Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Навчально-науковий інститут комп'ютерних наук та штучного інтелекту
Кафедра комп'ютерних систем та робототехніки

До захисту допущено
Кафедрою комп'ютерних систем та робототехніки
протокол №__ від ____ червня 2026 р.

завідувач кафедри _____ Максим ХРУСЛОВ
(підпис)
«___»_____ 2026 р.

Кваліфікаційна робота

здобувача першого (бакалаврського) рівня вищої освіти

«СИСТЕМА РАННЬОГО ВИЯВЛЕННЯ НЕСПРАВНОСТЕЙ У ВИРОБНИЧИХ ПРОЦЕСАХ НА ОСНОВІ ШТУЧНОГО ІНТЕЛЕКТУ»

Спеціальність ***123 – Комп'ютерна інженерія***

Освітня програма ***Комп'ютерна інженерія***

Виконавець _____ **Олександр САВОЧКА**
(підпис)

Науковий керівник _____ **Ольга МОРОЗ**
(підпис)

Харків – 2026

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та 4 додатків. Загальний обсяг роботи становить 64 сторінки, з яких 46 сторінок основної частини, що містить 8 рисунків, 10 таблиць, 19 найменувань у списку використаних джерел.

Метою кваліфікаційної роботи є підвищення надійності функціонування промислового обладнання шляхом створення системи раннього виявлення відмов на основі аналізу вібраційно-температурних часових рядів із використанням методів глибокого навчання та практичної реалізації результатів у вигляді програмного веб-застосунку.

Об'єкт дослідження – процеси функціонування та технічного стану промислового обладнання, що характеризуються змінами вібраційних і температурних параметрів у часі.

Предмет дослідження – методи та моделі глибокого навчання для інтелектуального аналізу вібраційно-температурних часових рядів з метою раннього виявлення відмов обладнання.

Проблема, яка вирішується в роботі, полягає у своєчасному виявленні аномальних станів обладнання в умовах шумових, нестаціонарних і багатовимірних даних, що ускладнюють застосування традиційних методів діагностики, та потребують використання інтелектуальних алгоритмів аналізу.

Область застосування – промислові підприємства, системи моніторингу технічного стану обладнання, інформаційно-вимірювальні системи, SCADA та IIoT-середовища, а також освітній процес за спеціальністю 123 «Комп'ютерна інженерія».

Ключові слова: інформаційні технології, раннє виявлення відмов, вібраційні дані, часові ряди, глибоке навчання, CNN-LSTM, аналіз аномалій, прогнозне обслуговування, веб-застосунок, Python, машинне навчання.

ABSTRACT

The explanatory note to the qualification work consists of an introduction, four chapters, conclusions, a list of references, and 4 appendices. The total volume of the work is 64 pages, including 46 pages of the main content with 8 figures, 10 tables, 19 references.

The aim of the qualification work is to increase the reliability of industrial equipment operation by creating a system for early fault detection based on the analysis of vibration and temperature time series using deep learning methods and the practical implementation of the results in the form of a web application.

The object of research is the processes of operation and technical condition of industrial equipment characterized by changes in vibration and temperature parameters over time.

The subject of research is deep learning methods and models for intelligent analysis of vibration and temperature time series aimed at early fault detection of equipment.

The problem addressed in this work is the timely detection of abnormal equipment conditions in the presence of noisy, non-stationary, and multidimensional data that complicate the use of traditional diagnostic methods and require the application of intelligent data analysis algorithms.

The field of application includes industrial enterprises, equipment condition monitoring systems, information and measurement systems, SCADA and IIoT environments, as well as the educational process in the specialty 123 "Computer Engineering".

Keywords: information technology, early fault detection, vibration data, time series, deep learning, CNN-LSTM, anomaly detection, predictive maintenance, web application, Python, machine learning.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ..... 6

ВСТУП..... 7

РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ ПРОГНОЗНОГО ОБСЛУГОВУВАННЯ ТА МОНІТОРИНГУ СТАНУ ОБЛАДНАННЯ..... 11

1.1 Сучасні концепції та стратегії технічного обслуговування (CBM, PdM, Predictive Maintenance)..... 11

1.2 Основні причини відмов промислового обладнання та їх попереджувальні ознаки..... 12

1.3 Вібраційний та температурний моніторинг як джерело діагностичної інформації..... 13

1.4 Методи раннього виявлення аномалій у часових рядах..... 14

1.5 Порівняння моделей машинного та глибокого навчання для прогнозування відмов..... 15

1.6 Порівняльний аналіз існуючих рішень та систем прогнозного обслуговування..... 17

Висновок до розділу 1..... 19

РОЗДІЛ 2 АНАЛІЗ ТА ПІДГОТОВКА ВХІДНИХ ДАНИХ..... 20

2.1 Джерела даних: промислові датчики вібрації та температури..... 20

2.2 Опис використаних наборів даних..... 22

2.3 Характеристики часових рядів вібрації та температури..... 23

2.4 Попередня обробка даних (очищення, фільтрація шумів, нормалізація, масштабування)..... 25

2.5 Інженерія ознак (Feature Engineering): статистичні, частотні (FFT), вейвлет-перетворення, ентропія тощо..... 26

2.6 Обробка незбалансованих класів та визначення моменту відмови..... 27

Висновки за розділом 2..... 28

РОЗДІЛ 3 РОЗРОБКА ТА ПОБУДОВА МОДЕЛІ ПРОГНОЗУВАННЯ 30

3.1 Вибір архітектури моделі прогнозування 30

3.2 Проєктування структури нейронної мережі 32

3.3 Навчання та валідація моделі 35

Висновок за розділом 3 37

РОЗДІЛ 4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНІ РЕЗУЛЬТАТИ 38

4.1 Архітектура програмного веб-застосунку 38

4.2 Опис програмної реалізації 39

4.3 Організація експерименту 40

4.4 Функціональне тестування програмної системи 42

4.5 Оцінювання якості моделі 45

Висновки за розділом 4 46

ВИСНОВКИ 47

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 49

ДОДАТКИ 51

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

AE	–	Autoencoder (автоенкодер)
AI	–	Artificial Intelligence (штучний інтелект)
ARIMA	–	AutoRegressive Integrated Moving Average
CBM	–	Condition-Based Maintenance (обслуговування за фактичним станом)
CNN	–	Convolutional Neural Network (згорткова нейронна мережа)
DAQ	–	Data Acquisition System (система збору даних)
DL	–	Deep Learning (глибоке навчання)
FFT	–	Fast Fourier Transform (швидке перетворення Фур'є)
GRU	–	Gated Recurrent Unit (рекурентна нейронна мережа)
IIoT	–	Industrial Internet of Things (промисловий інтернет речей)
IoT	–	Internet of Things (інтернет речей)
LSTM	–	Long Short-Term Memory (рекурентна нейронна мережа довготривалої пам'яті)
MAE	–	Mean Absolute Error (середня абсолютна помилка)
MES	–	Manufacturing Execution System (система управління виробництвом)
ML	–	Machine Learning (машинне навчання)
OEE	–	Overall Equipment Effectiveness (загальна ефективність обладнання)
PdM	–	Predictive Maintenance (прогнозне технічне обслуговування)
PLC	–	Programmable Logic Controller (програмований логічний контролер)
RMS	–	Root Mean Square (середньоквадратичне значення сигналу)
RMSE	–	Root Mean Square Error (середньоквадратична помилка)
RUL	–	Remaining Useful Life (залишковий ресурс обладнання)
SCADA	–	Supervisory Control and Data Acquisition
SVM	–	Support Vector Machine (метод опорних векторів)
TCN	–	Temporal Convolutional Network
XAI	–	Explainable Artificial Intelligence (інтерпретований штучний інтелект)

ВСТУП

Сучасне промислове виробництво характеризується високим рівнем автоматизації, складністю технологічних процесів та значною вартістю обладнання. Незаплановані зупинки через раптові відмови обладнання призводять до суттєвих економічних втрат, пов'язаних із простоями, аварійним ремонтом, недоотриманим прибутком та, у деяких випадках, загрозою безпеці персоналу й довкіллю.

