.

Особливо важливу роль у розвитку інтелектуальних систем обслуговування відіграє поширення технологій «великих даних» (Big Data). Завдяки здешевленню польових пристроїв, збільшенню обчислювальної потужності та доступності систем зберігання, з'явилась можливість збирати та аналізувати великі обсяги телеметричних даних з виробничого обладнання.

Застосування Big Data у сфері технічного обслуговування дозволяє вирішувати низку критичних завдань:

- прогнозування залишкового ресурсу вузлів та агрегатів;
- виявлення та класифікація аномалій у роботі обладнання на ранніх етапах;
- оптимізація графіків профілактичних перевірок і ремонтів;

- зниження простоїв завдяки запобіганню аварійним відмовам;
- оцінка економічної ефективності технічного обслуговування та планування витрат.

Це зумовлює перехід від реактивного або планового обслуговування до адаптивного та орієнтованого на отримані дані, що відображають реальний стан обладнання з урахуванням умов експлуатації. У результаті підприємства отримують змогу значно підвищити надійність, безпечність і рентабельність виробничих систем.

## 1.2 Архітектура систем технічного обслуговування в Індустрії 4.0

У межах концепції Індустрії 4.0 технічне обслуговування трансформується із пасивного допоміжного процесу на стратегічно важливу функцію, що тісно інтегрується в загальну цифрову архітектуру підприємства. Системи технічного обслуговування нового покоління базуються на принципах гнучкості, самоадаптації, прогнозування і взаємодії між машинами, що реалізується через кіберфізичні системи та Інтернет речей.

Архітектура сучасної системи технічного обслуговування має низку логічних рівнів:

- фізичний рівень (польові пристрої). Цей рівень включає усі апаратні компоненти, які забезпечують безперервне зчитування параметрів обладнання: вбудовані сенсори (температура, вібрація, струм, тиск); пристрої збору даних (датчики, контролери, PLC); системи ідентифікації (RFID, QR-коди для технічного обліку). Завдяки цим пристроям відбувається первинна генерація даних для подальшого аналізу;
- комунікаційний (мережевий) рівень. Дані з фізичних пристроїв передаються по промисловим протоколам, таких як: OPC UA; MQTT; Modbus TCP/RTU; EtherCAT, Profinet. Цей рівень відповідає за надійність і швидкість передачі інформації до системи обробки та зберігання;

– аналітичний рівень (рівень обробки даних). На цьому рівні реалізуються: збір і зберігання даних у базах (наприклад, SQL, NoSQL, TimeSeries DB); обробка даних у реальному часі; побудова моделей машинного навчання для прогнозу технічного стану; виявлення аномалій, класифікація станів, регресійний аналіз. Центральними компонентами є платформи аналітики (наприклад, Python/ML-сервери, Apache Kafka, Spark, Edge-обчислення);

– рівень візуалізації та взаємодії з користувачем (HMI/SCADA). На цьому рівні здійснюється подання результатів аналізу: графіки стану обладнання; сигнали тривоги та повідомлення; прогнози залишкового ресурсу; інтерфейси керування ремонтами та технічними заявками. Технічний персонал має змогу швидко реагувати на зміну стану або поведінки технологічного процесу та приймати відповідні рішення;

– рівень керування та інтеграції (MES/ERP). Включає системи: керування виробництвом (MES); планування ресурсів підприємства (ERP); системи управління обслуговуванням (CMMS). Цей рівень забезпечує стратегічне планування ремонтів, закупівель, обліку запасних частин, фінансову оцінку ефективності технічного обслуговування.

Для повноцінної реалізації системи, що виконує технічне обслуговування, її архітектура повинна відповідати наступними ключовим принципам:

– модульність – система може бути масштабована в залежності від складності підприємства;

– інтероперабельність – підтримка стандартів (OPC UA, ISA-95, RAMI 4.0) для взаємодії між компонентами;

– висока доступність – дані доступні для обробки цілодобово та в режимі реального часу;

– кібербезпека – шифрування, контроль доступу, аудит дій користувачів;

– інтелектуальність – впровадження алгоритмів машинного навчання

для автономного прийняття рішень.

На практиці алгоритм системи може мати наступний вигляд: на польовому пристрої фіксується підвищена температура двигуна, дані передаються через OPC UA на центральний аналітичний сервер, модель машинного навчання ідентифікує ймовірність виходу з ладу протягом 48 годин, система SCADA подає попередження оператору, MES-система формує заявку на ремонт, ERP-платформа резервує запасні частини.

Архітектура сучасних систем технічного обслуговування в Індустрії 4.0 орієнтована на автономність, передбачення і цифрову трансформацію процесів. Її впровадження забезпечує не лише зниження витрат на ремонт та обслуговування, а й підвищення безпеки, надійності та гнучкості промислових систем у динамічному виробничому середовищі.

### 1.3 Роль машинного навчання у прогностному обслуговуванні

У межах концепції Індустрії 4.0 прогностне технічне обслуговування (Predictive Maintenance, PdM) стало одним з ключових напрямів цифрової трансформації виробництва. Метою PdM є своєчасне виявлення ознак зношення або наближення до відмови обладнання на основі фактичних експлуатаційних даних. Застосування машинного навчання є визначальним чинником ефективності таких систем, оскільки саме воно забезпечує інтелектуальну обробку складних і багатовимірних потоків даних у реальному часі.

Прогностне технічне обслуговування покладає на машинне навчання завдання наступного характеру:

- прогноз залишкового ресурсу (Remaining Useful Life, RUL) – оцінювання часу, що залишився до відмови, на основі історичних і поточних технічних параметрів;

- виявлення аномалій – ідентифікація нетипової поведінки обладнання, що може свідчити про потенційні збої;

- класифікація стану – автоматичне визначення технічного стану вузлів (наприклад: нормальний, зношений, критичний);

- планування технічного обслуговування – оптимізація графіків на основі прогнозних моделей для мінімізації простоїв і витрат.

Різні класи алгоритмів машинного навчання застосовуються для вирішення специфічних задач прогнозного обслуговування:

- класифікатори дозволяють визначити категорію технічного стану;
- регресійні моделі забезпечують чисельну оцінку RUL;
- алгоритми кластеризації використовуються для виявлення аномалій та групування схожих режимів роботи;

- нейронні мережі та глибинне навчання особливо ефективні при роботі з часовими рядами та складними залежностями між параметрами.

Машинне навчання є основою побудови ефективних, адаптивних і розумних систем прогнозного технічного обслуговування. Воно забезпечує перехід до активних стратегій обслуговування, що дозволяє підприємствам не тільки зменшити витрати, а й підвищити надійність та стабільність виробничих процесів у рамках концепції Індустрії 4.0.

Ефективність систем прогнозного технічного обслуговування значною мірою залежить від якості, повноти та доступності вхідних даних. У контексті Індустрії 4.0 підприємства оснащуються великою кількістю польових пристроїв, які генерують безперервні потоки інформації про стан обладнання та параметри технологічного процесу. Збір, зберігання та обробка цих даних є фундаментом для побудови моделей машинного навчання, які використовуються для прогнозного технічного обслуговування.

Джерелами даних для навчання моделей машинного навчання у прогнозованому обслуговуванні можуть бути такі типи даних:

- сенсорні дані з польових пристроїв такі, як температура, тиск, вібрація, рівень шуму, струм, напруга тощо. Вони зчитуються у реальному часі та передаються через промислові протоколи;

- логи та повідомлення від обладнання – системні повідомлення контролерів, журнал помилок, аварійні звіти, події запуску/зупинки, позаштатні режими;

- журнали технічного обслуговування – історія ремонтів, заміні, профілактик або дані з комп'ютеризованої системи управління технічним обслуговуванням (Computerized Maintenance Management System, CMMS);

- експлуатаційна документація – специфікації, рекомендації виробника, посібники оператора;

- фінансово-економічна інформація – витрати на обслуговування, час простоїв, втрати продуктивності.

Джерела можуть передавати дані у різному вигляді, наприклад: структурованому (таблиці, SQL/NoSQL бази); напівструктурованому (JSON, XML, лог-файли); неструктурованому (звукові сигнали, зображення, відео). У виробничих системах частіше використовуються структуровані або напівструктуровані дані з мітками часу.

Перед тим як дані можуть бути використані в алгоритмах машинного навчання, вони повинні пройти етап попередньої обробки, що включає:

- очищення – видалення дублікатів, заповнення пропущених значень, фільтрація шуму;

- нормалізацію та стандартизацію – приведення параметрів до спільної шкали;

- синхронізацію часових рядів – вирівнювання даних з різною частотою вимірювання;

- розмітку – встановлення міток відмов або станів для навчання моделей;

- згортання вікон – побудова послідовностей даних заданої довжини для навчання моделей типу LSTM або CNN;

- методи виділення ознак.

Особливо важливою частиною підготовки даних є виділення ознак,

яке дає змогу зменшити розмірність вхідних даних та забезпечити кращу продуктивність моделей. Серед найбільш поширених технік: статистичні характеристики (середнє, дисперсія, пік-фактор); частотні характеристики (перетворення Фур'є, спектральний аналіз); ковзні середні та експоненціальні згладжування; побудова похідних та індексів (темپ зростання, градієнт зміни).

Якість даних – критичний фактор успішності моделей. Погана якість даних може призвести до: помилкових прогнозів; «навчання» на шумових або спотворених вибірках; втрати довіри до системи обслуговування.

Для цього впроваджуються системи автоматичного моніторингу якості даних, контроль цілісності, періодичні аудити та аномалія-детектори на рівні джерел.

Збір даних починається на рівні фізичних пристроїв, які взаємодіють із технологічним середовищем. До них належать:

- інтелектуальні сенсори (температури, вібрації, тиску, струму тощо), які можуть мати вбудовану логіку попередньої обробки;
- програмовані логічні контролери (PLC) – керують роботою обладнання та накопичують значення параметрів;
- RTU (Remote Terminal Units) – пристрої збору даних у розподілених мережах;
- DAQ-системи (Data Acquisition) – спеціалізовані платформи для багатоканального збору аналогових сигналів.

Ці пристрої здійснюють перетворення аналогових сигналів у цифрову форму, синхронізацію часових міток і передачу інформації до серверів або хмарних платформ.

Для забезпечення надійної та безпечної комунікації в промисловому середовищі застосовуються спеціалізовані протоколи:

- OPC UA (Open Platform Communications Unified Architecture) – один із найбільш універсальних стандартів для обміну даними між пристроями та ІТ-системами; підтримує моделювання об'єктів, безпеку, маршрутизацію;

- Modbus TCP/RTU – простий, широко поширений протокол для зчитування даних з пристроїв у реальному часі;

- MQTT (Message Queuing Telemetry Transport) – легкий протокол, орієнтований на IoT-застосування, використовується для передачі даних у хмару або до аналітичного сервісу;

- PROFINET, EtherCAT, CANopen – протоколи реального часу, характерні для систем автоматизації з високими вимогами до затримок.

Вибір протоколу залежить від вимог до швидкості, масштабованості, надійності та підтримки існуючого обладнання.

Передача даних може здійснюватися у кількох конфігураціях:

- Peer-to-peer – обмін між двома пристроями;

- Client-server – типова модель OPC UA;

- Publisher-subscriber – модель, яка реалізується у MQTT, де дані транслюються на багато одержувачів;

- Edge-to-cloud – гібридна архітектура, де попередня обробка даних здійснюється на рівні Edge (периферії), а аналітика – у хмарі.

Для забезпечення стабільної роботи системи важливим є тимчасове зберігання даних на рівні пристроїв або шлюзів:

- Edge computing – попередня обробка безпосередньо на місці збору даних для зменшення обсягу трафіку;

- Gateways – пристрої, що виконують роль буфера між сенсорами і сервером/хмарою;

- Time Series Databases (TSDB) – спеціалізовані бази (наприклад, InfluxDB, TimescaleDB) для зберігання даних з часовими мітками.

У рамках цифровізації все частіше використовуються хмарні рішення для аналітики, візуалізації та зберігання (наприклад, Amazon AWS IoT, Microsoft Azure IoT Hub, Google Cloud IoT тощо). Хмарні платформи також підтримують інтеграцію з ERP/MES/CMMS для передачі сигналів про технічний стан у бізнес-рівень.

Промислові дані часто мають критичний характер, тому передача

повинна відповідати критеріям шифрування (TLS, VPN), мати процес аутентифікації (сертифікати, ключі доступу) та контрольованого доступу (рольова модель користувачів) та бути тією, що відстежується (аудит, журнали подій).

Впровадження машинного навчання у прогнозне обслуговування у стратегічному баченні є частиною цифрової трансформації підприємств.

#### 1.4 Методи машинного навчання

Методи машинного навчання та статистичного аналізу, що використовуються для прогнозування, виявлення та інтерпретації аномальних станів:

- кластеризація та візуалізація даних;
- виявлення багатовимірних (multivariate) аномалій;
- використання відстані Махаланобіса;
- багатовимірний статистичний аналіз.

У багатьох прикладних задачах ефективнішою стратегією є вимірювання ступеня схожості між об'єктами, ніж побудова складної ознакової моделі. Наприклад, у задачах біоінформатики простіше порівнювати послідовності ДНК напряму, ніж переводити їх у вектори ознак. Аналогічна ситуація спостерігається у роботі з текстами, часовими рядами або зображеннями. Метод кластеризації дозволяє об'єднувати об'єкти у групи (кластери) на основі їхньої подібності, не потребуючи попереднього маркування – що є надзвичайно важливим у задачах моніторингу, де нормальні та аномальні режими не завжди чітко визначені.

Візуалізація даних слугує первинним інструментом для виявлення аномалій, однак при зростанні розмірності даних (велика кількість змінних) її ефективність знижується. У таких випадках застосовуються методи багатовимірної статистики. Зокрема, ймовірнісні підходи дозволяють визначати, наскільки нове спостереження відповідає розподілу нормальних

даних. Якщо імовірність  $p(x)$  для нової точки  $x$  нижча за поріг  $r$ , то така точка розглядається як аномальна. Це дозволяє ефективно виявляти відхилення в роботі обладнання, яке наближається до критичного стану чи відмови.

Відстань Махаланобіса – це метрика, що враховує кореляції між змінними та масштаби розкиду даних у різних напрямках простору. Її застосування передбачає:

- визначення центроїда (центра маси) нормального розподілу даних;
- оцінку коваріаційної матриці вибірки для врахування форми та орієнтації розподілу;
- розрахунок відстані до центроїда з урахуванням коваріації: чим далі точка знаходиться, тим меншою є ймовірність її належності до нормального стану.

Ця відстань особливо корисна у випадках, коли розподіл даних має еліпсоїдальну форму, а не сферичну. У таких випадках класичні евклідові метрики можуть давати хибні результати. У задачах класифікації, пов'язаних з розмежуванням між «нормальними» та «аномальними» станами обладнання, відстань Махаланобіса дозволяє з високою точністю визначити ймовірність належності нового спостереження до нормального класу.

Багатовимірний статистичний аналіз полягає в обробці високовимірних даних, яка потребує зменшення розмірності. Аналіз основного компонента – є основним методом зменшення кількості змінних, принцип якого полягає у виконанні лінійного відображення даних у просторі нижнього розміру при максимальній дисперсії даних. Створюється коваріаційна матриця даних та обчислюються її власні вектори, використання яких потрібно для реконструкції великої частки дисперсії вихідних даних. Простір функцій зменшується від початкового до охопленого кількома власними векторами при можливій втраті даних.

Принципом роботи машинного навчання є використання аналітичних алгоритмів обробки даних, на отриманих результатах яких приймаються поведінкові рішення залежні від передбачених наслідків. Алгоритми машинного навчання самостійно виконують поставлені задачі на великій кількості даних.

Машинне навчання охоплює низку завдань, що застосовуються у сфері моніторингу та прогнозного обслуговування:

- класифікація – визначення категорії стану обладнання (нормальний/аномальний);
- кластеризація – групування режимів роботи без попереднього маркування;
- регресія – прогнозування значення технічного параметра (наприклад, температура, залишковий ресурс);
- виявлення аномалій – знаходження спостережень, що відхиляються від типових сценаріїв;
- зменшення розмірності – спрощення ознакового простору для візуалізації або прискорення обчислень.

Усі задачі, які вирішуються за допомогою методів машинного навчання, зазвичай поділяються на дві основні категорії: навчання з учителем та навчання без учителя.

Навчання з учителем передбачає наявність заздалегідь підготовленого набору даних, де кожне спостереження супроводжується відповідною «правильною» відповіддю (міткою або цільовим значенням). На основі таких мічених даних модель навчається встановлювати залежність між вхідними параметрами та результатом. У контексті промислового виробництва навчання з учителем дозволяє будувати моделі, здатні прогнозувати ймовірність відмови конкретного елемента обладнання. Для цього використовуються історичні технічні дані, які можуть включати:

- кількість годин роботи з моменту введення в експлуатацію;
- частоту та причини виникнення дефектів;

– інші параметри, що відображають режим навантаження або умови експлуатації.

За допомогою регресійних моделей можна визначити час до можливої несправності (RUL – Remaining Useful Life), що дозволяє переходити до обслуговування за станом, знижуючи ризики аварій і простоїв.

Навчання без учителя застосовується у випадках, коли мітки або чіткі вихідні значення відсутні. Цей підхід дає змогу аналізувати структуру даних, виявляти приховані закономірності та групи спостережень без попереднього маркування. Наприклад, якщо в розпорядженні лише параметри, зібрані з датчиків (температура, тиск, струм, навантаження), але немає явної інформації про несправності, модель може кластеризувати ці дані за схожими характеристиками. Надалі будь-яке нове спостереження, що не відповідає жодному з кластерів «нормальної» роботи, може бути розцінене як потенційна аномалія або відхилення, яке вимагає уваги.

Таким чином, кластеризація та виявлення аномалій – це типові задачі навчання без учителя, що дозволяють реалізувати моніторинг стану обладнання навіть у відсутності повної інформації про відмови в минулому.

### 1.5 Найбільш уживані алгоритми машинного навчання на практиці

Наївні Байєсовські класифікатори відносяться до сімейства простих імовірнісних класифікаторів і виходять з теореми Байєса, яка розглядає функції як незалежні.

Принцип максимуму апостеріорної ймовірності є основою байєсівського класифікатора. Алгоритм його роботи полягає в обчисленні функції правдоподібності кожного з класів для класифікаційного об'єкта, так утворюються апостеріорні ймовірності класів. Відношення об'єкту до якогось класу визначається максимальністю апостеріорної ймовірності.

Теорема Байєса стверджує, оптимальний алгоритм має явний аналітичний вид при відомій щільності розподілу кожного класу. Цей

алгоритм характерний мінімальною ймовірністю помилок та тим, що щільність розподілу класів оцінюється навчальною вибіркою. При відновленні вибірки з маленькою погрішністю, байесовський алгоритм втрачає оптимальність. При короткій вибірці, шанси зробити розподіл під конкретні дані та отримати ефект перенавчання вищі.

Байєсівські методи класифікації включають в себе:

- наївний байєсівський класифікатор;
- лінійний дискримінант Фішера;
- квадратичний дискримінант;
- метод парзеновського вікна;
- метод радіальних базисних функцій (RBF);
- логістичну регресію.

Найбільш поширені підходи відновлення щільності:

- параметричний: призводять до нормального дискримінантного аналізу і лінійного дискримінанту Фішера при умові, що щільності гаусові;
- непараметричний: призводить до методу парзеновського вікна;
- поділ суміші ймовірнісних розподілів: призводить до методу радіальних базисних функцій при умові, що щільності компонентів суміші є радіальними. Також є можливість його розробки за допомогою EM-алгоритму.