Актуальність роботи. За даними досліджень McKinsey та General Electric [10], незаплановані простої промислових підприємств у середньому становлять 5–20% від загального фонду робочого часу, а впровадження систем прогнозного обслуговування (Predictive Maintenance) дозволяє скоротити витрати на технічне обслуговування на 25–30% та зменшити кількість аварійних зупинок на 70–75% [16]. Особливої актуальності ця задача набуває в умовах цифровізації виробництва та впровадження концепцій Industry 4.0, де значна роль відводиться інтелектуальному аналізу даних сенсорів для підвищення надійності обладнання.

Найефективнішими попереджувальними ознаками наближення відмови більшості обертового та вібраційно-навантаженого обладнання (електродвигуни, насоси, компресори, редуктори, верстати з ЧПК) є зміни характеру вібрації та температурного режиму. Сучасні промислові системи оснащуються великою кількістю датчиків, що генерують часові ряди вібраційно-температурних даних у реальному часі. Розробка автоматизованих моделей раннього виявлення аномалій на основі цих даних є одним із ключових напрямів Industry 4.0 та IIoT (Industrial Internet of Things) [3].

Таким чином, створення ефективної моделі прогнозування відмов обладнання з використанням методів машинного та глибокого навчання на основі багатофакторних вібраційно-температурних часових рядів є актуальним науково-прикладним завданням, що має значний економічний і технологічний потенціал.

Таким чином, виникає необхідність розроблення інтелектуальних підходів до аналізу вібраційно-температурних даних, які дозволяють виявляти початкові ознаки відмов на ранніх етапах.

Мета роботи – підвищення надійності функціонування промислового обладнання шляхом створення системи раннього виявлення відмов на основі аналізу вібраційно-температурних часових рядів із використанням методів глибокого навчання та її практичного впровадження у вигляді програмного веб-застосунку.

Досягнення поставленої мети потребує визначення об'єкта та предмета дослідження.

Об'єкт дослідження – процеси функціонування та технічного стану промислового обертового обладнання, що характеризуються змінами вібраційних і температурних параметрів у часі.

Предмет дослідження – методи глибокого навчання та алгоритми обробки часових рядів для раннього виявлення відмов промислового обладнання.

Тематика роботи безпосередньо відповідає змісту підготовки фахівців за спеціальністю 123 «Комп'ютерна інженерія», оскільки передбачає використання методів обробки даних, машинного навчання, розроблення програмних систем та інтеграцію інтелектуальних алгоритмів у прикладні інформаційні середовища.

Для досягнення поставленої мети в роботі сформульовано такі задачі:

1. Проаналізувати сучасні підходи до забезпечення надійності промислового обладнання та методи раннього виявлення відмов на основі аналізу даних технічного стану.
2. Дослідити структуру та властивості вібраційно-температурних часових рядів, що характеризують процеси деградації обладнання.
3. Виконати підготовку даних, зокрема очищення, нормалізацію, синхронізацію та формування навчальних вибірок.
4. Сформувати інформативний простір ознак на основі статистичних, спектральних та часово-частотних характеристик сигналів.

5. Обґрунтувати вибір та побудувати модель глибокого навчання для раннього виявлення відмов промислового обладнання.

6. Провести навчання та оцінювання ефективності моделі з використанням відповідних метрик якості.

7. Забезпечити практичне застосування результатів шляхом створення програмного веб-застосунку для аналізу даних і виявлення відмов.

Методи дослідження: методи аналізу часових рядів, статистична обробка даних, інженерія ознак, глибоке навчання (LSTM-мережі, GRU, 1D-CNN, автоенкодери), веб-розробка (FastAPI), порівняльний аналіз моделей за метриками точності, чутливості та швидкодії.

Наукова новизна одержаних результатів полягає у:

– удосконаленні підходу до раннього виявлення відмов промислового обладнання за рахунок комплексного використання вібраційних і температурних даних як взаємодоповнюючих джерел інформації;

– подальшому розвитку методів аналізу багатовимірних часових рядів шляхом застосування глибоких нейронних мереж для виявлення прихованих деградаційних процесів;

– узагальненні та адаптації сучасних моделей глибокого навчання до задачі раннього виявлення відмов із урахуванням особливостей промислових даних (шум, нестаціонарність, незбалансованість класів).

Робота безпосередньо пов’язана зі спеціальністю 123 «Комп’ютерна інженерія», оскільки передбачає розробку програмно-апаратного рішення для обробки та аналізу даних, отриманих із промислових сенсорів, із використанням сучасних інформаційних технологій, методів машинного навчання та архітектур розподілених систем. У межах дослідження розглядаються питання збору, передачі, обробки та інтелектуального аналізу даних у системах Industrial IoT, що є складовою сучасних комп’ютерно-інтегрованих систем.

Практичне значення одержаних результатів полягає в тому, що:

- запропоновані підходи до аналізу вібраційно-температурних часових рядів можуть бути використані для підвищення надійності та безперервності роботи промислового обладнання;
- розроблена модель раннього виявлення відмов забезпечує своєчасне виявлення аномальних станів, що дозволяє зменшити витрати на ремонт і уникнути аварійних зупинок;
- створений програмний веб-застосунок може бути використаний як інструмент підтримки прийняття рішень для технічного персоналу підприємств;
- результати роботи можуть бути інтегровані у системи моніторингу технічного стану (SCADA, IIoT) та використані для побудови інтелектуальних систем прогнозного обслуговування;
- розроблені рішення можуть бути застосовані в навчальному процесі за спеціальністю 123 «Комп'ютерна інженерія» при вивченні дисциплін, пов'язаних з аналізом даних, машинним навчанням та комп'ютерно-інтегрованими системами.

Структура кваліфікаційної роботи відповідає поставленій меті та задачам дослідження і включає чотири розділи, у яких послідовно розглянуто аналіз сучасних підходів, підготовку даних, побудову моделі та її програмну реалізацію з експериментальною перевіркою ефективності.