Формула байєсівського класифікатора створює багато байєсівських алгоритмів, відмінною рисою одне від одного яких є індивідуальне бачення методів відновлення щільності.

Модель наївного байєсівського класифікатора складається з двох типів ймовірностей, які розраховуються за допомогою тренувальних даних:

- ймовірність кожного класу;
- умовна ймовірність для кожного класу при кожному значенні змінної.

Після розрахунку ймовірнісної моделі її можна використовувати для передбачення з новими даними за допомогою теореми Байєса. Наївний

Байес має таку назву через те, що алгоритм передбачає, що кожна вхідна змінна незалежна. Це припущення, яке не відображає реальні дані. Проте цей алгоритм ефективний для цілого ряду складних задач по типу класифікації спаму або розпізнавання рукописних цифр.

Для групування об'єктів або їх ознак застосовуються прийоми класифікації та кластеризації. Створення тематичних наборів зумовлене можливістю автоматичного групування тематично близьких об'єктів. Механізм класифікації встановлюється на відібраних об'єктах тільки після завершення стадії автоматичного виявлення згрупованих даних. Визначення правил розподілу об'єктів у відповідні категорії можливо завдяки використанню статистичних кореляцій при класифікації (рисунок 1.1).

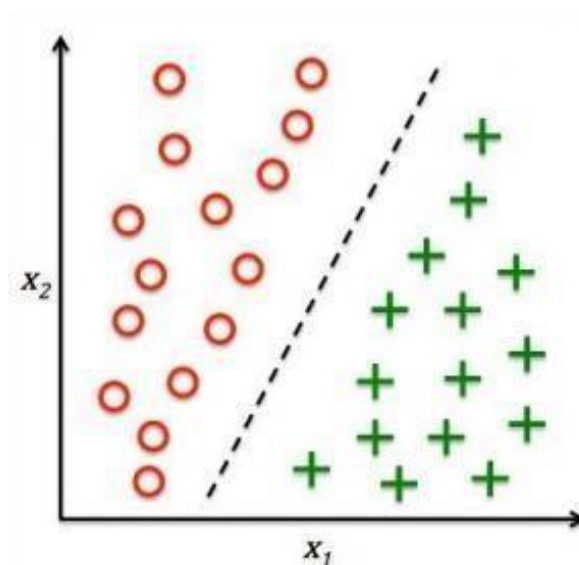


Рисунок 1.1 – Приклад задачі класифікації

Задана схема класифікації та повністю автоматизована сукупність вхідних даних може забезпечити автоматичну класифікацію. Завданням класифікації є формулювання приналежності певного об'єкту певному класу. Алгоритм класифікації під час навчання використовує масиви типових об'єктів, які сформовані експертами. Використовуючи спеціальні

алгоритми, система матиме здатність розподіляти вхідні набори даних на класи, базуючись на досвіді, який отримала при проходженні етапу навчання. Ефективність результатів роботи побудованого класифікатора оцінюється аналізом колекції, яка поділяється на дві складові, незважаючи на рівномірність поділу:

- навчальна колекція;
- тестова колекція.

Характеристики об'єктів навчальної колекції слугують основою процесу побудови класифікатора. В цьому процесі не повинні приймати участь об'єкти тестової колекції, вони використовуються для перевірки якості класифікатора.

Наявність двох непересічних категорій визначає класифікацію як чітку бінарну. Відрізок  $[0, 1]$  визначає множину значень цільової функції при використанні ранжирування. Принцип ранжирування характеризується тим, що об'єкт має різні ступені приналежності до декількох категорій, що, в свою чергу, зумовлює можливість перетину категорій між собою.

Вибір порогового значення для кожної категорії – є першочерговим прийомом побудови функції ранжирування, що призводить до більш точної класифікації. Відповідність об'єкту до категорії напряму визначається тим, що значення функції перевищує або не перевищує порогове значення.

Способи вибору порогового значення:

- пропорційний метод;
- метод  $k$ -найближчих категорій.

Алгоритм роботи пропорційного методу:

- розподіл навчальної колекції на дві частини;
- обчислення кількості об'єктів, що належать до кожної категорії, в одній частині;
- вибір граничних значень з урахування незмінності кількості об'єктів другої частини навчальної колекції, що відносяться до іншої категорії.

Порогове значення обирається відповідно до того, як кожний окремий об'єкт належить до  $k$ -найближчої категорії.

Кількісне вираження ознаки категорії у формі комбінації векторів вагових значень об'єктів натякає на використання регресійного аналізу. До якої категорії буде відноситися об'єкт визначається на основі отриманої комбінації. Лінійна регресія, як стандартний статистичний метод, з легкістю вирішує це завдання. Принцип регресії – це один з варіантів лінійної класифікації, навчання якої відбувається на всій колекції.

Метод найменших квадратів – це вид регресійного аналізу, результатом якого є отримання лінії, що забезпечує оптимальну візуальну демонстрацію зв'язку між точками вхідного набору даних. Зв'язок між відомою незалежною змінною та невідомою залежною змінною характеризує кожену точку даних. Метод найменших квадратів визначає розташування найбільш підходящої лінії для точок даних, що оброблюються. Метою створення цієї лінії є мінімізація суми квадратів помилок, що виникають після вирішення зв'язаних рівнянь, таких як очікуване значення.

Логістична регресія схожа на лінійну тим, що в ній треба знайти значення коефіцієнтів для вхідних змінних. Різницю складає те, що вихідне значення перетворюється за допомогою нелінійної або логістичної функції. Передбачення логістичної регресії можна використовувати для відображення ймовірності приналежності зразка до класу 0 або 1. Це допомагає, коли потрібно мати більше обґрунтувань для прогнозування. Логістична регресія зможе працювати ефективніше, якщо прибрати зайві або схожі змінні. Модель логістичної регресії швидко навчається і добре підходить для задач бінарної класифікації. Завдяки логістичній регресії є можливість визначати залежність між змінними, серед яких лише одна залежить від категорії. Через те, що логістична регресія характерна як потужний статистичний метод передбачення подій, який має незалежні змінні, зростає його перевага у використанні на практиці (рисунок 1.2).

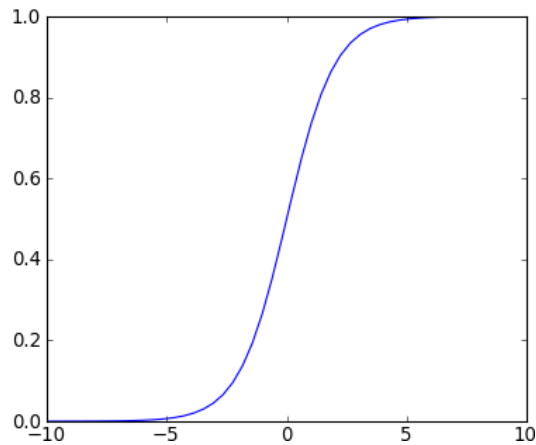


Рисунок 1.2 – Приклад логістичної функції

Метод опорних векторів (SVM) являє собою набір алгоритмів, необхідних для вирішення завдань класифікації та регресійного аналізу. Зробивши висновок, що об'єкт, який знаходиться в  $N$ -вимірному просторі, відноситься до одного з двох класів, спосіб опорних векторів будує гіперплощину з мірністю  $(N-1)$  так, щоб всі об'єкти виявилися в одній з двох груп. Окрім сепарації точок, даний метод створює гіперплощину, яка максимально віддалена від найближчої точки кожної групи (рисунок 1.3).

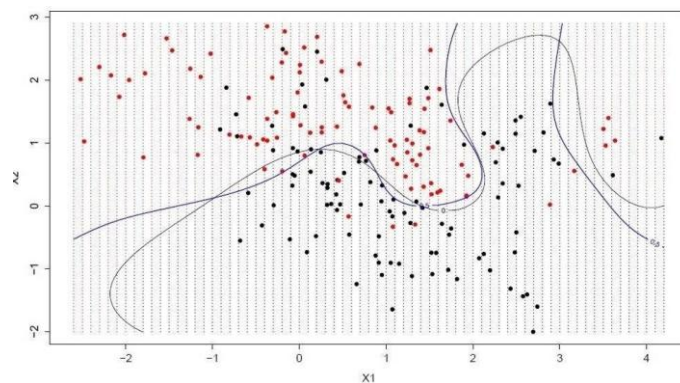


Рисунок 1.3 – Метод опорних векторів

Метод ансамблів полягає у створенні безлічі класифікаторів та розподілі об'єктів з отриманих даних по принципам усереднення або підсумків голосування. Додаткові алгоритми методу ансамблів включають:

- бустінг (формування ансамблю класифікаторів забезпечує перетворення моделей в сильні в порівнянні з вихідними);
- беггінг (навчання базових класифікаторів при паралельному збиранні ускладнених класифікаторів);
- коригування помилок вихідного кодування.

Переваги в порівнянні з іншими методами прогнозування:

- зведення до мінімального впливу ефекту випадковості за допомогою усереднення помилки базових класифікаторів;
- відсутня можливість перетинання межі безлічі;
- зменшення дисперсії.

Кластеризація – це розподіл безлічі об'єктів за категоріями таким чином, щоб в кожному кластері опинилися найбільш схожі між собою елементи. Кластеризацію можна проводити такими алгоритмами: на основі центра ваги трикутника; на базі підключення; скорочення розмірності; щільності (засновані на просторової кластеризації); імовірнісні; машинне навчання, в тому числі нейронні мережі.

Набір алгоритмів кластеризації:

- метод головних компонентів (PCA);
- аналіз незалежних компонент (ICA);
- сингулярне розкладання (SVD).

Метод головних компонентів полягає у принципі ортогонального перетворення спостережень за взаємопов'язаними змінними в набір головних лінійно некорельованих значень. При наявності високої дисперсії вихідних даних цей метод не працює. Його використання потребує належного рівня обізнаності та опису предметної галузі.

В основі аналізу незалежних компонент лежить процес виявлення прихованих чинників, які впливають на випадкові величини. Цей алгоритм має на меті формування моделі, що підходить до баз багатofакторних даних. Змінні можуть складатися з прихованих змінних, які не містять інформації про принципи їх змішування та являються незалежними

компонентами вибірки. Застосування аналізу головних компонент доречно при потребі виявлення прихованих факторів подій.

Сингулярне розкладання, з точки зору лінійної алгебри, – це розклад прямокутної матриці комплексних або дійсних чисел. Матриця  $M$ , що має розмірність  $[m * n]$  розкладається так, що  $M = U\Sigma V$ , де  $U$  і  $V$  – унітарні матриці, а  $\Sigma$  – діагональна. Метод головних компонент є частковим випадком сингулярного розкладання. Алгоритм принципу роботи сингулярного розкладання: патерн перетворюється в набір базисних компонент, зменшується розмірність цих компонент, вони зіставляються з об'єктами вибірки.

## 1.6 Глибинне навчання, як розвиток машинного навчання

Машинне навчання є однією з ключових підгалузей штучного інтелекту, яка зосереджується на розробці алгоритмів, здатних навчатися з даних і покращувати свою продуктивність у розв'язанні конкретних задач без необхідності явного програмування. Базуючись на статистичних методах та методах оптимізації, машинне навчання дає змогу будувати адаптивні системи, які аналізують вхідні дані, виявляють закономірності та приймають рішення на основі накопиченого досвіду.

Одним із найбільш перспективних напрямів розвитку машинного навчання є глибинне навчання (deep learning) – підхід, що використовує штучні нейронні мережі з великою кількістю шарів (глибиною). Глибинне навчання також називають ієрархічним або багаторівневим навчанням, оскільки воно дозволяє моделювати високорівневі абстракції даних шляхом багатоступеневої обробки. Кожен наступний шар нейронної мережі витягує дедалі складніші ознаки з попереднього рівня, що дає змогу будувати моделі, здатні до розпізнавання складних структур – зображень, сигналів, послідовностей, а також технічних параметрів обладнання.

Успішне застосування алгоритмів машинного та глибинного навчання

у виробничих задачах передбачає виконання кількох критично важливих кроків:

- ретельний вибір методів попередньої обробки та очищення даних;
- налаштування гіперпараметрів моделей відповідно до структури даних;
- оптимізація архітектури мережі, зокрема кількості шарів, нейронів або дерев у разі використання ансамблевих моделей.

Висока ефективність глибинного навчання обумовлена його здатністю автоматично виявляти релевантні ознаки без необхідності ручного інженерного опису. Це особливо важливо в умовах Індустрії 4.0, де обробляються великі потоки гетерогенних даних у реальному часі.

Впровадження глибинного навчання в промисловості сприяє значному зростанню точності прогнозування, надійності систем технічного обслуговування, а також загальній продуктивності виробничих процесів. Водночас, варто зазначити, що процес проєктування та тренування моделей глибинного навчання є ресурсомістким, вимагає потужних обчислювальних систем та висококваліфікованих фахівців з галузей аналізу даних, прикладної математики та інформаційних технологій.

Попри складність, попит на автоматизовані рішення на основі глибинного навчання в промисловій індустрії постійно зростає. Це пов'язано з необхідністю обробки складних виробничих сценаріїв, підвищення надійності критичних систем та переходу до самостійно функціонуючих виробничих модулів – так званих «розумних фабрик».

## 1.7 Аргументація використання машинного навчання

Застосування алгоритмів машинного навчання в сучасному промисловому виробництві значно прискорює процес аналізу, обробки та прийняття рішень на основі великих масивів даних. Ці алгоритми забезпечують ефективну обробку даних, що надходять з обладнання в

режимі реального часу, а також покращують точність побудови прогнозних моделей. У результаті працівники підприємства мають змогу приймати обґрунтовані управлінські рішення, базуючись на результатах, отриманих з оброблених і корельованих наборами даних.

Для промислових підприємств, як великих, так і малих, пріоритетом є впровадження сучасних стандартів управління технологічними процесами з метою підвищення ефективності та надійності виробництва. У випадку невеликих виробничих об'єктів, обсяги даних обмежуються переліком основного обладнання, параметрами експлуатації, історією обслуговування та ремонтів. Обробку таких даних ще може виконувати оператор вручну. Але на великих підприємствах, де одночасно експлуатується велика кількість машин та установок, які генерують потоки інформації з різних джерел, ручна обробка втрачає ефективність. Саме тут доцільним стає застосування систем машинного навчання, здатних адаптуватися до високої динаміки даних та забезпечувати їх аналіз безперервно.

Ключовими складовими успішного застосування ML у виробничій сфері є:

- великі обсяги вхідних даних (Big Data);
- структурована історія параметрів експлуатації;
- навчальний етап для формування правил прийняття рішень.

Основні напрями застосування аналітичних систем з використанням машинного навчання включають:

- контроль та оптимізація технологічних процесів;
- прогнозування якості продукції;
- оцінка економічних показників;
- виявлення потенційних поломок або відмов обладнання.

Процес формування моделей у системах прогнозного технічного обслуговування може реалізовуватись двома шляхами: Експертна система, яка перебирає можливі варіанти параметрів процесу і вибирає оптимальний; Автоматизований підхід, який використовує ЕОМ та алгоритми оптимізації

для пошуку найкращих параметрів на основі історичних і поточних даних.

Серед методів математичної оптимізації, що використовуються в машинному навчанні:

- нейронні мережі для нелінійного прогнозування;
- алгоритми градієнтного спуску та похідні методи першого й другого порядку;
- еволюційні методи та генетичні алгоритми для задач комбінаторної оптимізації.

Завдання прогнозування стану виробничого обладнання потребує створення адаптивних моделей, які не лише реагують на зміни, а й вміють самостійно оновлювати внутрішні параметри на основі вхідних даних. Це особливо важливо при роботі з неструктурованими або частково маркованими наборами даних, характерними для промислових середовищ.

Система машинного навчання повинна: встановлювати зв'язки між різними джерелами інформації; виявляти закономірності у зміні технологічних параметрів; оперативно ідентифікувати потенційні відхилення та ризики (рисунок 1.4). Переваги впровадження машинного навчання:

- підвищення продуктивності праці за рахунок автоматизації рутинного аналізу;
- покращення систем обліку та моніторингу технічного стану;
- підвищення якості управління обслуговуванням;
- мінімізація витрат на ремонт і простої;
- скорочення часу реагування на потенційні несправності;
- оптимізація виробничих процесів через прогнозування попиту та планування ресурсів;
- зниження людського фактору в критичних процесах;
- збирання та аналіз великих обсягів даних для ухвалення стратегічних рішень.

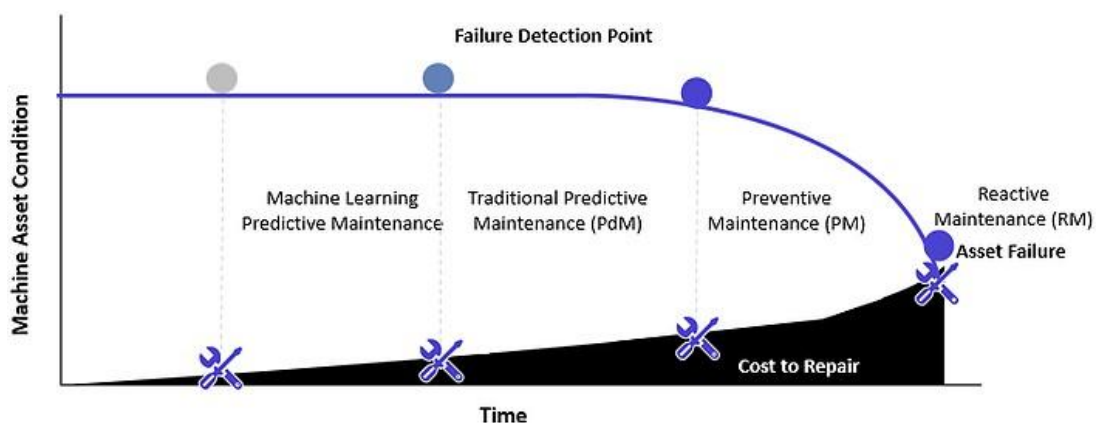


Рисунок 1.4 – Процес виявлення помилок у виробничому процесі

Завдяки впровадженню систем машинного навчання на підприємствах з'являється можливість виявляти несправності обладнання на ранніх стадіях, значно знижуючи витрати на ремонт та запобігаючи серйозним наслідкам. Найбільш ефективними в таких сценаріях є алгоритми класифікації без учителя, які дозволяють визначити відхилення без потреби у попередньому маркуванні усіх можливих станів обладнання. Такий підхід є основою побудови інтелектуальних систем технічного обслуговування в умовах Індустрії 4.0.

### 1.8 Переваги та виклики впровадження ML у промисловості

Практичне впровадження систем технічного обслуговування, що базуються на алгоритмах машинного навчання, супроводжується низкою викликів, що потребують урахування як технічних, так і організаційних аспектів.

Основні переваги машинного навчання в промисловості:

- зменшення незапланованих простоїв – алгоритми машинного навчання дозволяють виявляти ознаки потенційних відмов ще до їх фактичного настання, що значно зменшує кількість аварійних зупинок обладнання;

- оптимізація витрат на технічне обслуговування – прогножуючи залишковий ресурс вузлів та агрегатів, система забезпечує обслуговування лише за потреби, що знижує витрати на запасні частини та трудові ресурси;

- підвищення продуктивності та якості виробництва – за рахунок прогнозного аналізу стану обладнання зростає стабільність технологічного процесу та якість кінцевої продукції;

- автоматизація прийняття рішень – інтелектуальні системи здатні самостійно ініціювати сповіщення, заявки на обслуговування або коригування режиму роботи без участі оператора;

- масштабованість та адаптивність – системи можна легко адаптувати під нові типи обладнання або процесів, змінюючи архітектуру моделей чи додаючи нові джерела даних;

- розширення аналітичних можливостей – алгоритми здатні працювати з багатовимірними, неструктурованими даними (наприклад, звукові сигнали, зображення), що відкриває нові напрямки в діагностиці.

Основні виклики та обмеження при реалізації систем технічного обслуговування:

- доступність та якість даних – у багатьох виробництвах історичні дані відсутні, неповні або містять помилки, що ускладнює навчання моделей;

- інтеграція з існуючими системами – поєднання алгоритмів машинного навчання з SCADA, MES, ERP-системами часто вимагає значної адаптації та додаткових витрат на IT-інфраструктуру;

- недостатня компетенція персоналу – впровадження підходів машинного навчання вимагає знань з Науки про дані, що може бути бар'єром для традиційних інженерних колективів;

- пояснення моделей (Explainability) – для складних моделей, особливо нейронних мереж, важко пояснити причини конкретного прогнозу, що ускладнює їх прийняття в критичних системах;

- кібербезпека та захист даних – передача та зберігання даних у

відкритих мережах створює ризики, які потрібно враховувати при розгортанні інтелектуальних систем;

– проблеми супроводу та масштабування – моделі потребують регулярного оновлення, перенавчання, контролю якості, що вимагає створення спеціалізованої IT-інфраструктури.

Попри значні переваги, широке впровадження машинного навчання в промисловому середовищі вимагає системного підходу: технічної готовності інфраструктури, високої якості даних, інтеграції з корпоративними системами управління та підготовки персоналу. Успішне подолання викликів дозволяє реалізувати повноцінну цифрову екосистему обслуговування, яка підвищує надійність виробництва, зменшує втрати і формує основу для сталого розвитку підприємств у цифрову епоху.

### 1.9 Огляд сучасних ML-підходів у технічному обслуговуванні

Прогнозне технічне обслуговування передбачає використання моделей машинного навчання для аналізу технічного стану обладнання з метою виявлення потенційних відмов до їх фактичного настання. У цій сфері застосовуються як класичні статистичні моделі, так і сучасні підходи глибокого навчання, здатні ефективно працювати з великими масивами промислових даних, зокрема часовими рядами, сигналами від польових пристроїв та логами систем керування.

Методи машинного навчання, що застосовуються в прогнозному технічному обслуговуванні, умовно поділяються на кілька груп залежно від типу задачі:

– класифікація стану обладнання – основною метою є віднесення об'єкту (наприклад, насос або двигун) до одного з визначених класів стану: «нормальний», «зношений», «критичний» (Logistic Regression, Decision Trees, Random Forest, Gradient Boosting (XGBoost, LightGBM));

– регресія та оцінка залишкового ресурсу (RUL) – прогнозування

часу до можливої відмови або інших кількісних показників (наприклад, тиск, температура в майбутньому) (Linear Regression, SVR (Support Vector Regression), Ensemble Methods (Boosting, Bagging), Deep Neural Networks);

– виявлення аномалій – використовуються для знаходження нетипової поведінки без потреби в мітках, особливо корисні в умовах відсутності історії відмов (K-Means, DBSCAN, One-Class SVM, Autoencoders, Isolation Forest);

– аналіз часових рядів (Time Series Forecasting) – моделі передбачають динаміку змін параметрів у часі, що особливо важливо для моніторингу технічного стану в режимі реального часу (ARIMA, SARIMA, Prophet, LSTM (Long Short-Term Memory), TCN (Temporal Convolutional Networks)).

AutoML-платформи, такі як TPOT, H2O.ai, Google AutoML, автоматизують вибір алгоритмів, їхню конфігурацію, попередню обробку даних і оцінювання продуктивності. Це особливо актуально для підприємств, які не мають доступу до великої команди фахівців з машинного навчання.

У складних виробничих системах ефективними є гібридні моделі, які поєднують:

- фізичне моделювання (rule-based logic);
- машинне навчання (для обробки даних реального часу);
- експертні системи (для інтеграції знань персоналу).

Такий підхід дозволяє враховувати як емпіричні знання, так і статистичні закономірності в роботі обладнання.

При виборі підходу слід ураховувати:

- тип даних (кількісні, якісні, часові ряди);
- наявність історичних відмов (для supervised learning);
- розмір та якість датасету;
- інтерпретованість результатів (особливо важливо для критичних об'єктів);

– обчислювальні ресурси (deep learning потребує GPU/TPU).

Сучасні методи машинного навчання, які наведені в таблиці 1.1, надають широкі можливості для впровадження інтелектуального технічного обслуговування у виробництві. Вибір алгоритму визначається конкретною задачею, характером даних та вимогами до точності й інтерпретованості результатів. Комбінація методів у гібридних системах є потужним інструментом підвищення надійності промислових процесів та реалізації повноцінних рішень прогностного технічного обслуговування в рамках Індустрії 4.0.

Таблиця 1.1 – Порівняльна таблиця підходів машинного навчання

Підхід	Тип задачі	Переваги	Недоліки
Random Forest	Класифікація	Інтерпретованість, стійкість до шуму	Важко працює з послідовностями
LSTM	Time Series	Висока точність на динамічних даних	Необхідність великих датасетів
Autoencoder	Аномалії	Без вчителя, виявлення нових шаблонів	Потребує налаштування структури
XGBoost	Регресія/ Класифікація	Швидкість, точність	Складна інтерпретація

### 1.10 Перспективи розвитку

Індустрія 4.0 та пов'язана з нею трансформація процесів технічного обслуговування знаходяться у фазі стрімкого розвитку. Впровадження машинного навчання в системи управління життєвим циклом обладнання вже продемонструвало вагомі переваги, однак у найближчі роки очікується ще ширше використання новітніх технологій, що здатні кардинально змінити підходи до діагностики, моніторингу та ремонту:

– перехід до автономного обслуговування (Self-maintaining systems):

інтелектуальні системи обслуговування майбутнього будуть не лише прогнозувати відмови, а й самостійно планувати ремонтні дії, розміщувати замовлення на запчастини, комунікувати з іншими виробничими модулями та виконувати оптимальні сценарії реагування. Така автономність забезпечуватиметься синергією ML, оптимізаційних алгоритмів та ІоТ (Industrial IoT);

- використання цифрових двійників (Digital Twins): одним із найперспективніших напрямів є побудова цифрових двійників – віртуальних копій фізичних об'єктів, які точно відтворюють їхню поведінку в реальному часі. Вони дозволяють моделювати наслідки відмов ще до їх настання, тестувати нові режими роботи без впливу на реальне виробництво, проводити обслуговування на основі симуляційних прогнозів та порівнювати фактичну поведінку з очікуваною;

- Edge AI та обчислення на периферії: ще один напрям – перенесення обробки даних ближче до джерела (edge computing), що дозволяє зменшити затримки, знизити навантаження на мережу та покращити надійність. Edge AI поєднує обчислювальні можливості з алгоритмами машинного навчання без потреби постійного підключення до хмари;

- пояснюваний штучний інтелект (Explainable AI, XAI): оскільки індустриальні системи обслуговування часто є критично важливими, виникає потреба у поясненні рішень, які приймають моделі. XAI дозволить фахівцям зрозуміти логіку моделей, виявити джерела помилок, підвищити довіру до прогнозів та обґрунтувати дії;

- інтеграція з ERP/MES/SCADA у єдиний інформаційний простір: очікується подальше зближення систем обслуговування з виробничими, планувальними та фінансовими контурами підприємства. Це дозволить оперативно враховувати стан обладнання в планах виробництва, автоматично коригувати графіки технічного обслуговування на основі навантаження та узгоджувати логістику та бюджетування обслуговування;

- використання моделей самонавчання (self-learning systems):

системи, що здатні адаптуватися до змін у поведінці обладнання без участі людини, будуть набувати все більшої актуальності. Постійне перенавчання моделей на нових даних дозволить підвищити точність прогнозів навіть у складних умовах змін навколишнього середовища чи технологічного процесу;

– стандартизація та відкриті платформи: розвиток стандартів (IEC 62890, ISA-95, OPC UA FX) та відкритих платформ обробки даних (Eclipse BaSys, FIWARE) спростить інтеграцію різномірних компонентів систем технічного обслуговування, знизить вартість впровадження та підвищить повторюваність рішень.

Системи технічного обслуговування, побудовані на основі машинного навчання, еволюціонують у напрямку самоадаптивних, інтероперабельних і пояснюваних цифрових екосистем. Перспективи їхнього розвитку охоплюють як інженерні технології, так і організаційні зміни, що веде до формування нового покоління промислових підприємств – гнучких, автономних і максимально ефективних.

## 2 АНАЛІЗ ТЕХНІЧНИХ ЗАСОБІВ ТА МЕТОДІВ

### 2.1 Опис Freelance Engineering від ABB

Для реалізації нижнього рівня, який виконує функції збору даних з польових пристроїв на програмований логічний контролер, було обрано розподільчу систему серії Freelance від шведсько-швейцарської компанії ABB. Важливою перевагою цієї системи є наявність функціоналу для імітації реального обладнання за допомогою емулятора контролера.

Freelance Engineering – це інженерний інструмент Freelance. Він використовується для налаштування та введення в експлуатацію всіх функцій автоматизації. Усю систему Freelance можна налаштувати або онлайн, коли Freelance Engineering підключена до контролера, або офлайн. Для офлайн-конфігурації контролер не потрібен. Прикладну програму, створену під час офлайн-конфігурації, можна пізніше завантажити на контролер. Зокрема, це також стосується конфігурацій FOUNDATION Fieldbus. Freelance Engineering може бути використаний для створення програми керування в польових умовах навіть без наявності будь-яких пристроїв.

Freelance Engineering пропонує такі функції для конфігурації, параметризації та введення в експлуатацію:

- єдиний програмний інструмент для конфігурації, функцій автоматизації, інтерфейсу оператора з дисплеями та журналами, а також параметрів польової шини: Графічне конфігурування з потужними редакторами відповідно до IEC 61131-3 будь-якою з наступних мов програмування: Схема функціональних блоків (FBD), Список інструкцій (IL), Рейкова діаграма (LD), Послідовна функціональна схема (SFC), Структурований текст (ST);

- бібліотека функціональних блоків з більш ніж 220 перевіреними функціями, що значно перевищують базові, описані в IEC 61131-3;

- велика бібліотека макросів, що містить понад 200 графічних символів, які можуть бути розширені користувачем;
- дерево проєкту для гнучкої генерації та прозорого структурування програм;
- перевірка функцій автоматизації з можливістю швидко та легко знаходити та усувати помилки;
- наявна можливість створення перехресних посилань на функції, що дозволить легко знаходити змінні та теги в будь-якому редакторі аж до графічного дисплея;
- імпорт та експорт програм, дисплеїв, змінних, тегів та частин дерева проєкту;
- захист паролем для запобігання несанкціонованому доступу до модифікації проєкту;
- захист паролем для блоків функцій, визначених користувачем;
- єдина та автоматично згенерована загальносистемна графічна документація всієї користувацької програми, системного зв'язку та всіх параметрів польових пристроїв;
- інтегрована довідка в режимі онлайн;
- резервне копіювання файлу проєкту (додатку) на будь-який носій даних (жорсткий диск, CD, флешка, SD-карта AC 900F та AC 700F). Файл проєкту містить повний проєкт з усіма програмами, графікою, контролерами та параметрами польових пристроїв;
- тестування та моделювання користувацьких програм (наприклад, блокувань) навіть без підключеного обладнання за допомогою емулятора контролера;
- менеджер Bulk data дозволяє імпортувати списки сигналів з інструментів планування через Excel та швидко дублювати типові рішення.

Дерево проєкту є центральним інструментом для керування всією програмою користувача та введенням в експлуатацію. Усі дані конфігурації проєкту відображаються у вигляді деревоподібної структури. Дерево

проєкту має наступні складові:

- структуровані дані конфігурації проєкту;
- визначені рівні завдань та час їхнього циклу виконання;
- програми, що відповідають рівням завдань;
- можливість відкривати програми, дисплеї та журнали для редагування, копіювання та переміщення;
- перевірка програм на достовірність та контроль статусу обробки;
- імпорт та експорт даних конфігурації проєкту;
- можливість завантаження користувацьких програм в технологічні та операторські станції.

Усі налаштовані сигнали, змінні та теги керуються в системі Freelance, як зазначено в загальній базі даних проєкту. Набір основних частин бази даних проєкту складається з: списку змінних (входи, виходи, внутрішні змінні), списку тегів (функціональні блоки), графіків та програм. Оскільки база даних охоплює всю систему, дані потрібно вводити лише один раз, що дозволяє уникнути подальших потенційних помилок під час конфігурації. Єдиний файл бази даних проєкту спрощує архівування або резервне копіювання. Список змінних та тегів створюється автоматично під час створення програми користувача. Інші функції списку включають:

- зміну назви, коментарів, даних або типу модуля в масштабах усього проєкту;
- пошук та відображення на основі заданих критеріїв;
- функцію перехресних посилань, що дозволяє швидко знаходити всі програми та дисплеї в масштабах всієї системи, в яких використовується вибрана змінна або тег.

Схема функціональних блоків (FBD) – це графічна мова програмування. Вона містить один або кілька функціональних блоків. Входи та виходи функціональних блоків можуть бути з'єднані для створення потоку сигналів. Freelance Engineering перевіряє, чи можна з'єднати термінали двох функціональних блоків. Входи завжди відображаються

ліворуч, а виходи завжди праворуч від функціонального блоку. За допомогою змінних можна посилатися на значення з однієї схеми до іншої. Доступні два різні типи доступу до змінних: читання та запис. У той час як змінні записуються одним функціональним блоком, змінні читання можуть використовуватися кількома блоками. Розташування терміналів та колір ліній потоку сигналів надають інформацію про тип даних. Усі параметри функціональних блоків визначені на схемі функціональних блоків. Доступні чітко структуровані та зрозумілі діалогові вікна параметрів, в яких можна вносити всі записи, специфічні для блоку. Після завершення схему функціональних блоків можна перевірити за допомогою перевірки правдоподібності на наявність помилок або синтаксичної точності. Будь-які помилки або попередження відображаються у списку, і можна перейти безпосередньо до джерела помилки, просто натиснувши на відповідний рядок у цьому списку. Перехресні посилання в програмі можуть відображатися для всієї системи. Відповідні дисплеї або програми можна викликати безпосередньо, щоб отримати легкий доступ до тегів змінних, на які посилаються. Схема функціональних блоків (програма FBD) налаштовується таким чином:

- визначення назви для програми FBD;
- відкриття редактора для програми FBD;
- вибір функціональних блоків (розташування в графічній області);
- з'єднання функцій з лініями потоку сигналів (введення вхідних та вихідних змінних);
- визначення параметрів для функцій;
- перевірка програми FBD на правдоподібність;
- виправлення будь-яких синтаксичних помилок.

Списки тегів або списки вводу/виводу від клієнтів можна легко імпортувати в систему за допомогою менеджера Bulk data. Типові комбінації функціональних блоків можна легко створити або адаптувати в таблиці Excel. Ця функція корисна на етапі планування або базової

інженерії, особливо для великих проєктів.

Послідовна функціональна схема (SFC) дозволяє легке та прозоре, графічне створення послідовних програм керування. Для створення SFC-програми кроки конфігуруються з призначеними діями (командами) та переходами з умовами активації кроків. Програми (функціональна блок-схема, східчаста діаграма, структурований текст або список інструкцій) можуть бути призначені крокам і переходам. Ще однією особливістю послідовної функціональної схеми є можливість створення альтернативних та паралельних гілок, а також синхронізація цих послідовних структур. Одночасно з конфігурацією послідовної функціональної схеми автоматично генерується дисплей SFC для роботи та спостереження на операторській станції.

Структурований текст – це одна з текстово-орієнтованих мов програмування IEC 61131-3, в якій обробка програм визначається інструкціями. Усі функції та функціональні блоки також можуть використовуватися в ST-програмах. Область дії функцій частково покривається ST-операндами. Функціональні блоки можуть використовуватися в ST-програмі після оголошення. Визначення параметрів функціональних блоків також відбувається так само, як і в східчастій діаграмі або блок-схемі функцій. На відміну від блок-схеми функцій (FBD), область дії функцій структурованого тексту також включає умовні команди та команди циклів, які викликаються за допомогою відповідних ключових слів. Послідовність обробки визначається порядком команд у редакторі ST. Єдиний спосіб спеціально змінити порядок – це вставити команди циклів.

Усі функції обробки Freelance можна визначити за допомогою списку інструкцій (IL). Область дії списку інструкцій перевищує область дії блок-схеми функцій та послідовної блок-схеми, оскільки також можна програмувати команди переходів та цикли програм. Операнди можна відображати та вводити за допомогою списку вибору відповідно до

IEC 61131-3. Визначення параметрів функціональних блоків також використовує ті ж екрани визначення параметрів, що й ті, що використовуються у схемі функціональних блоків.

Поряд зі схемою функціональних блоків (FBD) та послідовною функціональною схемою (SFC), сходова діаграма (LD) також є однією з графічних мов IEC 61131-3. Мова сходових діаграм походить з області електромагнітних релейних систем та описує протікання струму через окремі сходинок. Межі сходинок визначаються праворуч та ліворуч пристроями, відомими як силові шини, які мають логічний стан 1 (струм протікає). Ступінь створюється з елементів сходової діаграми (ланки, контакти та котушки). Функції та функціональні блоки у сходовій діаграмі можна викликати та використовувати так само, як і у схемі функціональних блоків. Параметри також визначаються для функціональних блоків за допомогою тих самих екранів параметрів.

## 2.2 Встановлення та налаштування Freelance Engineering

Програмна реалізація повинна передбачати роботу польових пристроїв для регулювання технологічних параметрів обладнання.

Розробка функціонального програмного забезпечення (ФПЗ) здійснюється за допомогою системи автоматизованого проектування Freelance Engineering з використанням стандартних функцій та операторів.

Для повноцінної роботи середовища програмування передбачається встановлення необхідних системних бібліотек, прошивок обладнання відповідних версій та драйверів для підключення комунікаційних каналів (протоколи Modbus, OPC, Ethernet/IP), як зображено на рисунку 2.1.

Після закінчення встановлення автоматично буде відкрито вікно налаштувань системи Freelance Settings, як зображено на рисунку 2.2. На вкладці «General settings» необхідно вказати IP адресу комп'ютера в полі «Line A», маска підмережі повинна бути «255.255.255.0».