Структура роботи: робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. У першому розділі розглянуто теоретичні основи прогнозного обслуговування. Другий розділ присвячено аналізу та підготовці даних. У третьому розділі розроблено моделі глибокого навчання. Четвертий розділ містить опис програмної реалізації та результати експериментів.

РОЗДІЛ 1
ТЕОРЕТИЧНІ ОСНОВИ ПРОГНОЗНОГО ОБСЛУГОВУВАННЯ ТА МОНІТОРИНГУ СТАНУ ОБЛАДНАННЯ

1.1 Сучасні концепції та стратегії технічного обслуговування (CBM, PdM, Predictive Maintenance)

Технічне обслуговування та ремонт (TOiP) промислового обладнання еволюціонувало разом із розвитком технологій і вимогами виробництва. Спочатку домінувала стратегія реактивного обслуговування (Run-to-Failure), за якої ремонт виконували лише після відмови. Хоча вона була простою в реалізації, такий підхід спричиняв значні незаплановані простої й високі витрати на аварійні роботи [1], [2].

Згодом поширилось планово-попереджувальне обслуговування (Preventive Maintenance), що базується на фіксованих інтервалах профілактики незалежно від фактичного стану обладнання. Воно зменшило кількість раптових відмов, однак частина робіт виконувалась передчасно, що призводило до перевитрат ресурсів.

Із розвитком сенсорних технологій сформувалась концепція обслуговування за фактичним станом, яка передбачає моніторинг діагностичних параметрів і втручання лише за наявності відхилень. На її основі виникло прогнозне обслуговування (Predictive Maintenance, PdM) – ключовий елемент Industry 4.0, що дозволяє прогнозувати момент відмови або оцінювати залишковий ресурс обладнання за допомогою методів машинного навчання.

PdM – забезпечує істотне скорочення незапланованих простоїв і витрат на TOiP, подовжує термін служби обладнання та підвищує показник OEE. За оцінками Deloitte та General Electric, глобальний економічний ефект від його впровадження до 2030 року може перевищити 1.5 трлн доларів США.

1.2 Основні причини відмов промислового обладнання та їх попереджувальні ознаки

Відмови промислового обладнання зумовлені поєднанням експлуатаційних, конструктивних і зовнішніх факторів, що спричиняють поступову деградацію або раптовий вихід з ладу механічних, електричних і гідравлічних компонентів. Найпоширенішими причинами є механічний знос у підшипниках, шестернях і валах через тертя та навантаження, а також втома матеріалу під дією циклічних напружень, характерна для роторних систем турбін і компресорів [17]. Перегрів унаслідок перевантаження, недостатнього охолодження чи дефіциту змащення призводить до зниження міцності та термічних відмов двигунів і трансформаторів. Корозія й окислення в агресивних середовищах послаблюють конструкції трубопроводів, насосів і резервуарів.

Суттєву роль відіграють також помилки монтажу, невирівняність і дисбаланс роторів, що генерують надмірні вібрації та прискорений знос. Електричні відмови виникають через проблеми ізоляції, короткі замикання чи перевантаження, тоді як гідравлічні та пневматичні системи страждають від витоків і забруднення робочого середовища. Додатковими чинниками є перепади напруги, зовнішні вібрації та людські помилки.

Попереджувальні ознаки проявляються у зміні робочих параметрів: зростанні амплітуди та дисперсії вібрацій, появі характерних частотних піків, підвищенні температури понад нормативні межі, а також у зниженні ефективності та збільшенні енергоспоживання. У часових рядах це виглядає як аномальні тренди й відхилення від базових патернів, що дає змогу оцінювати залишковий ресурс (RUL) і запобігати аварійним зупинкам. Використання сенсорів вібрації та температури є основою прогнозного обслуговування, оскільки своєчасне виявлення таких змін дозволяє уникнути каскадних відмов, економічних втрат і ризиків для безпеки.

1.3 Вібраційний та температурний моніторинг як джерело діагностичної інформації

Обидва підходи ґрунтуються на аналізі фізичних параметрів – механічних коливань і теплових змін – та формують основну систем прогнозного обслуговування.

Вібраційний моніторинг передбачає реєстрацію коливань за допомогою акселерометрів, віброметрів або лазерних датчиків. Аналізуються амплітуда, частота й фаза сигналу, що дозволяє виявляти дисбаланс, невирівняність, дефекти підшипників чи зубчастих передач. Дані формуються в вигляді часових рядів і досліджуються в часовій та частотній областях із застосуванням швидкого перетворення Фур'є (FFT). Діагностичним індикаторами виступають RMS, пік-фактор, куртозис і спектральні піки, пов'язані з характерними частотами вузлів. Перевагами є висока чутливість до механічних дефектів і можливість онлайн-інтеграції з системами SCADA, однак метод потребує калібрування та чутливий до зовнішніх збурень.

Температурний моніторинг базується на використані термопар, RTD, інфрачервоних камер або оптоволоконних сенсорів для контролю теплових профілів. Він дозволяє виявляти перегрів, спричинений тертям, перевантаженням чи електричними несправностями. Аналіз включає відстеження трендів, порогових перевищень і кореляцію з режимами роботи. Термографія забезпечує локалізацію гарячих точок із високою роздільною здатністю. Метод відзначається простотою впровадження та низькою вартістю, проте має меншу специфічність і залежить від умов довкілля.

Інтеграція обох підходів формує мультимодальну діагностичну систему, у якій кореляція вібраційних і температурних параметрів підвищує точність ідентифікації дефектів. Такі дані використовуються як вхід для моделей машинного навчання, що оцінюють залишковий ресурс обладнання. Практичне застосування підтверджене в проєктах зокрема NASA Turbofan та IMS Bearing, де комбінований моніторинг суттєво зменшує простої. Регламентування

здійснюється стандартами International Organization for Standardization ISO 10816 [5] для вібрацій і ISO 18434 [6] для термографії, що визначаються методики вимірювання та граничні рівні параметрів.

1.4 Методи раннього виявлення аномалій у часових рядах

Раннє виявлення аномалій у вібраційно-температурних часових рядах є ключовим етапом систем прогнозного обслуговування, оскільки дозволяє ідентифікувати відхилення від нормального режиму функціонування обладнання ще до настання відмови. Методи раннього виявлення аномалій у часових рядах доцільно класифікувати за підходом до аналізу даних, як показано на рис. 1.1 на три основні групи: статистичні, прогнозні та методи машинного навчання.

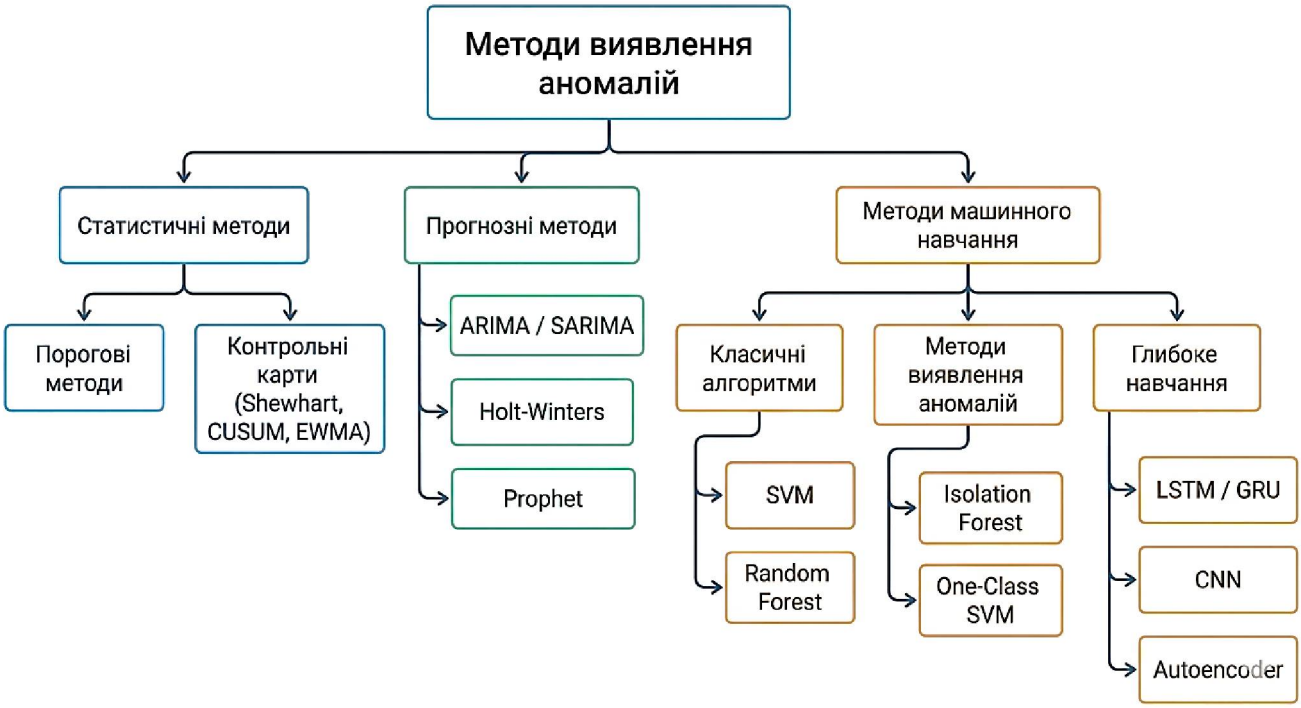


Рисунок 1.1 – Класифікація методів раннього виявлення аномалій у часових рядах.

До першої групи належать класичні статистичні методи, які базуються на аналізі розподілу даних і контролі відхилень від нормального стану.

Найпоширенішими є контрольні карти Шухарта, методи CUSUM та EWMA, а також порогові підходи, що використовують показники середньоквадратичного значення (RMS), стандартного відхилення або коефіцієнта ексцесу. Такі методи характеризуються простотою реалізації та високою швидкодією, однак мають обмежену ефективність у випадку складних нелінійних залежностей [1], [3].

Друга група включає методи прогнозування часових рядів, які передбачають побудову моделі нормальної поведінки сигналу з подальшим аналізом відхилення фактичних значень від прогнозованих. До них належать моделі ARIMA, SARIMA, Holt–Winters, а також сучасні підходи, зокрема Prophet. Аномалія визначається як значне перевищення залишкової помилки між прогнозом і фактичними даними. Ці методи ефективні для виявлення поступових змін стану системи та деградаційних процесів [11].

Третя група охоплює методи машинного навчання, які дозволяють автоматично виявляти складні закономірності у багатовимірних часових рядах. До них належать алгоритми класифікації (SVM, Random Forest), кластеризації та аномалійного аналізу (Isolation Forest, One-Class SVM), а також нейронні мережі. Особливу роль відіграють автоенкодери, що навчаються на нормальних даних і визначають аномалії за величиною помилки реконструкції [7], [14].

Як видно з рис. 1.1, найбільш перспективними для задач раннього виявлення відмов є методи машинного навчання, зокрема глибокі нейронні мережі, що здатні враховувати складні нелінійні залежності у даних.

На практиці найбільш ефективним є комбінований підхід, який поєднує статистичні методи для швидкого реагування та алгоритми машинного навчання для глибокого аналізу даних. Такий підхід забезпечує підвищення точності виявлення аномалій та зменшення кількості хибних спрацювань.

1.5 Порівняння моделей машинного та глибокого навчання для прогнозування відмов

У задачах прогнозного обслуговування ефективність системи значною мірою визначається вибором моделі машинного або глибокого навчання. Різні

підходи мають свої переваги та обмеження, що обумовлює необхідність їх порівняльного аналізу.

Класичні алгоритми машинного навчання, такі як метод опорних векторів (SVM), випадковий ліс (Random Forest) та градієнтний бустинг, демонструють високу ефективність при використанні інженерних ознак. Їх перевагами є відносно низькі обчислювальні витрати, інтерпретованість результатів та стабільність при роботі з невеликими наборами даних. Водночас вони потребують ретельної попередньої обробки та ручного формування ознак, що ускладнює їх застосування для складних часових рядів [3], [15].

Рекурентні нейронні мережі, зокрема LSTM та GRU, спеціально розроблені для роботи з послідовними даними та здатні враховувати довготривалі часові залежності. Вони забезпечують високу точність прогнозування деградаційних процесів і широко застосовуються для оцінки залишкового ресурсу обладнання. Проте такі моделі є обчислювально складними та потребують значних обсягів даних для навчання [12].

Згорткові нейронні мережі (1D-CNN) ефективно виявляють локальні патерни у сигналах, зокрема короткочасні аномальні коливання. Вони відзначаються високою швидкістю обробки та можливістю паралельних обчислень, однак менш ефективні для моделювання довготривалих залежностей.

Сучасні архітектури, такі як трансформери, базуються на механізмі уваги (attention) і дозволяють обробляти довгі послідовності даних із високою точністю. Вони перевершують традиційні рекурентні мережі у задачах аналізу складних часових рядів, проте потребують значних обчислювальних ресурсів [13].

Автоенкодери застосовуються переважно для виявлення аномалій у випадках обмеженої кількості розмічених даних. Вони дозволяють моделювати нормальний стан системи та виявляти відхилення на основі помилки реконструкції сигналу.

Для обґрунтування вибору моделі проведемо порівняльний аналіз основних підходів, результати якого наведено в табл. 1.1.

Таблиця 1.1.

Порівняння моделей машинного та глибокого навчання.

Модель	***Переваги***	***Недоліки***	***Застосування***
SVM	Висока точність на малих вибірках	Погано масштабується	Класифікація станів
Random Forest	Стійкість до шуму	Потребує інженерії ознак	Аналіз технічного стану
ARIMA	Простота, інтерпретованість	Не враховує нелінійність	Прогноз часових рядів
LSTM	Враховує часові залежності	Висока складність навчання	Прогноз деградації
CNN (1D)	Виявляє локальні патерни	Обмежена пам’ять	Аналіз сигналів
Autoencoder	Не потребує розмітки	Складність налаштування	Виявлення аномалій
Transformer	Робота з довгими рядами	Високі ресурси	Складні системи

Як видно з табл. 1.1, моделі глибокого навчання, зокрема LSTM та CNN, забезпечують найкращі результати при аналізі часових рядів, що обґрунтовує їх вибір для побудови системи раннього виявлення відмов.