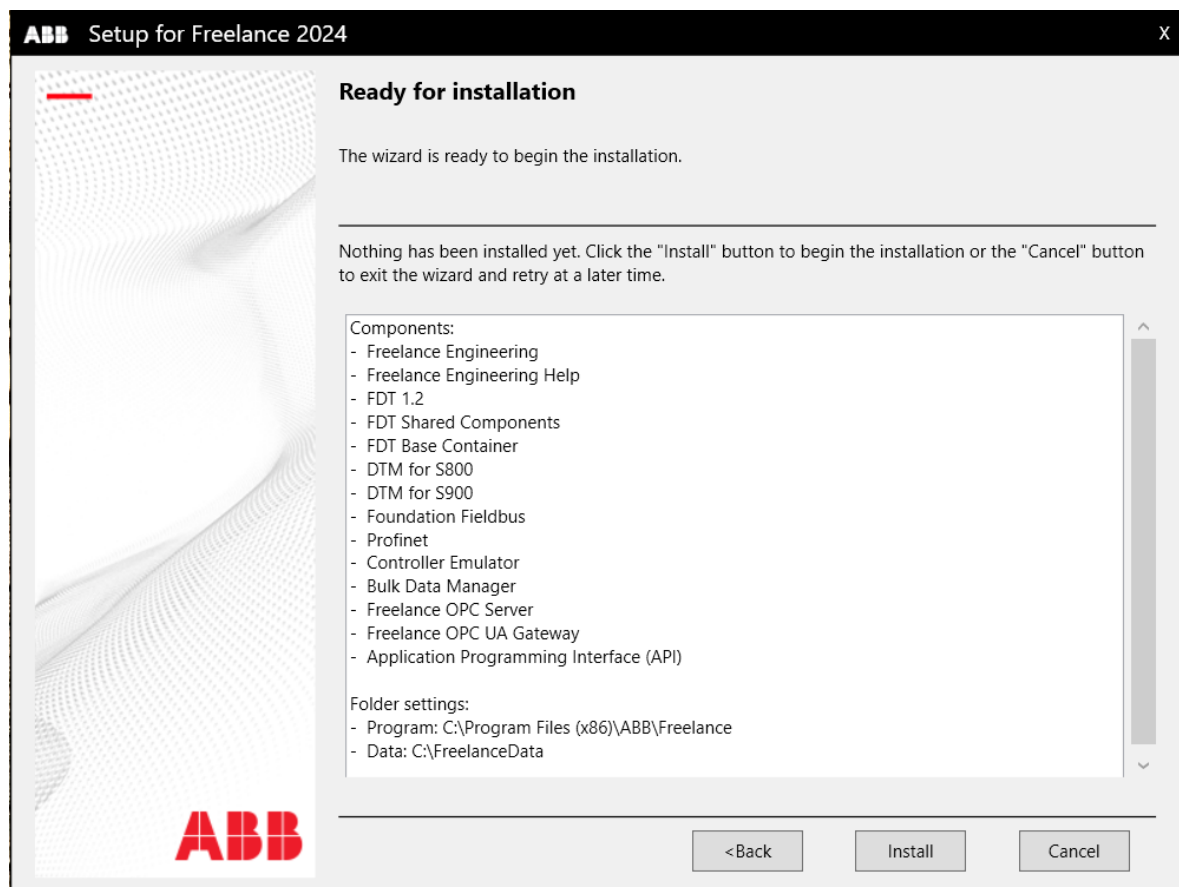


Рисунок 2.1 – Обрані компоненти

Якщо робота передбачає роботу з програмними емуляторами контролерів, значення адреси може бути будь-яким. Якщо передбачається робота з реальним контролером, то адреса має відповідати тій, що прошита в контролері за допомогою апаратного інтерфейсу на передній панелі контролера. Після встановлення мережевої адреси необхідно перезавантажити ПК.

Програмне забезпечення нижнього рівня здійснює зв'язок з програмним забезпеченням верхнього рівня за допомогою протоколу OPC COM. Для реалізації такого зв'язку повинен бути запущений OPC сервер. Для запуску OPC сервера необхідно Freelance Settings, в якій в дереві налаштувань перейти на пункт «Freelance OPC Server Options», у меню в полі «New resource ID:» необхідно вказати ідентифікатор OPC сервера у

системі, як зображено на рисунку 2.3. Ідентифікатор OPC сервера повинен бути унікальним в системі і не збігатися з жодним з ідентифікаторів логічного контролера, програмного емулятора логічного контролера.

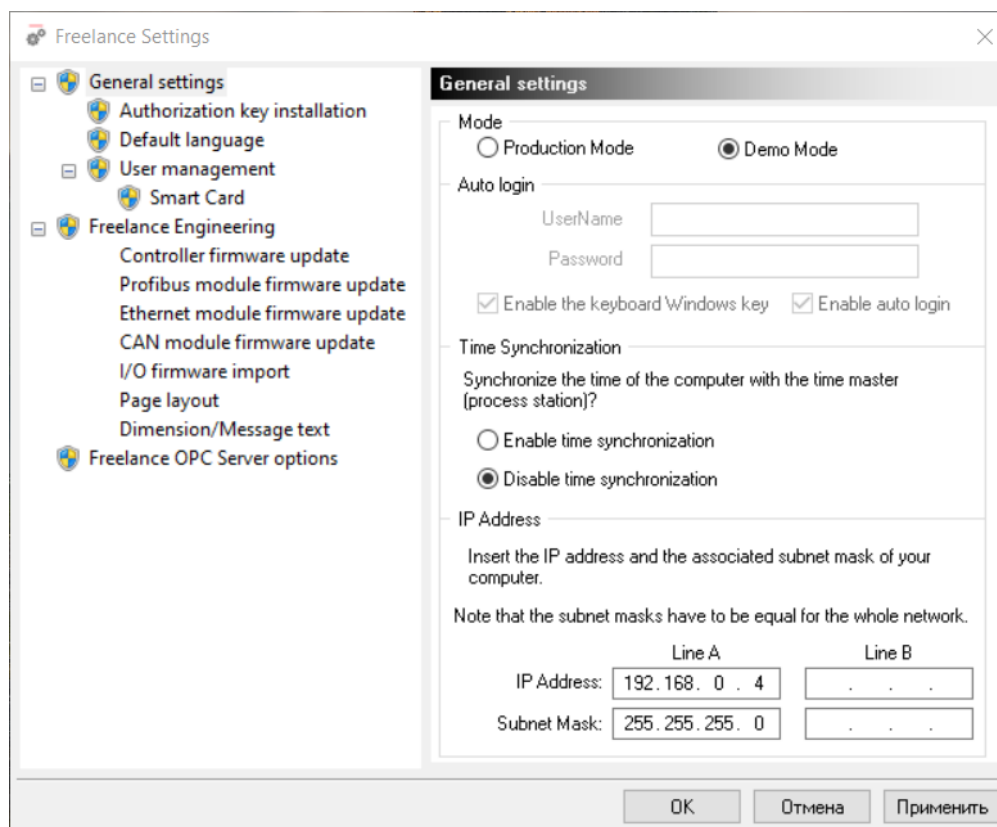


Рисунок 2.2 – Вікно основних налаштувань

Для повноцінної роботи зв'язку між рівнями необхідно встановити додатковий програмний компонент OPC Core Components Redistributables. Це набір бібліотек, драйверів і служб, необхідних для роботи клієнтських та серверних додатків, які використовують OPC Classic (COM/DCOM) технології. Ці компоненти розроблені та підтримуються OPC Foundation – організацією, яка розробляє міжнародні стандарти для інтерфейсів промислової автоматизації. Вони забезпечують основну інфраструктуру для OPC DA (Data Access), HDA (Historical Data Access), A&E (Alarms & Events) – тобто класичних OPC стандартів, які працюють через COM/DCOM у середовищі Windows.

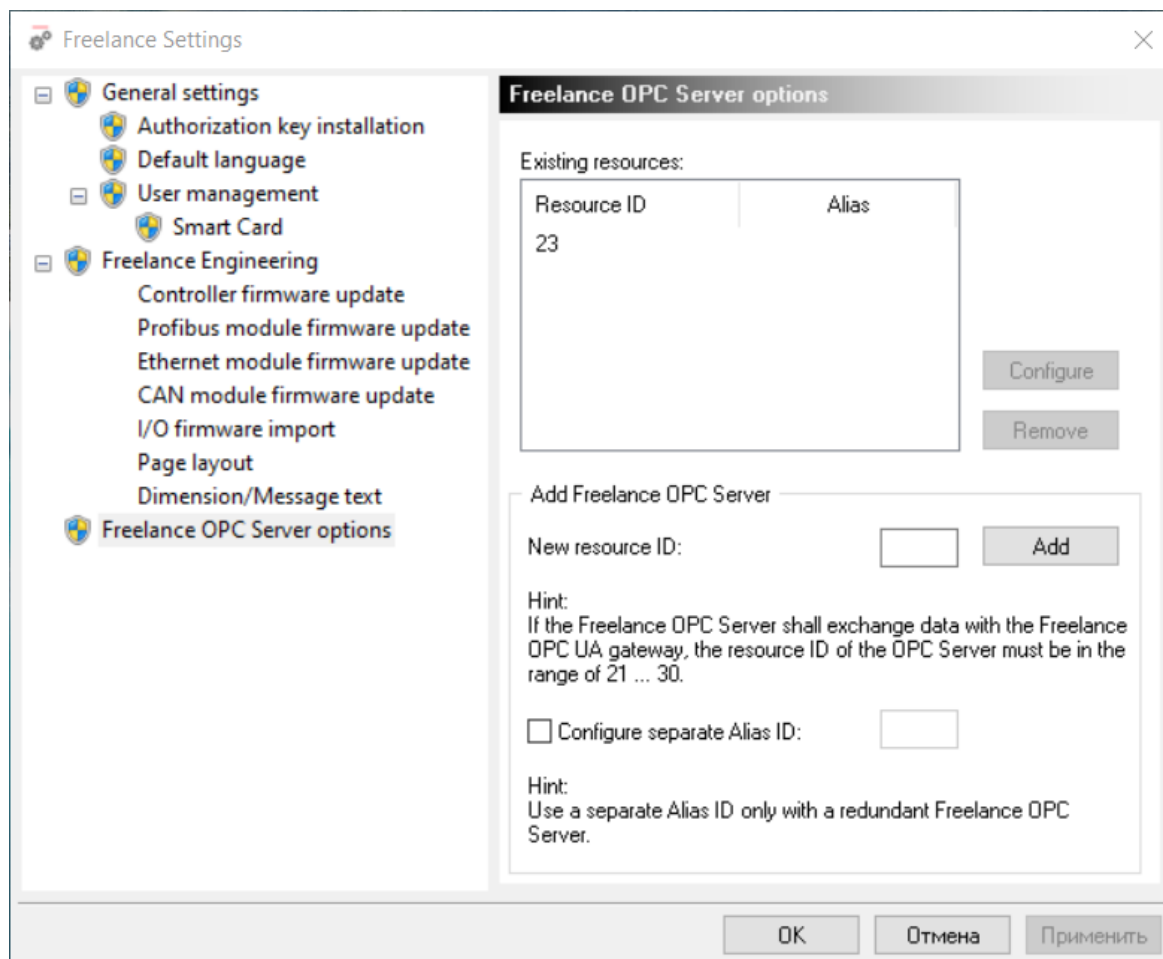


Рисунок 2.3 – Вікно налаштувань OPC-серверу

Налаштування DCOM повинні проводитися тільки після створення всіх необхідних серверів OPC. Послідовність дій під час налаштування DCOM наступна: у командному рядку ввести «dcomcnfg»; у вікні перейти на пункт «Console Root->Component Services>Computers->My Computer»; у контекстному меню «My Computer» вибрати «Properties»; у вікні властивостей: на вкладці «Default Properties» встановити «Default Authentication Level -> None», «Default Impersonation Level -> Identify»; на вкладці «COM Security» у всіх 4-х розділах: «Access Permissions», «Launch and Activation Permissions», «Edit Limits...» та «Edit Limits...» додати користувачів «Everyone» та «ANONYMOUS», надавши їм усі можливі дозволи, як зображено на рисунках 2.4 – 2.7.

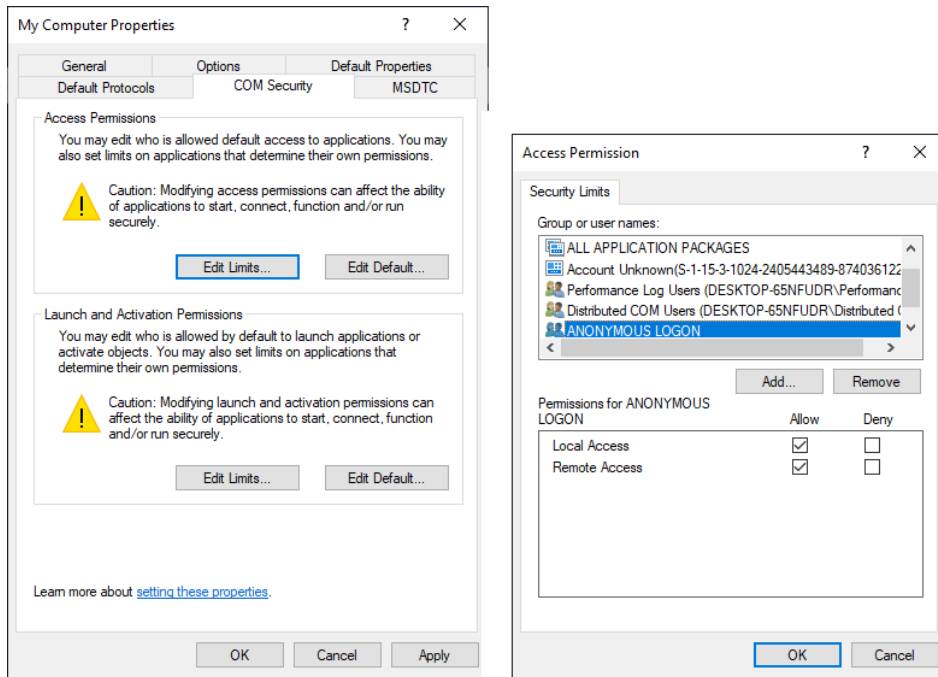


Рисунок 2.4 – Вікно DCOM налаштувань ПК

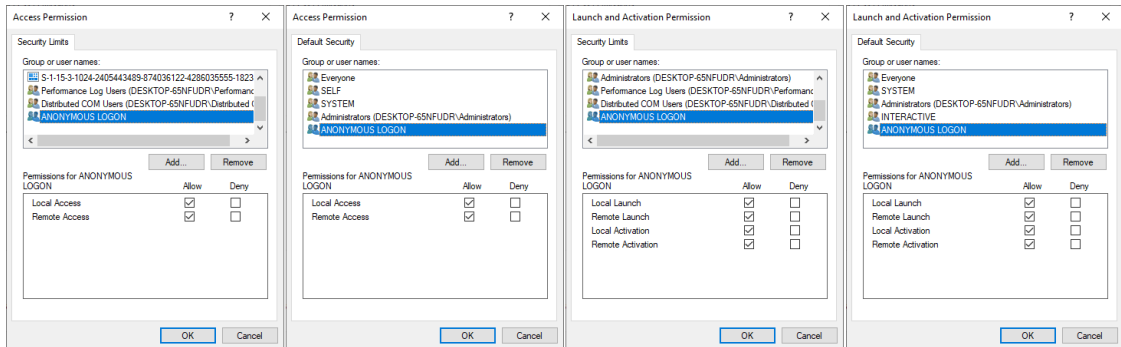


Рисунок 2.5 – Вікно DCOM налаштувань ПК

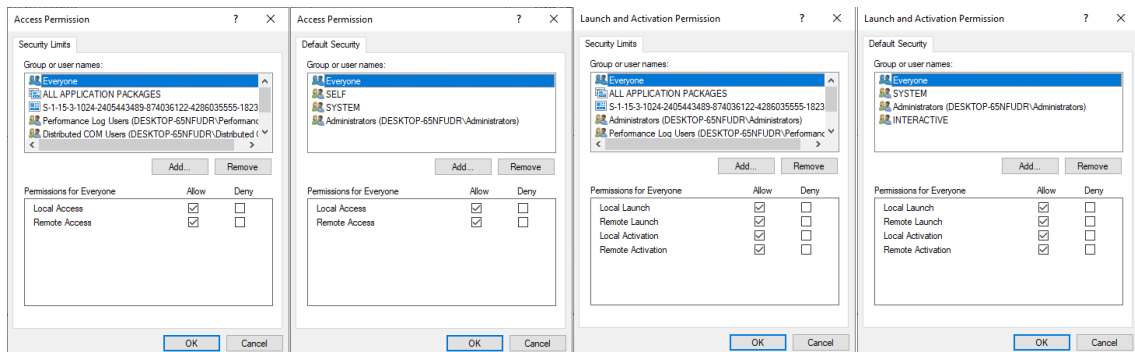


Рисунок 2.6 – Вікно DCOM налаштувань ПК

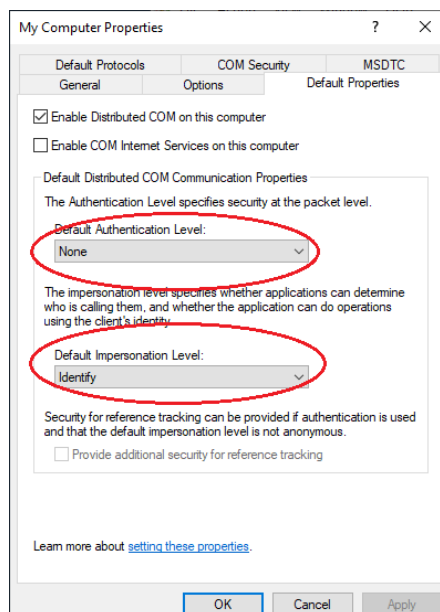


Рисунок 2.7 – Вікно DCOM налаштувань ПК

Після налаштування параметрів ПК необхідно налаштувати параметри серверу OPC. Для цього в папці «DCOM Config» потрібно вибрати необхідні сервери під назвою «Freelance 2000 OPCServer XXX», де XXX—це ідентифікатор, встановлений під час створення OPC сервера. У його контекстному меню, на вкладці «Security» необхідно встановити всі параметри за замовчуванням («Use Default»), як зображено на рисунку 2.8.

Після того, як буде проведено всіх налаштувань верхній та нижній рівні зможуть безперешкодно обмінюватися інформацією про поточний стан системи.

Налаштування та запуск програмного емулятора логічного контролера здійснюється за допомогою вбудованого в систему автоматизованого проектування web-ресурсу. Для запуску емуляторів необхідно за допомогою браузера перейти за посиланням «http://127.0.0.1:8888», ввести в поле «Station No» відповідний номер та натиснути кнопку «Initialize Controller». Після закінчення налаштувань сторінка запуску має виглядати як зображено на рисунку 2.9.