Таким чином, результати порівняльного аналізу свідчать, що найбільш ефективними для задач раннього виявлення відмов є гібридні моделі, які поєднують переваги різних підходів, зокрема CNN–LSTM або автоенкодерів із рекурентними мережами. Такий підхід забезпечує баланс між точністю, швидкодією та здатністю працювати з багатовимірними часовими рядами.

1.6 Порівняльний аналіз існуючих рішень та систем прогнозного обслуговування

Сучасний ринок систем прогнозного обслуговування (PdM) є зрілим і охоплює як рішення великих промислових корпорацій, так і стартапи та open-source спільноту. Усі підходи базуються на однакових фізичних принципах

(вібрація, температура, електричні параметри), однак відрізняються глибиною аналітики, вартістю впровадження та рівнем адаптивності. До лідерів корпоративного сегмента належать платформи зокрема GE Predix (SmartSignal), Siemens MindSphere, EcoStruxure Machine Advisor, ABB Ability Genix, а також рішення від SKF, Uptake, C3.ai і SparkCognition. Вони пропонують повний цикл – від сертифікованих сенсорів до хмарної аналітики та інтеграції з ERP/MES. Ядро моделей зазвичай гібридне: класичне ML на інженерних ознаках + LSTM або автоенкодери.

Переваги – інтеграція, підтримка та відповідність стандартам безпеки; недоліки – висока вартість і обмежена прозорість алгоритмів. Сегмент ретрофіту представлений компаніями Augury, Samotics, Tractian, Ronds та іншими. Вони використовують бездротові сенсори й DL-моделі (LSTM-AE, 1D-CNN, трансформери), забезпечуючи точність 90–97% для підшипників і редукторів та скорочення простоїв на 25–45% за публічними кейсами 2023–2025 рр. Частина рішень реалізує edge-обчислення на платформах типу NVIDIA Jetson, що зменшує затримку та залежність від хмари. У науковому середовищі бенчмарками стали датасети NASA Turbofan Dataset, IMS Bearing Dataset, PRONOSTIA, XJTU-SY та PU, де трансформерні й гібридні моделі демонструють RMSE 12–15 для RUL і F1 > 0.98 для детекції аномалій. У низці досліджень 2024–2025 років відкриті реалізації перевищують типові комерційні платформи за точністю прогнозу на 20–40%, особливо для довгих мультимодальних рядів.

Порівняльний аналіз показує: великі корпоративні рішення виграють за інтеграцією та сервісом, але поступаються за гнучкістю й економічною ефективністю сучасним DL-підходам. Для більшості українських підприємств раціональним є гібридний сценарій – використання доступних сенсорів у поєднанні з адаптованими open-source моделями, розгорнутими локально або в приватній хмарі. Такий підхід забезпечує конкурентну точність прогнозування при суттєво нижчих витратах впровадження.

Висновок до розділу 1

У результаті проведеного дослідження здійснений порівняльний аналіз сучасних стратегій технічного обслуговування, методів вібраційно-температурного моніторингу, алгоритмів раннього виявлення аномалій та існуючих комерційних і наукових рішень у сфері прогнозного обслуговування. Проаналізовано еволюцію підходів від реактивного та планово-попереджувального обслуговування до Condition-Based Maintenance і Predictive Maintenance, визначено їх переваги та обмеження. Окремо розглянуто ефективність статистичних методів, класичного машинного навчання та глибоких нейромереж (LSTM, CNN, трансформери, автоенкодери) у задачах детекції аномалій і прогнозування залишкового ресурсу. Також проведено порівняння комерційних платформ і open-source рішень з точки зору точності, вартості впровадження, масштабованості та адаптивності до реальних виробничих умов.

Встановлено, що найбільшу точність і гнучкість у задачах аналізу вібраційно-температурних часових рядів забезпечують гібридні моделі глибокого навчання, особливо при роботі з мультимодальними даними та довгими послідовностями. Встановлено, що інтеграція вібраційного й температурного моніторингу суттєво підвищує достовірність діагностики та якість прогнозу RUL. Водночас визначено, що комерційні системи переважають за рівнем інтеграції та сервісної підтримки, але поступаються за економічною ефективністю та прозорістю алгоритмів сучасним відкритим реалізаціям. Обґрунтовано доцільність застосування гібридного підходу, який поєднує доступні сенсорні рішення з адаптованими моделями глибокого навчання, як технічно та економічно оптимального для впровадження в умовах промислових підприємств.

РОЗДІЛ 2
АНАЛІЗ ТА ПІДГОТОВКА ВХІДНИХ ДАНИХ

2.1 Джерела даних: промислові датчики вібрації та температури

Надійність та ефективність системи прогнозного обслуговування безпосередньо залежить від якості первинних вимірювань. Основними джерелами діагностичної інформації в даній роботі є промислові датчики вібрації та температури, встановлені на критичних вузлах обертового обладнання (електродвигуни, насоси, редуктори, компресори, турбіни). Саме ці параметри найраніше реагують на деградаційні процеси – знос підшипників, дисбаланс ротора, недостатнє змащення, перевантаження чи порушення теплового режиму.

Промислові датчики вібрації

У промислових системах моніторингу найчастіше використовуються п'єзоелектричні акселерометри. Їх робота базується на п'єзоелектричному ефекті: при механічній деформації кристала генерується електричний заряд, пропорційний прискоренню. Вихідний сигнал є аналоговим (мВ/g) або струмовим (4–20 мА) і подається на систему збору даних (DAQ, PLC, SCADA) [18].

– Акселерометри (прискорення, g або м/с^2) – одні з найпоширеніших, ефективні для виявлення високочастотних дефектів, таких як пошкодження підшипників або удари.

– Віброметри (швидкість, мм/с) – дозволяють оцінювати загальний стан роторних машин і відстежувати зміни в динаміці обертових елементів.

– Датчики переміщення (μm) – використовуються для вимірювання низькочастотних коливань великих турбін, генераторів або важкого обладнання.

– Триосьові сенсори – забезпечуються одночасну реєстрацію коливань у трьох координатах (X, Y, Z), що підвищує точність діагностики та дозволяє оцінювати комплексну динаміку обертових вузлів.

Технічні характеристики

Діапазон частот: від 1 Гц до 10 кГц, що дозволяє реєструвати як низькочастотні коливання великих роторів, так і високочастотні удари та дефекти підшипників. Чутливість 10-100 mB/g, що визначає величину вихідного сигналу при заданому прискоренні, робоча температура -40°C … +120°C, що забезпечує стабільну роботу в умовах промислових середовищ. Частота дискретизації системи збору даних: 5–50 кГц, необхідна для точного відображення швидких коливань і проведення аналізу у частотній області (FFT), включаючи виявлення гармонік, бокових смуг і ударних компонентів.