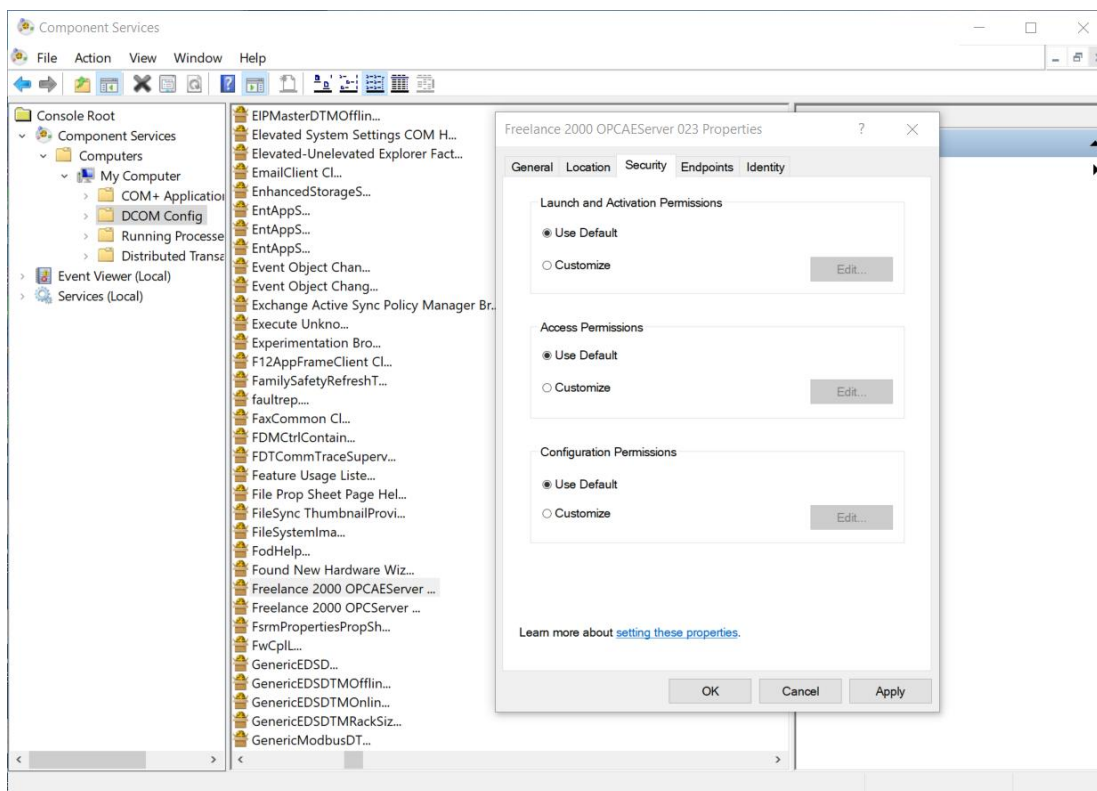


Рисунок 2.8 – Вікно DCOM налаштувань серверу OPC

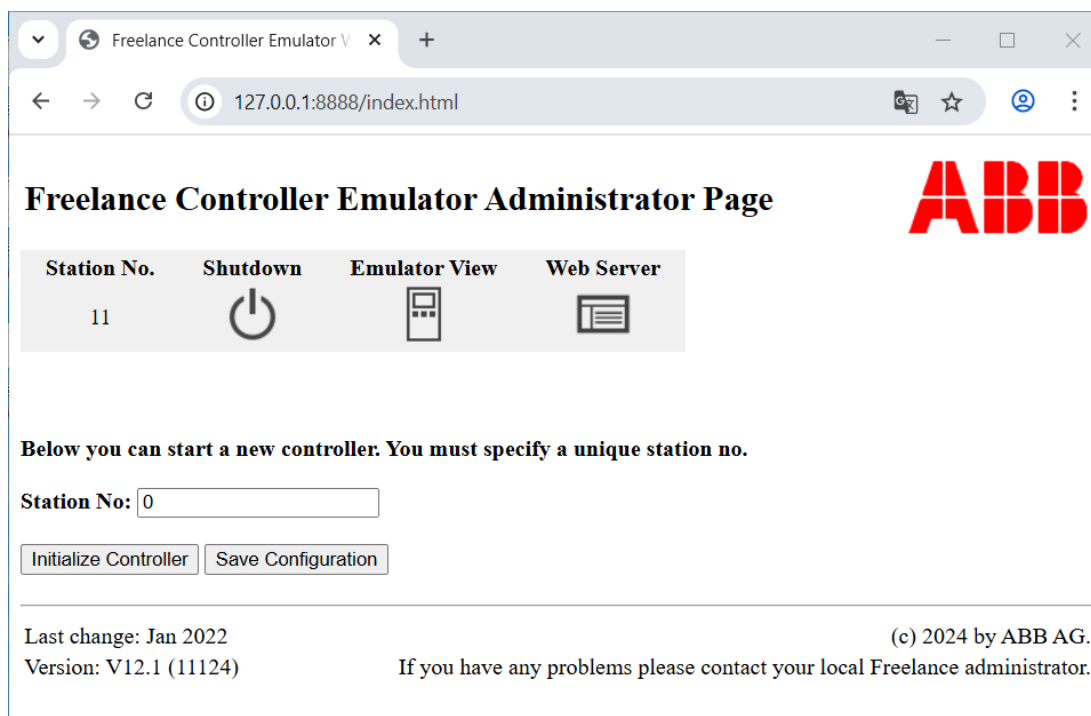


Рисунок 2.9 – Web-ресурс для налаштувань програмного емулятора

### 2.3 Опис використання програми UaExpert

Для належного забезпечення зв'язку між нижнім та верхнім рівнями програми було обрано програмне забезпечення UaExpert, яке орієнтоване на створення клієнт-серверного підходу передачі інформації.

UaExpert – це спеціалізована програма для роботи з технологією OPC UA, що дозволяє здійснювати моніторинг, налаштування та відлагодження мережі промислових пристроїв і систем, що підтримують стандарт OPC UA. UaExpert розроблений як клієнт загального призначення для тестування, що підтримує такі функції OPC UA, як доступ до даних, тривоги та умови, доступ до історії та виклик методів UA.

UaExpert – це кросплатформовий клієнт для тестування OPC UA, запрограмований на C++. Базова структура UaExpert включає загальні функції, такі як обробка сертифікатів, виявлення UA-серверів, підключення до UA-серверів, перегляд інформаційної моделі, відображення атрибутів та посилок на певні UA-вузли. Основною перевагою цього інструменту є спрямованість на забезпечення «класичних» COM/DCOM-базованих підключень OPC-серверів.

Інструмент налаштування UaGateway можна використовувати для додавання, налаштування та підключення серверів Freelance OPC AE/DA. Необхідною умовою є запуск OPC-сервера та UaGateway.

Щоб запустити інструмент конфігурації, треба вибрати «Налаштувати UaGateway» або «Уніфікована автоматизація», потім «Конфігурація UaGateway» у меню «Пуск». Щоб додати новий сервер, треба створити відповідний запис у вікні «Перегляд пристроїв». Вибір серверу відбувається у діалоговому вікні, що випадає, як зображено на рисунку 2.10.

UaGateway автоматично підключається до щойно доданих COM-орієнтованих серверів, як зображено на рисунку 2.11.

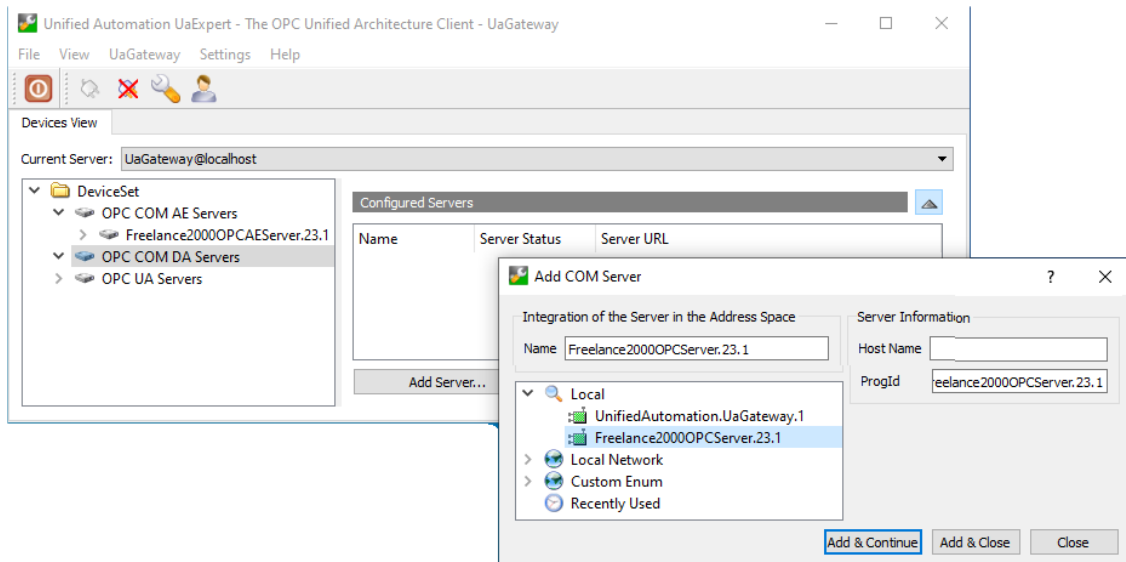


Рисунок 2.10 – Вікна додавання OPC серверу

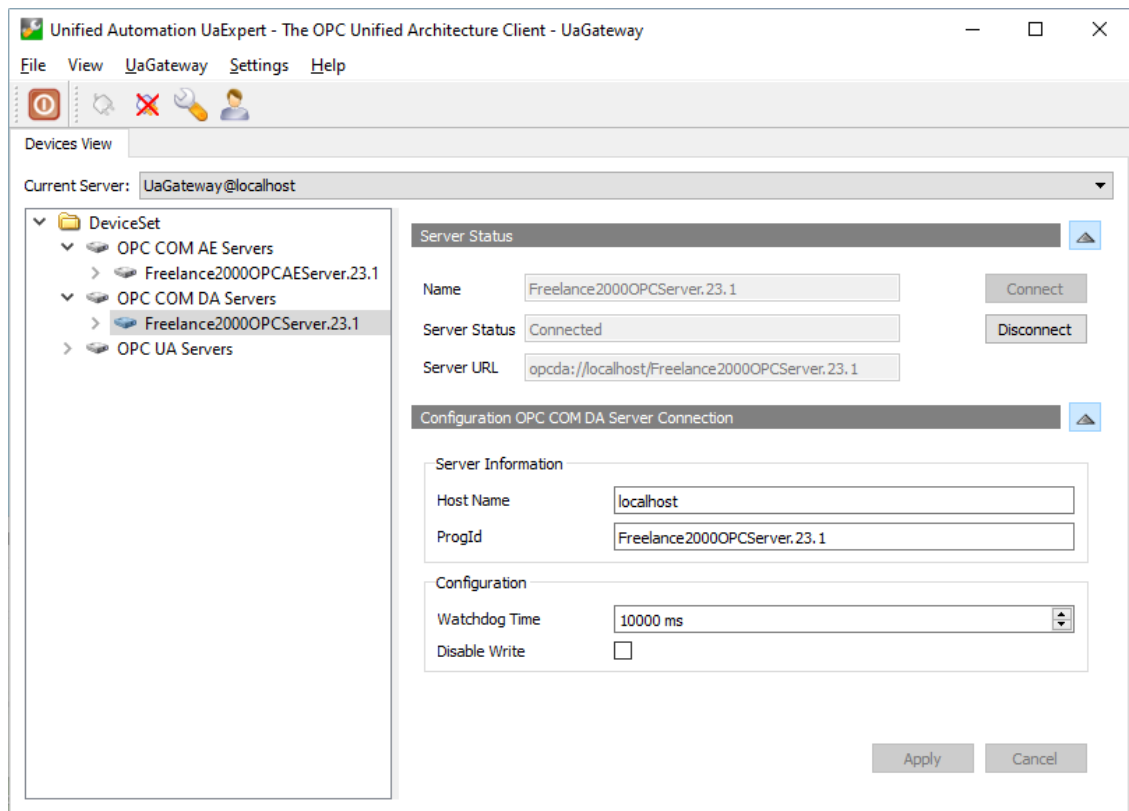


Рисунок 2.11 – Інтерфейс створених OPC серверів

Для створення клієнтського підключення потрібно створити сертифікат ідентифікації. У дереві проєкту UaExpert потрібно додати новий

екземпляр класу Servers. Ім'я позаштатного комп'ютера та порт зв'язку вводяться як URL-адреса кінцевої точки, як зображено на рисунку 2.12. Параметри безпеки та авторизації відповідають параметрам UaGateway. Їх можна налаштувати в адмініструванні UaGateway. Після того, як сертифікат сервера прийнято, з'єднання встановлюється. Відповідно до конфігурації Freelance, змінні можна читати та записувати з системи Freelance.

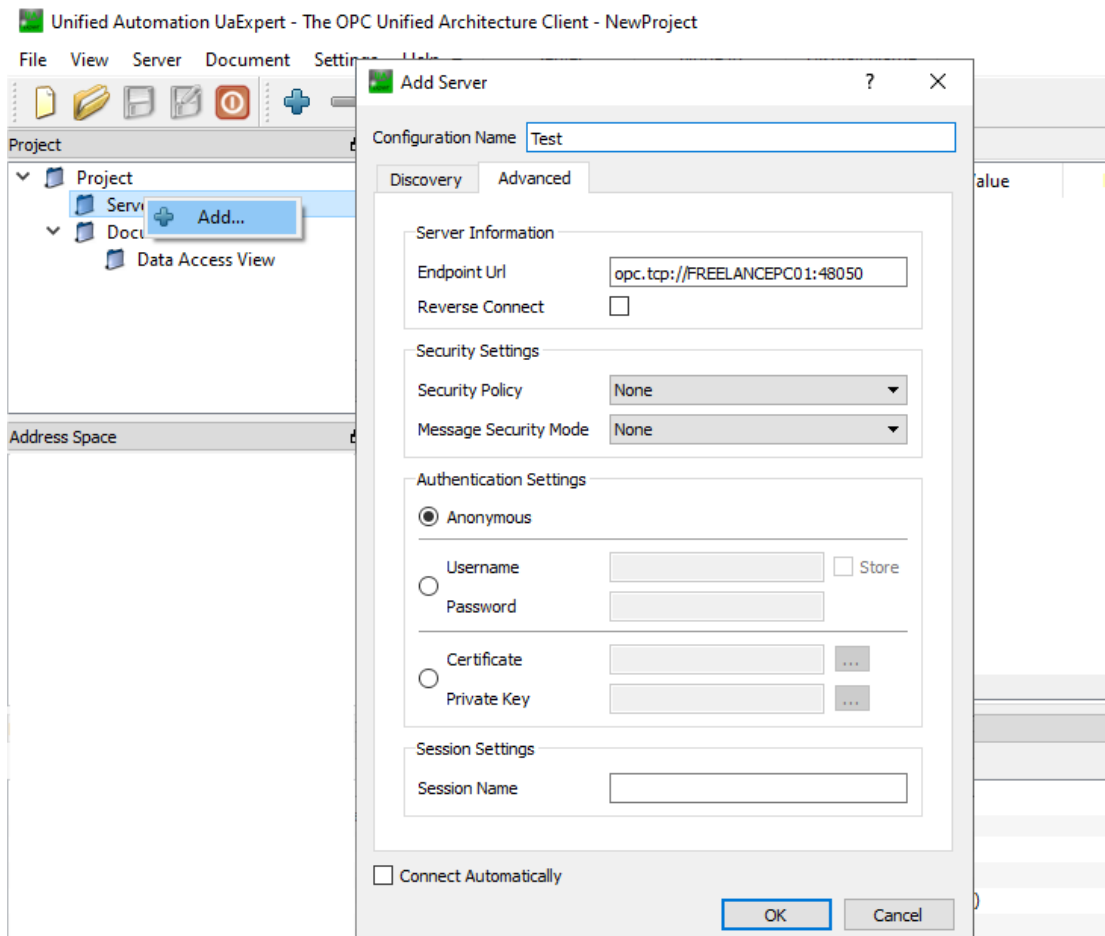


Рисунок 2.12 – Вікно створення клієнтського підключення

## 2.4 Опис Ignition від Inductive Automation

Для інтегрування розробленого програмного рішення було використано SCADA-систему від американської компанії Inductive Automation. Важливою перевагою цієї системи є її базова основа, а саме–

Java- платформа. Ця система є клієнт-серверною системою, яка основана на веб-технологіях, та розроблена на мові Java.

Ignition від Inductive Automation – це наступне покоління серверно-орієнтованого SCADA/HMI програмного забезпечення MES. Це промисловий сервер додатків, що встановлюється як серверне програмне забезпечення, він використовує веб-технології для створення широкого спектру промислових додатків. Ignition досягає широкої функціональності за допомогою модульної архітектури.

Основні модулі, які забезпечують необхідний стартовий функціонал:

- OPC UA сервер – провідний промисловий стандарт для доступу до даних. Використовуючи даний модуль Ignition виступає в якості сервера OPC UA, який відображує дані, зібрані за допомогою вбудованих драйверів для інших модулів Ignition, а також сторонніх клієнтів OPC UA;

- Data Logger (Ignition пропонує надійну функціональність реєстрації даних. Модуль SQL Btridge пропонує історичне логування та тригер збереження тегів. Крім того, функція SQL Tag Historian робить простішим зберігання і використання історичних даних про проходження процесу);

- статус та управління (Ignition пропонує можливість відслідковування статусу проходження процесу та функції управління, він може бути використаний як для створення одного операторського терміналу, так і для розподіленої систем з великою кількістю клієнтів. SQL Tags, система тегів Ignition, забезпечує безліч потужних функцій і виняткову простоту використання. Додавати теги для індикації процесу можна з допомогою функції drag-and-drop, просто перетягнувши теги на вікно. Такі функції, як резервування і панель коригування ліцензій забезпечують створення надійних, стійких до відмов систем);

- сповіщення сервера (гнучка система сповіщень та моніторингу побудована на модулі SQL Tags, шлюз Ignition підтримує широкі можливості фіксації повідомлень. Можливе створення груп розсилки, які дозволяють відправляти повідомлення по електронній пошті з високим

рівнем контролю. Історія аварійних сповіщень легко зберігається в базі даних і може бути відображена, що робить простим відстеження і аналіз загальних проблеми у технологічному процесі);

– аналіз даних (Ignition пропонує функціональний аналіз інформації про процес у вигляді трендів і таблиць. Потужність доступу до бази даних SQL пропонує величезну кількість можливостей в сучасних ІТ-орієнтованих компаній та організацій. Функціональні графіки, таблиці і звіти відкривають нові можливості в сфері аналізу даних).

## 2.5 Встановлення та налаштування Ignition

Для встановлення Ignition на комп'ютері повинно бути встановлено Java- платформу, так як вона не входить в інсталяційний пакет дистрибутиву Ignition. Сам Ignition встановлюється досить швидко, після встановлення можна відразу запустити Ignition Gateway, або він запуститься при наступному запуску операційної системи. Після встановлення Ignition потрібно встановити систему управління базами даних. Необхідно провести початкове налаштування бази даних, включаючи права користувачів та доступ до управління базою даних. Далі на веб-сторінці Ignition Gateway у вкладці конфігурації можна провести налаштування серверного шлюзу для роботи системи. На рисунку 2.13 зображено головне меню вкладки Config.

Одним із важливих пунктів є пункт Projects у блоці налаштувань Configuration, де можна управляти проектами, видаляти їх, дублювати. На вкладці Modules здійснюється управління модулями Ignition, як зображено на рисунку 2.14. Для додавання модулів потрібно натиснути на кнопку «Install or Upgrade a Module...», після чого потрібно вказати шлях до модуля. У вкладці Gateway Settings налаштовуються загальні налаштування роботи Ignition шлюзу, наприклад, кількість оперативної пам'яті, виділеної під роботу шлюзу, версії Java, які підтримуються, частота автоматичних бекапів системи.



Рисунок 2.13 – Головные меню вкладки Config

**Inductive Automation**  
(null/expired certificate)  
 [View Certificate](#)

Name	Version	Description	License	State		
Alarm Notification	5.0.17 (b2020111211)	Provides alarm notifications via email.	Trial	Running	More ▾	restart
Allen-Bradley Driver	5.0.17 (b2020111211)	Allen-Bradley driver suite for the OPC UA module.	Trial	Running	More ▾	restart
BACnet Driver	1.0.17 (b2020111211)	A driver for communicating with devices via BACnet.	Trial	Running	More ▾	restart
DNP3 Driver	3.0.17 (b2020111211)	A driver supporting DNP3 (Distributed Network Protocol) device.	Trial	Running	More ▾	restart
Enterprise Administration	2.0.17 (b2020111211)	A remote Gateway administration system, allowing you to manage Gateways and automate tasks from a single controller.	Trial	Running	More ▾	restart
Logix Driver	4.0.17 (b2020111211)	A driver for communicating with Allen-Bradley Logix5000 series PLCs, and includes firmware version 21 support.	Trial	Running	More ▾	restart
Modbus Driver	6.0.17 (b2020111211)	A driver for communicating with devices via Modbus-TCP.	Trial	Running	More ▾	restart
Omron Driver	3.0.17 (b2020111211)	Drivers for Omron PLCs.	Trial	Running	More ▾	restart
OPC-UA	8.0.17 (b2020111211)	Provides Ignition's OPC UA client and server functionality.	Trial	Running	More ▾	restart
OpCom	5.0.17 (b2020111211)	Bridge that exposes COM based OPC-DA servers to the system.	Trial	Running	More ▾	restart
Perspective	1.0.17 (b2020111211)	A module that provides modern, responsive HTML based graphical interfaces for Ignition projects.	Trial	Running	More ▾	restart
Reporting	5.0.17 (b2020111211)	Adds reporting capability.	Trial	Running	More ▾	restart
Serial Support Client	5.0.17 (b2020111211)	Adds platform-level serial comm support and system.serial functions to client-scoped scripts.	Trial	Loaded	More ▾	restart
Serial Support Gateway	5.0.17 (b2020111211)	Adds platform-level serial comm support and system.serial functions to gateway-scoped scripts.	Trial	Running	More ▾	restart
SFC	4.0.17 (b2020111211)	Sequential Function Charts are customizable logic execution environments.	Trial	Running	More ▾	restart
Siemens Drivers	5.0.17 (b2020111211)	Siemens S7-300, S7-400 and S7-1200 drivers.	Trial	Running	More ▾	restart
SMS Notification	5.0.17 (b2020111211)	Adds SMS notifications to Alarming.	Trial	Running	More ▾	restart
SQL Bridge	5.0.17 (b2020111211)	An OPC-to-SQL data logger and transaction manager.	Trial	Running	More ▾	restart
Symbol Factory	6.0.17 (b2020111211)	Vector graphics clipart library for the Vision module.	Trial	Loaded	More ▾	restart
Tag Historian	3.0.17 (b2020111211)	Turns any database into a powerful historian that can store and drive data in Ignition.	Trial	Running	More ▾	restart
UDP and TCP Drivers	5.0.17 (b2020111211)	Drivers for receiving and parsing UDP or TCP packets.	Trial	Running	More ▾	restart
Vision	10.0.17 (b2020111211)	A module that provides web-launched HMI/SCADA clients.	Trial	Running	More ▾	restart
Voice Notification	5.0.17 (b2020111211)	Provides alarm notifications via phone calls over VOIP.	Trial	Running	More ▾	restart

Рисунок 2.14 – Вкладка Modules

У пункті Security здійснюються налаштування прав доступу до SCADA-системи. Налаштування користувачів може здійснюватися у внутрішній базі даних шлюзу або у встановленій зовнішній. У вкладці Auditing налаштовується логування дій користувачами як в клієнтських вікнах, так і в дизайнері.

Вкладка Databases дає можливість налаштувати всі параметри з'єднання із базою даних, включаючи саме з'єднання, а також JDBC драйвери для різних реляційних систем управління базами даних для встановлення з'єднання та роботи з базою.

Вкладка Alarming дає можливість налаштувати всі параметри аварійних сповіщень системи. Основним налаштуванням є місце збереження всіх аварій (Legacy Storage), так як вони всі повинні зберігатися для аналізу системи чи інших дій по збору інформації.

Вкладка SQL Tags дає доступ до налаштування провайдерів для тегів. Теги реального часу, історія зміни яких ніде не записується працюють з використанням стандартного провайдера тегів.

Для запису історії значень тегу потрібно дати посилання на базу даних, в якій вони будуть зберігатися та налаштувати спосіб збереження. При збільшенні кількості цих значень виникає потреба створювати нові таблиці.

За бажанням можливо налаштувати частоту створення нової таблиці. За замовчуванням нові таблиці створюються кожного місяця. Також є можливість видалення старої інформації, за замовчуванням період видалення становить один рік.

Історичні записи значень тегів потрібні для відображення цих тегів на графіках. Якщо для тегу не підключена функція логування значень в базу, то він не буде доступний для додавання на графіки.

У вкладці OPC-UA налаштовується параметри вбудованого в Ignition OPC UA сервера. Також тут можна додавати, видаляти та налаштовувати вже існуючі пристрої, які під'єднані до даного сервера. В даній вкладці

відображено тип з'єднання, ім'я пристрою та його статус. Для його налаштування потрібно натиснути кнопку «edit». Для додавання нового пристрою потрібно натиснути «Create new Device...» (рисунок 2.15).

Name	Type	Description	Enabled	Status		
KM1_1	UDP Driver		false	Disabled	delete	edit
KM1_2	UDP Driver		false	Disabled	delete	edit
KM2_1	UDP Driver		false	Disabled	delete	edit
KM2_2	UDP Driver		false	Disabled	delete	edit
sim1	Programmable Device Simulator		true	Running	More ▾	edit

Рисунок 2.15 – Вкладка OPC UA Device Connections

Першим кроком створення є вибір драйвера для з'єднання. Як зображено на рисунках 2.16 та 2.17 є як спеціалізовані драйвери для модельних рядів контролерів, так і універсальні драйвери та драйвери для з'єднання із симуляторами контролерів. В залежності від вибраного пристрою буде відображено різні набори налаштувань з'єднання. Але в кожному варіанті є поле імені пристрою і у всіх пристроїв, крім симуляторів контролерів, повинна бути адреса в мережі.

- Allen-Bradley CompactLogix (Legacy)**  
Connect to CompactLogix firmware v20 and prior processors.
- Allen-Bradley ControlLogix (Legacy)**  
Connect to ControlLogix firmware v20 and prior processors.
- Allen-Bradley Logix Driver**  
Connect to Allen-Bradley Logix family devices, including devices with firmware v21+.
- Allen-Bradley MicroLogix**  
Connect to MicroLogix 1100 and 1400 series PLCs.
- Allen-Bradley PLC5**  
Connect to PLC5s via Ethernet.
- Allen-Bradley SLC**  
Connect to SLC 5/05s via Ethernet.
- BACnet/IP**  
Connect to devices that implement BACnet/IP.
- DNP3 Driver**  
Connect to a DNP3 outstation.
- Modbus RTU**  
Connect to devices that implement the Modbus RTU protocol over a serial link.

Рисунок 2.16 – Вкладка додавання драйверу

**Modbus RTU over TCP**  
Connect to devices that implement the Modbus RTU protocol over TCP.

**Modbus TCP**  
Connect to devices that implement the Modbus TCP protocol.

**Omron FINS/TCP**  
Connect to devices that implement FINS over TCP.

**Omron FINS/UDP**  
Connect to devices that implement FINS over UDP.

**Omron NJ Driver**  
Connect to Omron NJ series PLCs.

**Programmable Device Simulator**  
A simulator device that can be configured with a user-defined hierarchy of static or function-driven values.

**Siemens S7-1200**  
Connect to Siemens S7-1200 PLCs over Ethernet.

**Siemens S7-1500**  
Connect to Siemens S7-1500 PLCs over Ethernet.

**Siemens S7-300**  
Connect to Siemens S7-300 PLCs over Ethernet.

**Siemens S7-400**  
Connect to Siemens S7-400 PLCs over Ethernet.

**TCP Driver**

**UDP Driver**

[Next >](#)

Рисунок 2.17 – Вкладка додавання драйверу

У полі OPC Connections є можливість додавання нового OPC сервера, якщо такий встановлено на ПК. Для додавання нового потрібно натиснути «Create new OPC Server Connection...» і у наступних вікнах слідувати інструкціям, вибираючи відповідні характеристики встановленого OPC-серверу (рисунок 2.18).

Name	Type	Description	Read Only	Status	
Ignition OPC UA Server	OPC UA	A "loopback" connection to the Ignition OPC UA server running on this gateway.	false	Connected	<a href="#">More ▾</a> <a href="#">edit</a>
OPC_SHU_1	OPC-DA COM Connection		false	Disabled	<a href="#">delete</a> <a href="#">edit</a>
OPC_SHU_2	OPC-DA COM Connection		false	Disabled	<a href="#">delete</a> <a href="#">edit</a>
OPC_SHU_3	OPC-DA COM Connection		false	Disabled	<a href="#">delete</a> <a href="#">edit</a>
OPC_SHU_4	OPC-DA COM Connection		false	Disabled	<a href="#">delete</a> <a href="#">edit</a>
SARTO_OPC	OPC-DA COM Connection		false	Connected	<a href="#">delete</a> <a href="#">edit</a>

Рисунок 2.18 – Вкладка OPC Client Connections

Веб-запуск є механізмом, за допомогою якого клієнти і дизайнери відкриваються на ПК. Вони запускаються з шлюзу Ignition, який встановлюється як основна складова серверної частини Ignition і функціонує лише на сервері. Клієнти та дизайнери можуть бути відкриті на будь-якому пристрої, який знаходиться в мережі із сервером та має встановлену відповідну версію Java- платформи.

Веб-запуск спирається на технології Java Web Start. Коли користувач натискає на посилання проєкту або дизайнера у вікні шлюзу Ignition чи вбудованого в окремому веб-сайті, завантажується невеликий файл JNLP, який описує додаток. При відкритті цього файлу, Java викликається на комп'ютері користувача і посилається на віддалений додаток на сервері. Додаток завантажується та виконується на комп'ютері користувача.

Запущений застосунок спілкується зі шлюзом через HTTP. Він кешується для підвищення швидкості запуску, і може додатково встановити посилання в меню Пуск і на робочому столі для швидкого доступу. Кешовані проєкти запуску містять адресу посилання на шлюз сервера. Якщо змінилися налаштування серверу, ці посилання не будуть працювати. Вони можуть бути оновлені шляхом повторного запуску від шлюзу. До шлюзу Ignition доступ можна отримати через веб-браузер. За замовчуванням, Ignition встановлює зв'язок через порт 8088, його можна змінити у вікні налаштувань шлюзу.

У верхній частині веб-сторінки розміщено кілька кнопок, які посилаються на ключові розділи сервера.

Домашня сторінка показує короткий огляд основних модулів, а також готові проєкти. Можливі наступні функції:

- перегляд клієнтів-учасників проєктів;
- перегляд поточного стану модуля SQL Bridge;
- перегляд стану з'єднань пристроїв за допомогою використання модулю OPC UA.

Сторінка статусу надає детальну інформацію про різні частини

системи Ignition (рисунки 2.19, 2.20). Є посилання на сторінки, що містять інформацію про:

- стан встановлених модулів;
- стан всіх з'єднань бази даних, з'єднань з OPC-серверами, і SQL Tag постачальників;
- статус механізму Store and Forward, у тому числі показників продуктивності та інформації про кеш даних;
- відкриті в даний момент дизайнери та клієнти, що підключені до шлюзу;
- стан усіх скриптів шлюзу, що працюють у кожному з проєктів на шлюзі Ignition;
- інформація та статистика про сервер OPC UA.

Systems	
Alarm Pipelines	0 active
EAM Role	Unknown
Modules	23 installed
Performance	5% CPU   735mb
Redundancy	Not configured
Reports	0 scheduled
SFCs	0 running
Tags	10 730 tags

Рисунок 2.19 – Елемент Systems на вкладці Status









Connections	
 <b>Databases</b> 2 / 2 connected	 <b>Designer Sessions</b> 1 open
 <b>Devices</b> 1 enabled	 <b>Gateway Network Connections</b> 0 / 0 connections
 <b>OPC Connections</b> 2 / 6 connected	 <b>Perspective Sessions</b> 0 open
 <b>Store &amp; Forward</b> 0 stores quarantined	 <b>Vision Clients</b> 0 open

Рисунок 2.20 – Елемент Connections на вкладці Status

У розділі Конфігурація реалізовано все налаштування шлюзу/платформи. Можливо виконувати наступні дії:

- створення нових проєктів, з'єднань з базою даних та підключень до OPC-серверів;
- налаштування продуктивності та історичних SQLTags;
- управління користувачами та їх ролями.

Способи запуску клієнтів: віконний та повноекранний режими. Режим може бути вибраний з випадаючого списку поряд з ім'ям проєкту. Натиснувши на назву проєкту він буде запуснений в режимі за замовчуванням Налаштування для кожного проєкту окремо в його дизайнері.

Віконний режим є стандартним методом запуску. Клієнт буде запуснений з використанням Java Web-Start і буде мати своє власне вікно. У цьому режимі, він буде працювати в якості повноправного, самостійного застосунку. Після запуску браузер може бути закритий і проєкт може в майбутньому запускатися за допомогою ярлика на робочому столі, який автоматично створюється при першому запуску клієнтського вікна.

Повноекранний режим запуску, схожий на віконний режим, також буде використовувати веб-запуск для запуску клієнта в повному обсязі, як самостійного застосунку. Проте, у цьому режимі клієнт буде займати весь екран, і не матиме заголовку. Цей режим ідеально підходить для сенсорних панелей та інших дисплеїв, де проєкт Ignition буде єдиним актуальним вікном на екрані.

Для даної SCADA-системи є можливість запуску клієнтського застосунку на мобільному пристрої – смартфоні чи планшеті. Але для цього повинен бути встановлений модуль Mobile Module, який не входить в стандартний набір модулів системи. Все що потрібно зробити користувачеві, щоб запуснити мобільний клієнт – підключити свій мобільний пристрій до бездротової мережі і вказати у полі адреси у веб-браузері адресу шлюзу. В результаті буде відкрито мобільну версію

сторінки веб-інтерфейсу, де можна буде запустити клієнтський застосунок на мобільному пристрої. Важливим моментом є те, що проекти повинні мати принаймні одне вікно і в їх налаштуваннях має бути включений запуск мобільної версії для того, щоб такий варіант було показано в списку для запуску. Після вибору проекту і входу в систему проект працює як і на звичайному ПК.

В усіх системах диспетчеризації присутній деякий функціонал, який базується на деяких алгоритмах і комунікації пристроїв в системі. Відповідно такий функціонал повинен бути реалізований за допомогою скриптів. В цій SCADA-системі типи скриптів, що виконуються на сервері або на клієнтському застосунку.

Між ними є принципова різниця, так як скрипти серверної частини виконуються завжди, поки функціонує шлюз Ignition. Клієнтські працюють лише при запусненому клієнті.

В Ignition як скриптова мова використовується Python. Вона була обрана із поглядів швидкодії, так як є дуже швидкою та використовує мінімум ресурсів. Крім того, ця мова може співпрацювати із Java-платформою, що теж дуже важливо, виходячи із того, чого вся SCADA-система написана на ній. В свою чергу це дає неймовірно широкі можливості для роботи і для створення алгоритмів функціонування системи диспетчеризації за рахунок підключення будь-яких Java бібліотек. Таким чином реалізується модифікація Python мови, яку називають Jython.

Перевагою цієї SCADA-системі є швидкодія, так як ПК призначений для роботи на базі операційної системи і його технічні характеристики є досить високими для виконання таких завдань, як виконання коду, написаного на високорівневій мові програмування. Таким чином, SCADA-система стає посередником в комунікації системи автоматизації. Це непогана альтернатива, тим більше якщо система велика і в будь-якому випадку має використовувати систему диспетчеризації.

Недоліком є залежність нижнього рівня системи автоматизації від

верхнього рівня. Це не дуже зручно і дещо ненадійно з огляду на принцип побудови операційних систем для комп'ютерів та ОС реального часу. До останніх висуваються значно вищі вимоги до надійності та тривалості безперебійної роботи, тоді як ОС для ПК працюють з великою кількістю програмного забезпечення, що може призвести до деяких критичних помилок, а отже і виходу з ладу системи. Звісно, ОС буде перезавантажено, але за цей час система автоматизації не буде функціонувати, що може нанести шкоду проходженню технологічних процесів різного характеру, в залежності від того, які функції покладені на SCADA-систему та від масштабів технологічного процесу, яким керує система автоматизації.

## 2.6 Опис PostgreSQL

PostgreSQL – це об'єктно-реляційна система управління базами даних, яка була розроблена в Науковому Комп'ютерному Департаменті Берклі Каліфорнійського Університету. POSTGRES є першопрохідником у багатьох аспектах, які стали доступні в деяких комерційних СУБД набагато пізніше.

PostgreSQL – це продукт із відкритим вихідним кодом, який є нащадком оригінального коду, написаного в Берклі. PostgreSQL підтримує більшу частину стандарту SQL і пропонує безліч сучасних можливостей:

- комплексні запити;
- зовнішні ключі;
- тригери;
- уявлення;
- транзакційна цілісність;
- різноманітність управління паралельним доступом.

Також можливості PostgreSQL можуть бути розширені за потребою користувача шляхом додавання нових:

- типів даних;

- функцій;
- операторів;
- агрегатних функцій;
- індексних методів;
- процедурних мов.

На базі PostgreSQL компанією EnterpriseDB створені потужніші варіанти цієї СУБД – PostgresPlus (складається з продуктів із відкритими початковими кодами) та PostgresPlusAdvanced Server (розширення PostgreSQL спеціальними можливостями для забезпечення сумісності з OracleDatabase). У комплекті поставки цих товарів міститься набір програмного забезпечення для розробників:

- PostgresStudio – більш потужний аналог pgAdmin;
- MigrationStudio – інструмент для автоматичного перетворення БД з MySQL/Oracle на PostgreSQL.

На рисунку 2.21 зображено структуру підключеного серверу створеної бази даних.

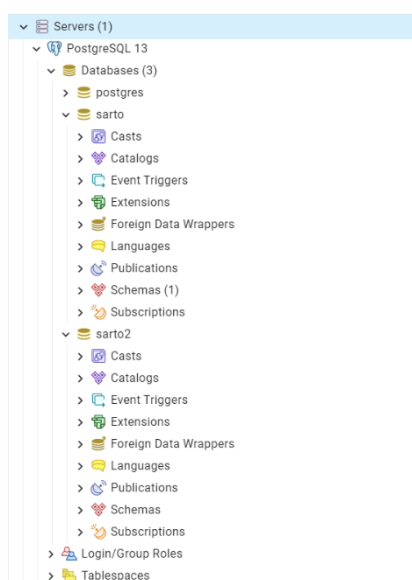


Рисунок 2.21 – Сервер PostgreSQL 13 на платформі pgAdmin 4

## 3 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

### 3.1 Розробка нижнього рівня

Для реалізації нижнього рівня програмного комплексу на базі середовища програмування Freelance Engineering потрібно створити апаратну структуру. Вона виглядає наступним чином, як зображено на рисунку 3.1:

- резервований контролер серії AC900FR, в склад якого входить програмований логічний контролер типу PM 902F та модуль зв'язку на основі Profibus протоколу типу CI 930F;
- дубльований інтерфейс зв'язку з програмованою шиною опитування модулів введення-виведення типу CI840A;
- модулі введення-виведення дискретного та аналогового типів DI814, DO814, AI810, AO810 у кількості по одному екземпляру на тип та модулі аналогового типу AI825 у кількості п'яти екземплярів.