Висока частота дискретизації необхідна для подальшого аналізу сигналу у частотній області (FFT), виявлення гармонік, бокових смуг та ударних компонентів.

Діагностика стану обладнання на основі вібраційних сигналів передбачає формування інформативних ознак, зокрема середньоквадратичного значення (RMS), пікових та peak-to-peak параметрів, коефіцієнта форми (Crest Factor), ексцесу (Kurtosis), а також спектральних характеристик після перетворення Фур'є (FFT) і енергетичних показників у визначених частотних діапазонах. Зростання RMS або поява нових спектральних компонентів зазвичай є ранніми індикаторами зносу підшипників, розбалансування чи інших дефектів роторних вузлів.

Температурні датчики доповнюють вібраційний аналіз, відображаючи тепловий стан обладнання. Підвищення температури часто пов'язане з тертям, перевантаженням, електричними несправностями або погіршенням змащення. У промисловості застосовуються термопари (типи K, J, T), RTD-сенсори (Pt100, Pt1000), термістори NTC/PTC та безконтактні інфрачервоні датчики. Типовий діапазон вимірювання становить –50°C…+250°C із точністю ±0.1–1°C та інтервалом опитування 1–10 секунд. Хоча температурні дані мають нижчу частоту дискретизації, вони забезпечують цінну трендову інформацію щодо деградації.

У процесі аналізу враховують середні температурні значення, динаміку зростання (dT/dt), відхилення від номінального режиму та кореляцію з вібраційними показниками; підвищення температури на 10–20°C часто передує відмові підшипника або електродвигуна. Тепловий контроль регламентується стандартом ISO 18434 і рекомендаціями виробників обладнання. Інтеграція вібраційних і температурних сенсорів у PLC, SCADA, IoT-шлюзи та хмарні платформи реалізується в архітектурі edge + cloud, де первинна обробка виконується локально, а поглиблений аналіз – на сервері або в хмарі. Поєднання мультимодальних сигналів формує багатовимірні часові ряди, що підвищує точність раннього виявлення аномалій і прогнозування залишкового ресурсу (RUL).

2.2 Опис використаних наборів даних

У даній роботі для розробки, навчання та тестування системи використані синтетичні дані, що моделюють характеристики реальних промислових сигналів вібрації та температури. Синтетичний набір формується за допомогою сервісу DemoDataService і містить чотири канали вимірювань: vibration_x, vibration_y, vibration_z та temperature. Дані охоплюють три режими роботи обладнання — нормальний, деградаційний та критичний, — що дозволяє відтворити типові сценарії розвитку несправностей у промисловому середовищі.

У літературі з прогнозного обслуговування широко використовуються відкриті наукові бенчмарки. NASA CMAPSS містить багатовимірні часові ряди роботи турбореактивних двигунів із 21 сенсорним каналом і повною інформацією про момент відмови [4]. IMS Bearing Dataset (20 кГц, запис до повного руйнування) є еталонним для аналізу вібраційних сигналів деградації підшипників. Набір PRONOSTIA поєднує вібрацію та температуру за змінних режимів навантаження [3], а XJTU-SY Bearing Dataset містить експерименти до повної відмови підшипників при різних умовах експлуатації. Перелічені датасети слугують бенчмарками для оцінки якості подібних підходів у наукових публікаціях, однак у поточній реалізації системи не використовуються.

Архітектура системи розроблена з урахуванням можливості подальшої інтеграції з реальними датасетами через ендпоінти /api/v1/ingest/csv та /api/v1/ingest/json.

Синтетичні дані генеруються з частотою 1 вимірювання на секунду та містять реалістичні шуми, температурні дрейфи й вібраційні аномалії, що відповідають типовим несправностям підшипників і електродвигунів. Для завантаження зовнішніх даних система підтримує формати CSV та JSON з обов'язковими полями timestamp, vibration_x, vibration_y, vibration_z, temperature. Після завантаження дані проходять попередню обробку (інтерполяція пропусків, видалення викидів, масштабування) та нарізаються на ковзні вікна для подальшого вилучення ознак і навчання моделі.

2.3 Характеристики часових рядів вібрації та температури

Аналіз характеристик часових рядів є критично важливим етапом перед побудовою моделей машинного навчання. Вібраційні та температурні сигнали мають різну фізичну природу, частоту дискретизації, спектральні властивості та типові патерни деградації. Розуміння цих особливостей дозволяє правильно обрати методи попередньої обробки, інженерію ознак і архітектуру нейронної мережі.

Вібраційні сигнали в системах моніторингу зазвичай реєструються з частотою 10–50 кГц, що необхідно для виявлення ударних компонентів, дефектів підшипників, гармонік частоти обертання та бокових смуг модуляції. Така частота формує значні обсяги даних, тому застосовується віконування (sliding window) і моделі глибокого навчання, здатні ефективно працювати з довгими послідовностями. Вібрація є нестаціонарним і нелінійним процесом: із часом зростають RMS і дисперсія, з'являються нові спектральні компоненти, підвищується куртозис. Деградація проходить кілька фаз – від слабких імпульсів до різкого зростання амплітуди, що обґрунтовує застосування LSTM, TCN або Transformer замість класичних ARIMA. Після FFT виділяються фундаментальна частота (1×RPM), гармоніки, характерні частоти дефектів підшипників (BPFO, BPFI, BSF, FTF) та широкосмуговий шум на пізніх стадіях зносу.

Температурні ряди мають значно нижчу частоту дискретизації (0.1–1 Гц), оскільки теплові процеси є інерційними. Вони демонструють переважно трендову поведінку: поступове зростання при деградації, стабілізацію в номінальному режимі та різкий підйом перед відмовою. Температура менш насичена спектральними особливостями, але чутливіша до зовнішніх факторів – навантаження та умов середовища, що потребує нормалізації. На таблиці 2.1 зображені ключові характеристики.

Таблиця 2.1

Порівняльна таблиця ключових характеристик

Параметр	Вібрація	Температура
Частота дискретизації	10–50 кГц	0.1–1 Гц
Обсяг даних	Дуже великий	Помірний
Нестаціонарність	Висока	Помірна
Спектральна структура	Складна	Мінімальна
Інформативність для ранніх дефектів	Дуже висока	Середня
Чутливість до зовнішніх факторів	Середня	Висока

Комбінування вібрації та температури формує багатовимірний часовий ряд $Xt=[vibrationt,temperaturet]$, що підвищує точність прогнозування RUL і зменшує кількість хибних спрацювань. Зростання лише температури може свідчити про перевантаження, лише вібрації – про механічний дефект, тоді як одночасне зростання обох параметрів зазвичай сигналізує про критичну деградацію.

2.4 Попередня обробка даних (очищення, фільтрація шумів, нормалізація, масштабування)

Попередня обробка є критичним етапом підготовки вібраційно-температурних часових рядів до навчання моделей машинного та глибокого навчання. Сирі дані з промислових датчиків можуть містити шум, пропуски, аномальні викиди, різні масштаби вимірювання та часову несинхронність. Без коректної обробки навіть найсучасніша нейронна мережа демонструватиме нестабільну або некоректну роботу.

Пропуски в даних можуть виникати через втрату зв'язку з датчиком, помилки запису або перезавантаження PLC/SCADA. Для їх обробки застосовують лінійну інтерполяцію, forward/backward fill, ковзне середнє або видалення фрагментів у разі значних втрат. Викиди можуть бути спричинені апаратними збоями, електромагнітними перешкодами або аварійними режимами; їх виявляють за допомогою Z-score ($|z| > 3$), IQR, Hampel-фільтра або обмеження фізично допустимими межами.

Вібраційні сигнали зазвичай містять електричний шум і фонові коливання, тому застосовують фільтрацію: low-pass для усунення високочастотних перешкод, high-pass для видалення повільних трендів, band-pass для виділення потрібного діапазону та медіанний фільтр для імпульсних викидів. Через великий обсяг даних використовується віконування (256–4096 семплів із перекриттям 25–75%), що дозволяє сегментувати сигнал, сформувати батчі та зменшити обчислювальне навантаження.

Оскільки різні сенсори мають різні масштаби (вібрація – g або мм/с, температура – °C), застосовується нормалізація: Min–Max scaling для приведення до діапазону [0,1] або Standard Scaling для отримання нульового середнього та одиничної дисперсії; для LSTM і Transformer частіше використовується StandardScaler. Через різну частоту дискретизації мультимодальні дані синхронізуються шляхом downsampling вібрації, інтерполяції температури або агрегації показників у спільних часових інтервалах.

2.5 Інженерія ознак (Feature Engineering): статистичні, частотні (FFT), вейвлет-перетворення, ентропія тощо

Інженерія ознак є критичним етапом підготовки вібраційно-температурних даних до навчання моделей машинного та глибокого навчання. Навіть попри здатність LSTM, TCN і Transformer працювати із сирими сигналами, структуровані ознаки зменшують розмірність даних, пришвидшують навчання, підвищують інтерпретованість і стабільність моделей, особливо за обмежених вибірок.

У часовій області обчислюються статистичні характеристики: середнє значення, RMS як індикатор загального рівня вібрації, дисперсія та стандартне відхилення, пікові значення, Crest Factor і kurtosis. Зростання RMS, Crest Factor або kurtosis зазвичай сигналізує про розвиток ударних дефектів підшипників. Також використовується skewness для оцінки асиметрії розподілу.

У частотній області застосовується FFT для виділення фундаментальної частоти (1×RPM), гармонік і характерних частот дефектів підшипників (BPFO, BPFI, BSF, FTF), а також розраховуються спектральна ентропія, енергія в частотних смугах і spectral centroid. Частотні ознаки особливо ефективні для раннього виявлення локальних пошкоджень.

Оскільки вібраційні сигнали є нестаціонарними, використовується вейвлет-перетворення (CWT, DWT), яке забезпечує одночасний часово-частотний аналіз і кращу локалізацію короткочасних імпульсів; скалограми можуть подаватися до CNN як зображення. Додатково застосовуються ентропійні метрики (Shannon, spectral, approximate, sample entropy), що відображають зростання хаотичності сигналу під час деградації.

Для температурних рядів обчислюються середнє значення, тренд, похідна dT/dt, ковзне середнє, відхилення від номіналу та кореляція з вібрацією. У підсумку формується інтегрований вектор, що поєднує статистичні, спектральні, вейвлет- та ентропійні характеристики разом із температурними параметрами, забезпечуючи багатовимірний опис технічного стану обладнання. Таблиця 2.2 є

порівнянням основних типів інженерних ознак для того щоб обгрунтувати вибір найбільш інформативних ознак для подальшого аналізу та побудови моделей діагностики.

Таблиця 2.2

Таблиця порівняння основних типів інженерних ознак

Тип ознак	Переваги	Недоліки
Статистичні	Простота, швидкість	Менша чутливість до ранніх дефектів
FFT	Виявлення частот дефектів	Втрата часової локалізації
Вейвлет	Час + частота	Вища обчислювальна складність
Ентропія	Виявлення хаотичності	Складна інтерпретація

2.6 Обробка незбалансованих класів та визначення моменту відмови

Підготовка даних для навчання моделі є критично важливою через особливості промислових сигналів: різні одиниці вимірювання, масштаби значень, сильну диспропорцію між нормальними та аварійними станами, а також наявність шуму та аномалій. Без нормалізації та балансування класів модель може навчитися неправильно або зміститися в бік «здорового» стану обладнання. Для масштабування застосовують Min–Max Normalization, Standardization (Z-score) та Robust Scaling, причому для часових рядів важливо нормалізувати кожен канал окремо і використовувати параметри, обчислені лише на train-наборі. Сильний дисбаланс класів (95–99% нормальних станів) призводить до низької чутливості до відмов, тому застосовуються методи oversampling (Random, SMOTE), undersampling (Random, NearMiss, Tomek Links), ваги класів у функції втрат або Focal Loss. Для оцінки ефективності моделей незбалансованих даних доцільно використовувати метрики Precision, Recall, F1-score, ROC-AUC та PR-AUC, причому пріоритетом є високий Recall.

У разі обмежених даних відмов можливе використання аномальних підходів: Autoencoder, Isolation Forest, One-Class SVM, які навчаються лише на «нормі» і виявляють відхилення. Для уникнення витоку даних слід дотримуватися часової структури при розділенні наборів та нормалізації, щоб не використовувати майбутні значення, що забезпечує коректну оцінку якості моделі.

Висновки за розділом 2

У розділі проведено системний аналіз джерел даних і повного циклу їх підготовки для задачі прогнозного обслуговування. Показано, що інформативність і надійність моделей безпосередньо визначаються якістю первинних вимірювань і коректністю попередньої обробки. Встановлено, що вібраційні та температурні сенсори формують взаємодоповнюючі інформаційні потоки. Вібраційні сигнали (10–50 кГц) є високочутливими до ранніх дефектів (ударні події, дефекти підшипників, дисбаланс), тоді як температурні ряди (0.1–1 Гц) відображають інерційні теплові процеси та забезпечують надійні трендові індикатори деградації.

Їх інтеграція у вигляді багатовимірних часових рядів підвищує діагностичну роздільну здатність і зменшує кількість хибних спрацювань завдяки крос-перевірці фізичних механізмів відмов. У науковому середовищі для оцінки подібних підходів використовуються відкриті бенчмарки (CMAPSS, IMS, PRONOSTIA, XJTU-SY), що охоплюють сценарії повного життєвого циклу до відмови, реалістичні режими навантаження та мультимодальні вимірювання. У даній роботі застосовано синтетичні дані, що відтворюють ключові характеристики таких сигналів; архітектура системи розроблена з можливістю підключення реальних датасетів.

Уніфікація форматів (віконування, нормалізація, тензоризація) забезпечує відтворюваність експериментів і коректне навчання сучасних архітектур (LSTM, TCN, Transformer). Доведено, що попередня обробка є критичною умовою стабільності моделей. Очищення даних (інтерполяція пропусків, обробка

викидів), спектрально-орієнтована фільтрація (low/high/band-pass, медіанні фільтри) та синхронізація каналів (downsampling/інтерполяція) суттєво знижують шумову складову й артефакти вимірювань. Віконування з перекриттям оптимізує компроміс між часовою локалізацією та обчислювальними витратами.