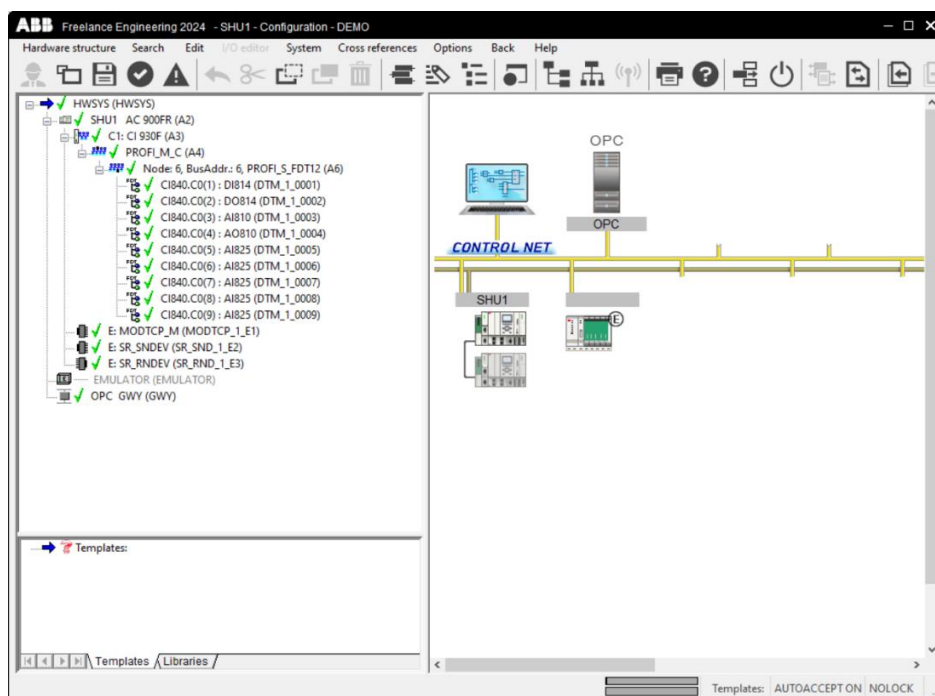
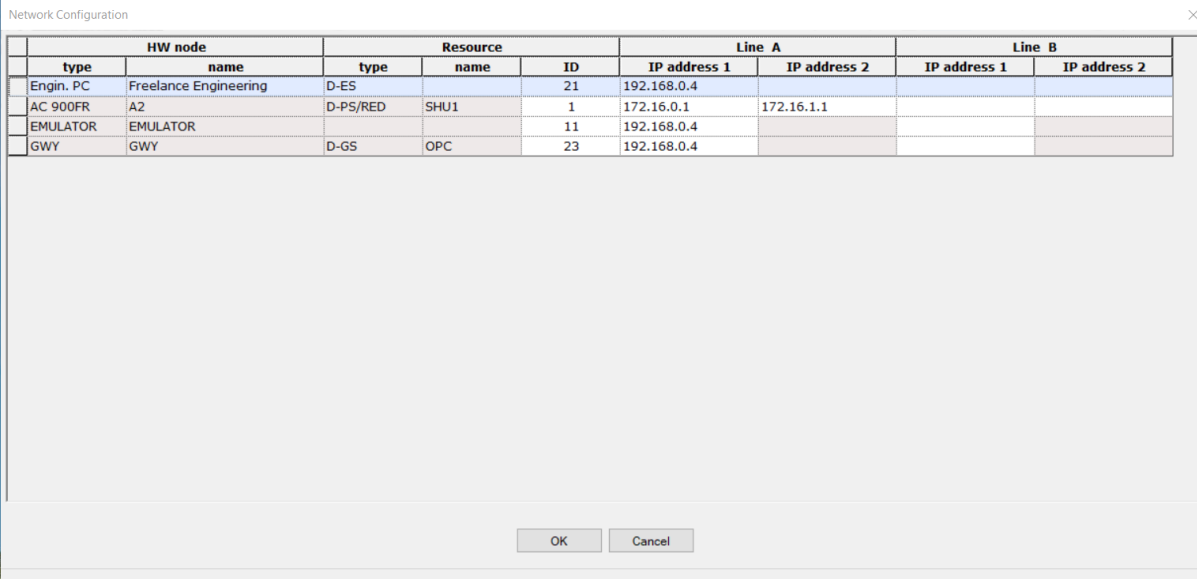


Рисунок 3.1 – Апаратна частина проекту

В апаратній частині реалізована технологія OPC-сервера, який отримує дані з контролера, що формуються шляхом виконання алгоритмів функціонального програмного забезпечення на основі сигналів, отриманих на модулі введення-виведення з польових пристроїв. OPC-сервер реалізує функції читання та запису змінних, що надходять зі SCADA-системі, яка виступає у ролі клієнтського підключення до серверу. Також, реалізована можливість застосування програмного емулятора контролера, що адаптується під резервовану структуру контролерів.

У розділі налаштувань мережевих зв'язків потрібно прописати IP-адреси та ідентифікаційні номери, задіяних апаратно-програмованих ресурсів проекту, таких як контролери, емулятори та OPC-сервера, як зображено на рисунку 3.2



HW node		Resource		Line A		Line B		
type	name	type	name	ID	IP address 1	IP address 2	IP address 1	IP address 2
Engin. PC	Freelance Engineering	D-ES		21	192.168.0.4			
AC 900FR	A2	D-PS/RED	SHU1	1	172.16.0.1	172.16.1.1		
EMULATOR	EMULATOR			11	192.168.0.4			
GWY	GWY	D-GS	OPC	23	192.168.0.4			

Рисунок 3.2 – Налаштування мережевих зв'язків

Завдяки web-ресурс, що входить у склад програмного забезпечення розробника у середовища, можна отримати інформаційні дані стан роботи контролера для належного контролю процесу виконання завдання та адаптації алгоритмів та циклів їх проходження відповідно до наявних можливостей обладнання, як зображено на рисунках 3.3 та 3.4.

The screenshot shows the ABB AC 900F web interface. The browser address bar indicates the URL is 172.16.1.1. The page title is "AC 900F". The left sidebar contains navigation links: Home, Diagnosis, Controller Status, Boot Configuration, Process Values, SD Card Files, and Application Displays. The main content area is titled "Controller Status" and contains the following data:

Controller	
CPU Load	17 %
Redundancy Load:	0 %
Temperature:	24 °C

Memory Pools	
Configuration Memory (PRAM):	4000 KB
Free Space:	3951 KB (98%)
Data memory (RAM):	4160 KB
Free Space:	3988 KB (95%)
Volatile Memory (VRAM):	8192 KB
Free Space:	7251 KB (88%)

Рисунок 3.3 – Основной контролер

The screenshot shows the ABB AC 900F web interface. The browser address bar indicates the URL is 172.16.0.1. The page title is "AC 900F". The left sidebar contains navigation links: Home, Diagnosis, Controller Status, Boot Configuration, Process Values, SD Card Files, and Application Displays. The main content area is titled "Controller Status" and contains the following data:

Controller	
CPU Load	1 %
Redundancy Load:	0 %
Temperature:	23 °C

Memory Pools	
Configuration Memory (PRAM):	4000 KB
Free Space:	3951 KB (98%)
Data memory (RAM):	4160 KB
Free Space:	3988 KB (95%)
Volatile Memory (VRAM):	8192 KB
Free Space:	7261 KB (88%)

Рисунок 3.4 – Резервный контролер

Стандартні програми, що є основою функціонального програмного забезпечення, призначеного для реалізації керуючих та інформаційних функцій систем регулювання розроблені мовами стандарту IEC 61131-3: IL (Instruction List), ST (Structured Text) та FBD (Function Block Diagram), – у середовищі програмування Freelance Engineering для контролерів серії AC 900F. Вони є функціонально закінченими програмами, кожна з яких вирішує логічні або обчислювальні завдання щодо реалізації керуючих та інформаційних функцій.

Програма первинного контролю датчиків струму (KD\_AI) забезпечує контроль працездатності датчика струму шляхом контролю його показань на максимальне і мінімальне значення (таблиця 3.1).

Таблиця 3.1 – Вхідна та вихідна інформація програми KD\_AI

Ім'я	Тип даних	Тип зберігання	Опис	Розширений опис
IN	INT	IN	Вхідний аналоговий сигнал	Вхідний аналоговий сигнал (0-28494)
FF	BOOL	IN	Несправність	Ознака несправності аналогового сигналу, що надходить від AI
OUT	INT	OUT	Вихід	Вихідний аналоговий сигнал (0-32767)
AIF	BOOL	OUT	Несправність датчику струму	AIF = TRUE якщо $IN > IN + 6\%$ , або $IN < IN - 6\%$

Програма фільтрації аналогових сигналів (AF) забезпечує фільтрацію аналогових сигналів шляхом включення до тракту їх проходження аперіодичного ланки першого порядку (таблиця 3.2).

Таблиця 3.2 – Вхідна та вихідна інформація програми AF

Ім'я	Тип даних	Тип зберігання	Опис	Розширений опис
SBR	BOOL	IN	Скидання несправності датчика	При SBR = TRUE, значенню XRF надається значення XR.
XR	INT	IN	Вхідне значення параметра (0-32767)	
N	INT	IN	Кількість тактів за секунду	Кількість тактів виконання програми протягом 1 с.
TF	INT	IN	Постійна часу фільтру (0-3277)	При $TF \leq 1/N$ , значенню XRF присвоюється значення XR.
VDR	BOOL	IN	Виведення датчика на ремонт	При VDR = TRUE, значенню XRF надається значення XR.
AMX	INT	IN	Обмеження зміни вхідного сигналу	При зміні XR величину більше AMX (тобто. $XR - XRF > AMX$ ), зміна XRF обмежується значенням AMX.
XRF	INT	OUT	Фільтроване значення параметра (0-32767)	$XRF = XRF + 1 / N / TPF \times (XR - XRF)$

Програма корекції вхідного аналогового сигналу (KX\_C) забезпечує поправочне перетворення вхідного аналогового сигналу на «гарячу» шкалу.

Таблиця 3.3 – Вхідна та вихідна інформація програми KX\_C

Ім'я	Тип даних	Тип зберігання	Опис	Розширений опис
IN	INT	IN	Вхідний параметр	
K	INT	IN	Поправочний коефіцієнт	
C	INT	IN	Поправна константа	
OUT	INT	OUT	$OUT = K \cdot IN + C$	

Програма формування серії періодичних імпульсів заданої шпаруватості (PULSATOR). При появі true на вході STR на виході OUT починається формування безперервної послідовності імпульсів з активним рівнем true. Формування імпульсів припиняється, коли сигнал на вході

START набуває значення false. Тривалість вихідного імпульсу визначається на вході TU як кількість тактів виконання програми. Тривалість паузи між сусідніми імпульсами задається на вході TN як кількість тактів виконання програми. Таким чином, період проходження імпульсів становить  $(TU + TN)$  тактів виконання програми. Прогальність проходження імпульсів визначається як  $TU/(TU+TN)$  (таблиця 3.4).

Таблиця 3.4 – Вхідна та вихідна інформація програми PULSATOR

Ім'я	Тип даних	Тип зберігання	Опис	Розширений опис
STR	BOOL	IN		Запуск пульсатора (true)
TU	INT	IN		Тривалість імпульсу (у тактах виконання програми)
TN	INT	IN		Тривалість паузи між імпульсами (у тактах виконання програми)
OUT	BOOL	OUT		Серія імпульсів заданої шпаруватості

Порядок включення та виконання стандартних програм та типових структур, що визначено відповідного до технологічного процесу, який розглядається у випадку цієї роботи, зображено на рисунку 3.5.

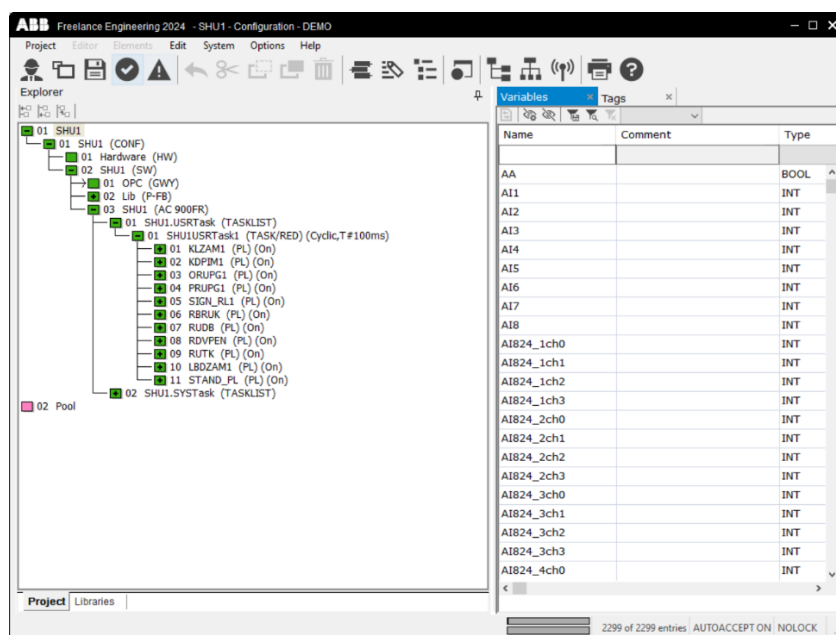


Рисунок 3.5 – Структура програмних алгоритмів

### 3.2 Розробка верхнього рівня

Створення окремих елементів технологічного процесу в Ignition Designer відбувається у розділі Template. Як готову функцію елемент Template можливо використовувати в різних створених екземплярах Window.

Для інформативного наповнення та підвищення швидкості та зручності аналізу роботи цього конкретного елемента технологічного процесу до шаблону треба додати елемент Numeric Label, як строку, що відображає граничне значення датчику контролю на резервуарі, та елемент базової засувки для відображення положення клапану в контексті з іншою інформацією можливо детально визначити та корегувати стан резервуару та відповідність рівня тиску в ньому нормальному діапазоні роботи. На рисунку 3.6 зображено приклад створеного кадру мнемосхеми.

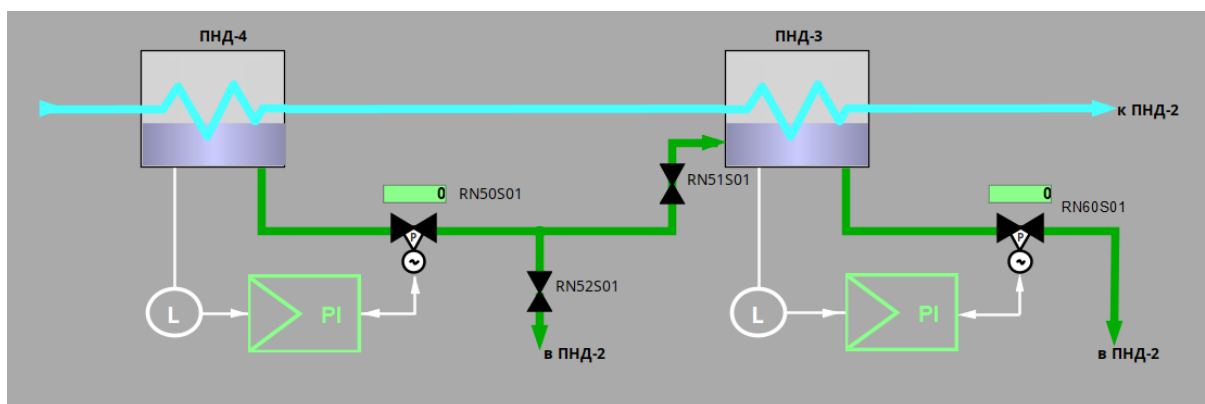


Рисунок 3.6 – Кадр мнемосхеми

Динамічні значення поточного стану роботи резервуара, що збираються датчиками контролю, які отримують інформацію від польових пристроїв, обробляються командами запрограмованого контролера та передаються на верхній рівень у SCADA-систему OPC-сервер. Резервуар має робочі діапазони, що повинні корегуватися відповідним положеннями

клапану для дотримування рівню пару у взаємопов'язаних елементах технологічного процесу. Діапазони визначаються технічними характеристиками індивідуального екземпляру обладнання залежно від складності виконання навантажень. На рисунках 3.7 та 3.8 зображено налаштування положень клапану за допомогою кольорового відображення.

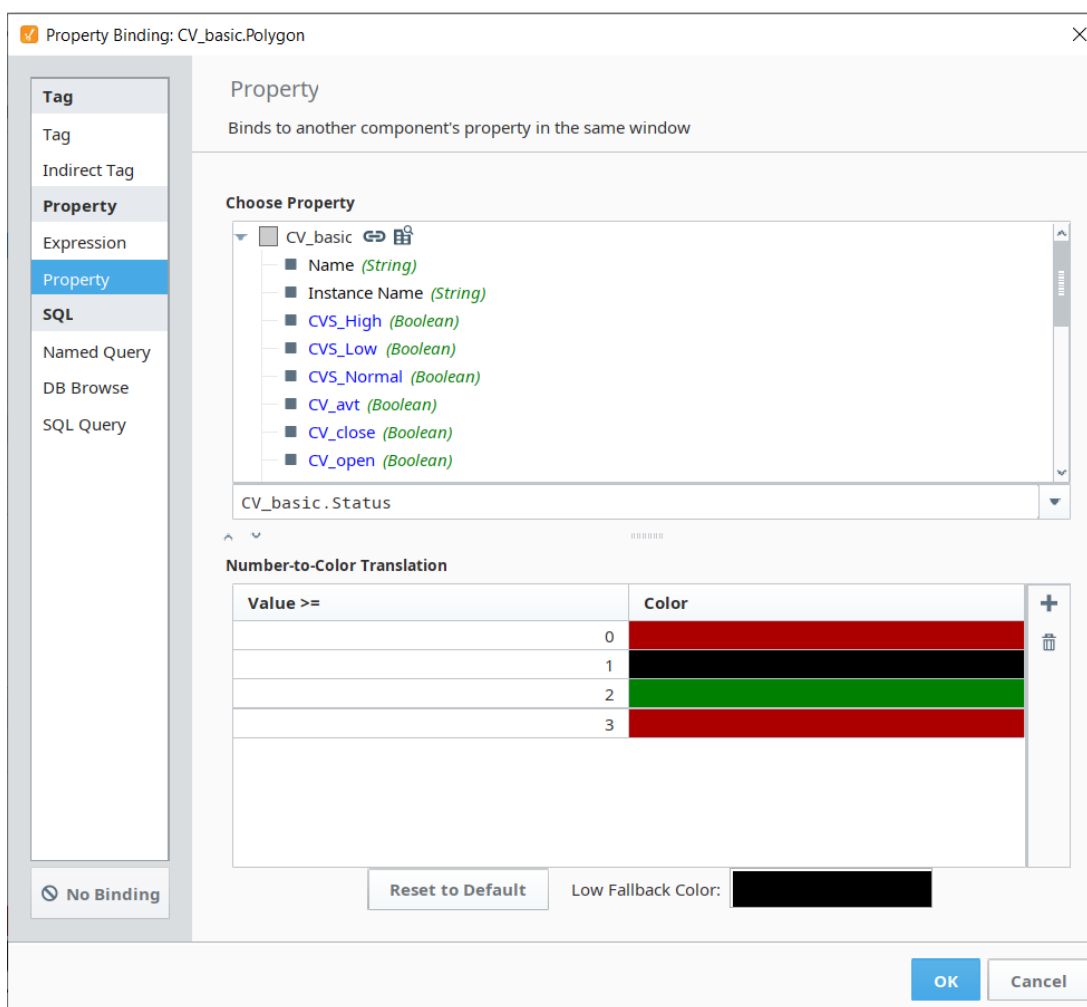


Рисунок 3.7 – Налаштування елемента відображення «Клапан» (частина 1)

На рисунку 3.9 зображено вкладку, на якій користувач має можливість експортувати або імпортувати дані, що містять коефіцієнти стану, які потрібні для відповідного налаштування режиму роботи системи, у вигляді таблиць. Значення цих коефіцієнтів зберігаються у підключеній базі даних.

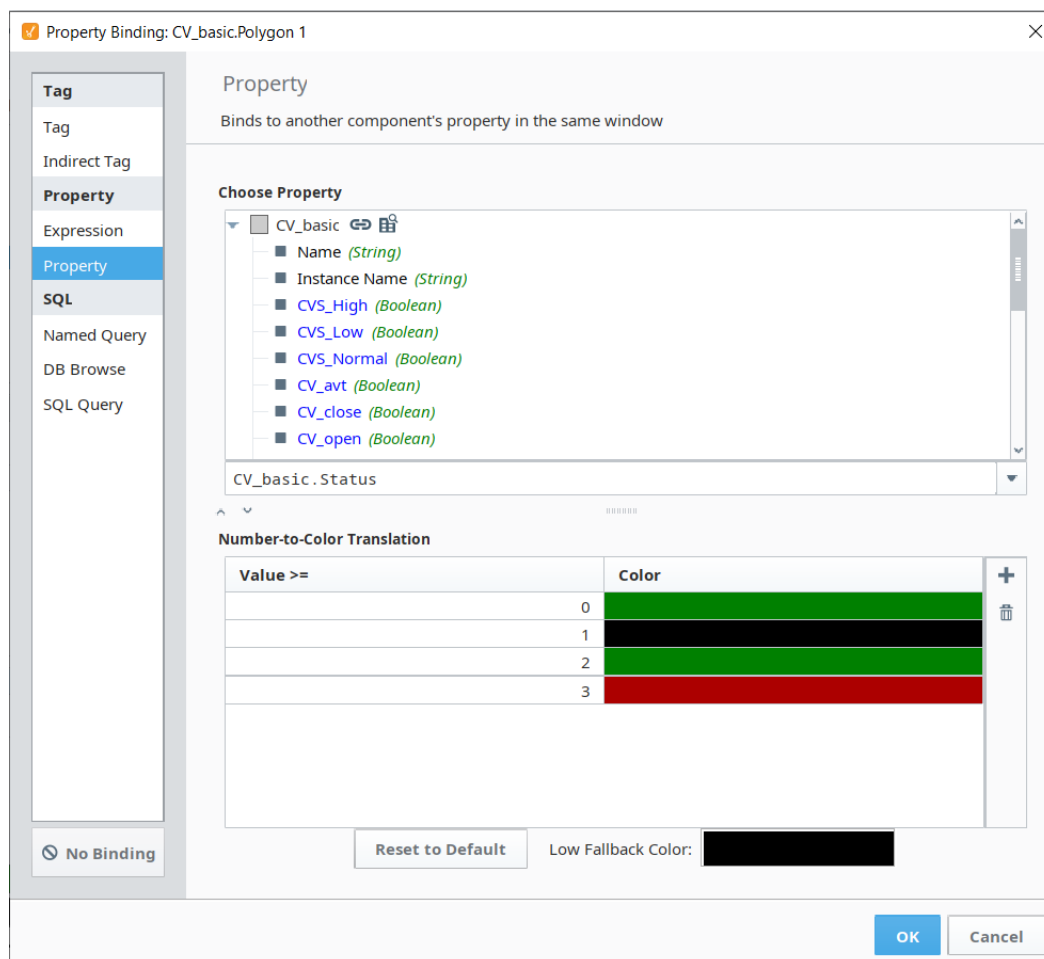


Рисунок 3.8 – Налаштування елемента відображення «Клапан» (частина 2)

SHU1	SHU2	SHU3	SHU4	Очистити таблицю	Записати у файл
Ідентифікатор	Опис	Значення	Повний шлях		
A2GCN1TF	К-т фільтр. к-та учит.кол.вкл.ГЦН	35	[default]SHU1/KO/A2GCN1TF		
BRUKDDZ	Величина корекції дозакрyття	0,5	[default]SHU1/KO/BRUKDDZ		
BRUKDKS	Величина корекції стат. помилки	1	[default]SHU1/KO/BRUKDKS		
BRUKDPDZ	DP для включення дозакрyття БРУ-К	0,4	[default]SHU1/KO/BRUKDPDZ		
BRUKDZMAX	Огр. поправки дозакрyття БРУ-К	4	[default]SHU1/KO/BRUKDZMAX		
BRUKKO1	Коефiц. ОС при DN	0,13	[default]SHU1/KO/BRUKKO1		
BRUKKSR	Коефiц. ОС при положеннi PK	0,04	[default]SHU1/KO/BRUKKSR		
BRUKNERP	Нелектр для переходу P→П по кн.	399,98	[default]SHU1/KO/BRUKNERP		
BRUKPZMAX	Мах значення Pзад	64	[default]SHU1/KO/BRUKPZMAX		
BRUKPZMIN	Min значення Pзад в реж. P	60	[default]SHU1/KO/BRUKPZMIN		
BRUKSDZ	S БРУ-К для включення дозакрyття	25	[default]SHU1/KO/BRUKSDZ		
BRUKSRP	S БРУ-К для переходу P→П	40	[default]SHU1/KO/BRUKSRP		
BRUKSTIPZ	Шаг изменения Pзад по кнопкам	0,1	[default]SHU1/KO/BRUKSTIPZ		
BRUKTDZ	Задержка T при дозакрyтти	20	[default]SHU1/KO/BRUKTDZ		
BRUKTKS	Задержка T при компенс. стат. ош.	60,01	[default]SHU1/KO/BRUKTKS		
BRUKTPR	Вудержка T переход P→P, 30 в P	15	[default]SHU1/KO/BRUKTPR		

Рисунок 3.9 – Вікно налаштувань коефіцієнтів

На вкладці відображено наступні елементи:

- інформаційна таблиця, що містить ідентифікатори, описи, значення та повний шлях знаходження у базі даних;
- контейнер вибору відповідного режиму;
- функціональні кнопки імпорту, експорту коефіцієнтів у вигляді Excel-таблиць та очищення інформації.

Функціональна кнопка експорту виконує експорт таблиці архівних даних у файлову систему з підключеного серверу бази даних у Excel-форматі (рисунок 3.10).

Алгоритм процесу експорту:

- імпорт необхідних бібліотек – для роботи з Excel-файлами, для обробки дат, для відкриття файлів та для роботи з датами у Java;
- визначення вхідних параметрів – шлях до Excel-файлу, признак, чи є в першому рядку заголовки, номер листа (за замовчуванням перший лист), визначення межі діапазону для читання даних (по рядках і стовпцях);
- читання файлу;
- визначення меж діапазону даних – якщо перша та остання строки не задані, то вони беруться автоматично з початку і кінця листа, якщо перший та останній стовбці не задані, вони обчислюються для першого рядка на основі кількості стовпців;
- читання рядків та стовпців – для кожної клітинки в рядку визначається тип даних (число, рядок, дата, булеве значення тощо), і відповідно зчитується значення клітинки;
- побудова результату – для кожного рядка створюється список значень клітинок;
- закриття потоку;
- перетворення даних у табличний вигляд.

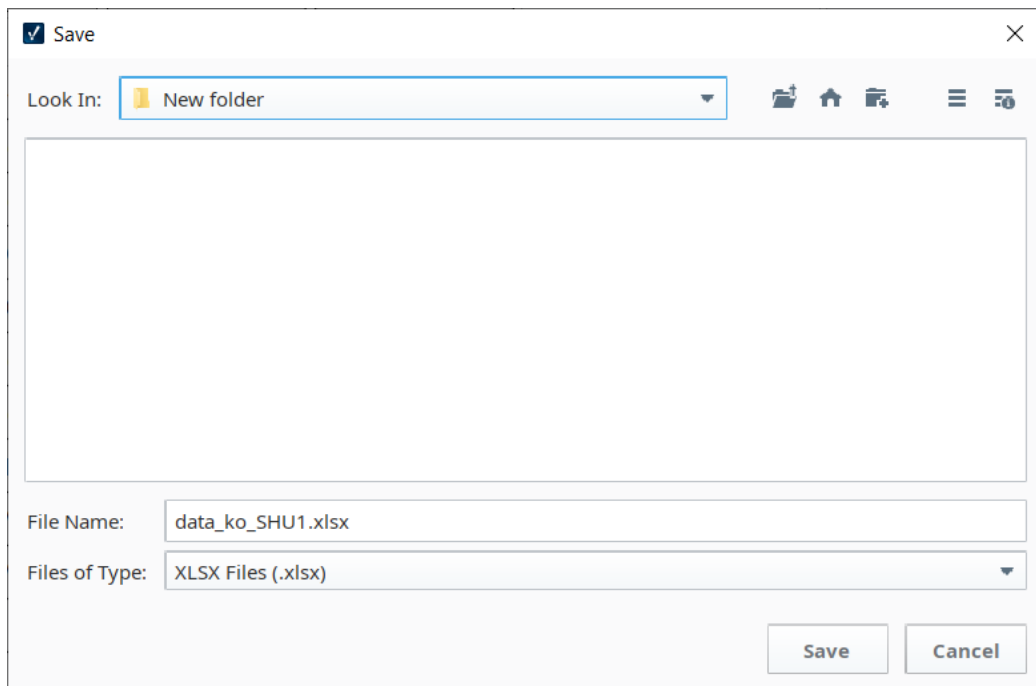


Рисунок 3.10 – Вікно файлового браузеру для виконання експорту

Опис алгоритму ініціалізації даних у табличному вигляді:

- ініціалізація стовпців та створення порожнього датасету;
- очищення існуючого датасету;
- обробка інформації від тегів та їх конфігурація (назва, опис, значення, шлях, формат);
- створення рядків, на основі даних від кожного тегу;
- наповнення датасету;
- оформлення датасету до відповідного формату.

### 3.3 Впровадження алгоритмів машинного навчання

На основі дослідження ефективності застосувань методів машинного навчання для задачі прогнозування технічного обслуговування були реалізовані дві моделі: Random Forest – як приклад класичного ансамблевого методу, та LSTM – як представник рекурентних нейромереж для роботи з часовими рядами (рисунки 3.11 та 3.12).

Було використано отримані з SCADA-архівів часові ряди параметрів обладнання (температура трансформатора, вібрації генератора, напруга на вводах тощо) з розміткою технічного стану в момент часу. Цільова змінна – значення контрольного параметра через визначений період (наприклад, температура ізоляції через 24 години).

Модель Random Forest була навчена на попередньо нормалізованих ознаках з використанням RandomForestRegressor. Для валідації було застосовано TimeSeriesSplit (3 фолди), що дозволяє враховувати часову залежність даних. Для підбору параметрів було реалізовано GridSearchCV.

Для LSTM модель будувалася з використанням Keras. Оскільки TimeSeriesSplit не сумісний з генераторами, було реалізовано ручну крос-валідацію, для кожного фолду створювались окремі TimeseriesGenerator для навчання та валідації.

Оцінювання виконувалось за метрикою MAE (mean absolute error) на тестових фолдах. У таблиці 3.5 приведено порівняння середніх значень для кожної моделі (рисунок 3.13).

Таблиця 3.5 – Середні значення MAE для кожної моделі

Модель	MAE (середнє по фолдах)
Random Forest	19.9
LSTM	15.0

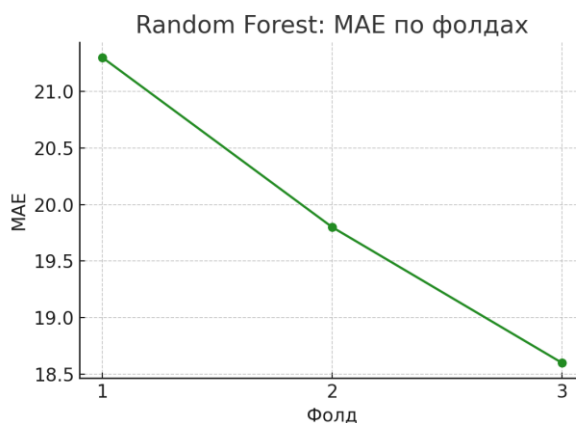


Рисунок 3.11 – Random Forest: MAE по фолдах

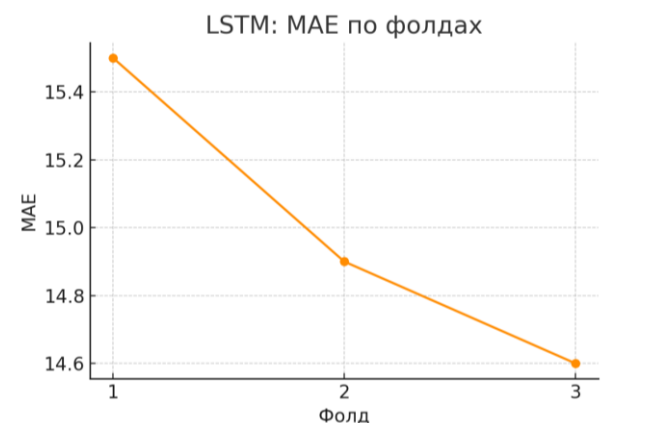


Рисунок 3.12 – LSTM: MAE по фолдах

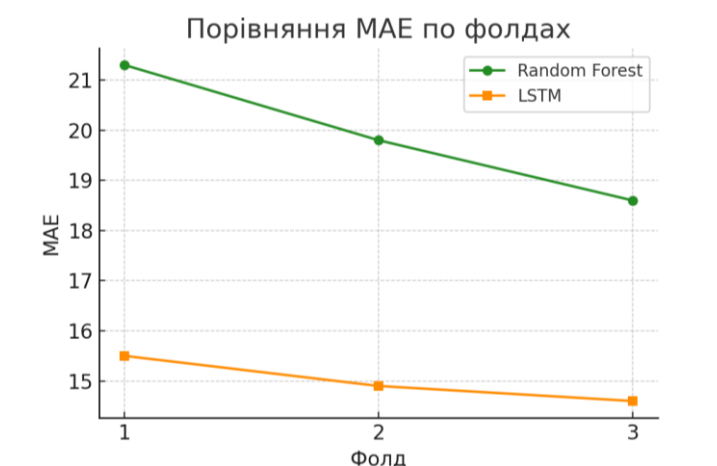


Рисунок 3.13 – Порівняння MAE по фолдах

Згідно результатів, отриманих у ході проведення експериментальних впроваджень обраних моделей машинного навчання у реалізованій програмній комплекс, що забезпечує відповідний процес технічного обслуговування, можна зробити наступні припущення:

- Random Forest показав стабільні, хоч і дещо вищі помилки, є зручним для швидкого аналізу та інтерпретації;
- LSTM виявилась більш точною, проте потребує більше ресурсів та ретельної підготовки даних;
- застосування TimeSeriesSplit дозволяє адекватно оцінювати моделі при роботі з часовими рядами.

## ВИСНОВКИ

Під час виконання кваліфікаційної роботи було розроблено програмний комплекс з елементами алгоритмів машинного навчання для роботи на автоматизованому виробництві. В якості програмного середовища було обрано SCADA-систему Ignition, розроблену на мові Java, що дає можливість для роботи із будь-якими модулями вихідною мови. Це розширює програмні можливості самої SCADA-системи і робить її потужним інструментом в роботі зі системи автоматизації технологічних процесів.

В першому розділі була визначена наукова проблематика кваліфікаційної роботи та було досліджено сучасний стан використання методів машинного навчання, що використовуються для технічного обслуговування обладнання на промисловому виробництві. Було детально досліджено проблематику використання нейронних мереж та алгоритмів кластеризації. Також було досліджено результати впровадження алгоритмів глибинного навчання для можливості належного контролю та реагування на несправності або збої роботи обладнання під час технічного обслуговування аномалій на промисловому виробництві.

В другому розділі було розглянуто та обрано оптимальні алгоритми для впровадження у програмний комплекс, програмне середовище для розробки верхнього та нижнього рівнів комплексу, сервер бази даних для зберігання інформації та комунікаційний канал зв'язку для коректного функціонування системи.

В третьому розділі було описано можливість використання програмного комплексу автоматизації з впровадженням алгоритмом машинного навчання на прикладі технологічного процесу.

Перевагами кваліфікаційної роботи можна вважати, що комплекс орієнтований на використання в автоматизованих системах управління

технологічним процесом, та надає можливість проводити контроль технологічного обслуговування обладнання.

Перспективами розвитку напрямку, що був розглянутий у кваліфікаційній роботі, вважається пошук шаблонів для обробки даних для підвищення ефективності застосування у промислових галузях та проведення технологічного обслуговування обладнання.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Як працює machine learning та його застосування на практиці. URL: <https://nachasi.com/tech/2019/01/31/yak-pratsyuye-machine-learning/> (дата звернення: 15.05.2025).
2. Машинне навчання простими словами. URL: <http://www.mmf.lnu.edu.ua/ar/1739> (дата звернення: 15.05.2025).
3. Industry 4.0 URL: <https://www.it.ua/knowledge-base/technology-innovation/industry-4> (дата звернення: 17.05.2025).
4. Четверта промислова революція: що це означає, як реагувати URL: <https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond> (дата звернення: 17.05.2025).
5. Що таке Індустрія 4.0?. URL: <https://www.ibm.com/topics/industry-4-0> (дата звернення: 17.05.2025).
6. Основні тенденції розвитку Індустрії 4.0 та її вплив на економічну безпеку держави: міжнародний аспект. URL: <http://snku.krok.edu.ua/index.php/vcheni-zapiski-universitetu-krok/article/view/569> (дата звернення: 20.05.2025).
7. Що таке Індустрія 4.0, Четверта промислова революція та 4IR? URL: <https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-are-industry-4-0-the-fourth-industrial-revolution-and-4ir> (дата звернення: 20.05.2025).
8. Цифрова трансформація виробничих компаній. URL: <https://www.sap.com/products/scm/industry-4-0/industry-4-0-strategy.html> (дата звернення: 21.05.2025).
9. Техніки та методи машинного навчання. URL: <https://addepto.com/blog/machine-learning-techniques-and-methods/> (дата звернення: 22.05.2025).
10. Методи машинного навчання. *Znanstvena misel journal №39/2020*. URL: <https://www.znanstvena-journal.com/wp->

<content/uploads/2020/10/Znanstvena-misel-journal-%E2%84%9639-2020-VOL.1.pdf> (дата звернення: 22.05.2025).

11. A Survey of Machine Learning in Industry 4.0: Applications, Challenges and Opportunities. URL: <https://www.sciencedirect.com/science/article/abs/pii/S2214785322075381> (дата звернення: 23.05.2025).

12. Огляд СУБД PostgreSQL. URL: <https://studfile.net/preview/1504673/> (дата звернення: 26.05.2025).

13. Огляд SCADA-системи Ignition. URL: <https://www.realpars.com/blog/ignition-scada> (дата звернення: 26.05.2025).

14. Огляд Freelance Engineering. URL: <https://new.abb.com/control-systems/essential-automation/freelance/system/freelance-overview> (дата звернення: 26.05.2025).

15. Огляд OPC-технологій. URL: <https://indusoft.com.ua/blog/2022/05/18/chomu-opc-ua-takij-vazhlij/> (дата звернення: 26.05.2025).