РОЗДІЛ 3
РОЗРОБКА ТА ПОБУДОВА МОДЕЛІ ПРОГНОЗУВАННЯ

3.1 Вибір архітектури моделі прогнозування

Одним із ключових етапів розробки системи прогнозування є вибір архітектури моделі машинного навчання, яка здатна ефективно аналізувати часові ряди технічних параметрів обладнання та виявляти закономірності, що передують відмовам або аномальним станам. Від правильності вибору архітектури безпосередньо залежить точність прогнозування, стійкість моделі до шуму в даних та можливість її подальшого застосування у реальних виробничих умовах.

Дані, які використовуються в системах прогнозування технічного стану обладнання, як правило мають часову природу. Вони формуються у вигляді послідовностей вимірювань різних сенсорів: температури, вібрацій, тиску, навантаження, швидкості обертання тощо. Такі дані характеризуються високою розмірністю, наявністю шуму, пропусків значень та нелінійними залежностями між параметрами. Тому традиційні статистичні методи або прості алгоритми машинного навчання не завжди дозволяють досягти необхідного рівня точності прогнозування.

Першу групу складають класичні алгоритми машинного навчання, зокрема логістична регресія, метод опорних векторів, дерева рішень та ансамблеві методи, такі як Random Forest або Gradient Boosting. Перевагою цих методів є відносна простота реалізації, невеликі обчислювальні витрати та висока інтерпретованість результатів. Однак їх ефективність значною мірою залежить від якості попередньо сформованих ознак. Для роботи з часовими рядами необхідно виконувати складну процедуру інженерії ознак: обчислення статистичних характеристик, спектральних параметрів, автокореляційних функцій тощо. Крім того, такі алгоритми не здатні безпосередньо враховувати довготривалі часові залежності у даних.

Другою групою методів є нейронні мережі, спеціально розроблені для роботи з послідовними даними. До них належать рекурентні нейронні мережі та їх модифікації, зокрема Long Short-Term Memory (LSTM) та Gated Recurrent Unit (GRU). Ці архітектури дозволяють моделі зберігати інформацію про попередні стани системи та враховувати довготривалі залежності між значеннями часових рядів. Завдяки механізмам внутрішньої пам'яті такі мережі добре підходять для задач прогнозування технічного стану обладнання, оскільки відмова системи часто є результатом накопичення невеликих змін параметрів протягом тривалого часу.

Ще одним підходом до аналізу часових рядів є використання згорткових нейронних мереж. Хоча первинно такі мережі розроблялись для обробки зображень, їх одновимірні варіанти (1D-CNN) ефективно застосовуються для аналізу сигналів сенсорів. Згорткові шари дозволяють автоматично виділяти локальні патерни у часових рядах, наприклад короткочасні піки вібрації або аномальні коливання параметрів. Перевагою такого підходу є висока швидкість обчислень та можливість ефективної паралельної обробки даних.

З метою поєднання переваг обох підходів у сучасних системах прогнозування часто застосовуються гібридні архітектури. Однією з найпоширеніших є модель CNN–LSTM. У такій архітектурі згорткові шари використовуються для автоматичного виділення інформативних ознак із сирих сенсорних сигналів, після чого рекурентні шари аналізують часову структуру цих ознак та формують прогноз майбутнього стану системи. Такий підхід дозволяє одночасно враховувати як локальні особливості сигналів, так і довготривалі часові залежності.

Крім того, для задач виявлення аномалій можуть використовуватись автоенкодери – спеціальний клас нейронних мереж, призначених для відновлення вхідних даних. Під час навчання така модель вивчає нормальну поведінку системи, після чого значне зростання похибки реконструкції може

інтерпретуватися як наявність аномалії. Цей підхід є особливо корисним у випадках, коли кількість прикладів реальних відмов обладнання є обмеженою.

Дані, наведені в табл. 3.1, свідчать про багатовимірний та нестаціонарний характер вхідних сигналів. Наявність шумів, коливань амплітуди та різної частоти змін параметрів ускладнює їх аналіз традиційними статистичними методами, що обґрунтовує доцільність застосування методів глибокого навчання.

Таблиця 3.1

Порівняння можливих архітектур моделей прогнозування

Архітектура	Переваги	Недоліки
Класичні алгоритми ML	Простота реалізації, невеликі обчислювальні витрати	Потребують складної інженерії ознак
LSTM / GRU	Можливість врахування довготривалих залежностей	Високі обчислювальні витрати
1D-CNN	Ефективне виділення локальних патернів у сигналах	Менш ефективні для довгих часових залежностей
CNN–LSTM	Поєднання локального аналізу сигналів і часових залежностей	Складніша архітектура моделі
Autoencoder	Можливість виявлення аномалій без повної розмітки даних	Складність інтерпретації результатів

3.2 Проєктування структури нейронної мережі

Після вибору архітектури моделі наступним етапом є проєктування структури нейронної мережі та визначення способу обробки вхідних даних. У межах даної роботи використовується гібридна архітектура CNN–LSTM, яка поєднує можливості згорткових та рекурентних нейронних мереж для аналізу багатовимірних часових рядів.

Вхідні дані представлені у вигляді часових послідовностей сенсорних параметрів обладнання. Для підготовки даних використовується метод ковзного

вікна (sliding window), який дозволяє розбити часові ряди на фрагменти однакової довжини. Кожне вікно містить послідовність значень параметрів системи за певний проміжок часу та використовується як окремий приклад під час навчання моделі.

Структура нейронної мережі складається з трьох основних блоків: згорткового блоку, рекурентного блоку та блоку прогнозування. Згорткові шари виконують автоматичне виділення інформативних ознак із часових сигналів, що дозволяє виявляти локальні закономірності та короткочасні аномальні коливання параметрів. Для підвищення стабільності навчання застосовується нормалізація пакетів даних (Batch Normalization), а для запобігання перенавчанню використовується метод Dropout.

Після обробки згортковими шарами сформовані ознаки передаються до рекурентного блоку, що складається з LSTM-шарів. Ці шари аналізують часові залежності між параметрами системи та дозволяють моделі враховувати динаміку зміни технічного стану обладнання.

Останнім етапом є повно-зв'язаний шар, який формує остаточний прогноз. Тип вихідного шару залежить від задачі: для класифікації використовується сигмоїдна функція активації, а для задач регресії – лінійна функція.

Структура запропонованої гібридної моделі CNN-LSTM, що поєднує можливості згорткових та рекурентних нейронних мереж для аналізу часових рядів, наведена на рис. 3.1.

Як показано на рис. 3.1, вхідні дані спочатку обробляються згортковими шарами для виділення локальних ознак сигналу, після чого отримані представлення передаються до шару LSTM, який враховує довготривалі часові залежності. Така послідовність обробки дозволяє підвищити точність виявлення аномальних станів обладнання.

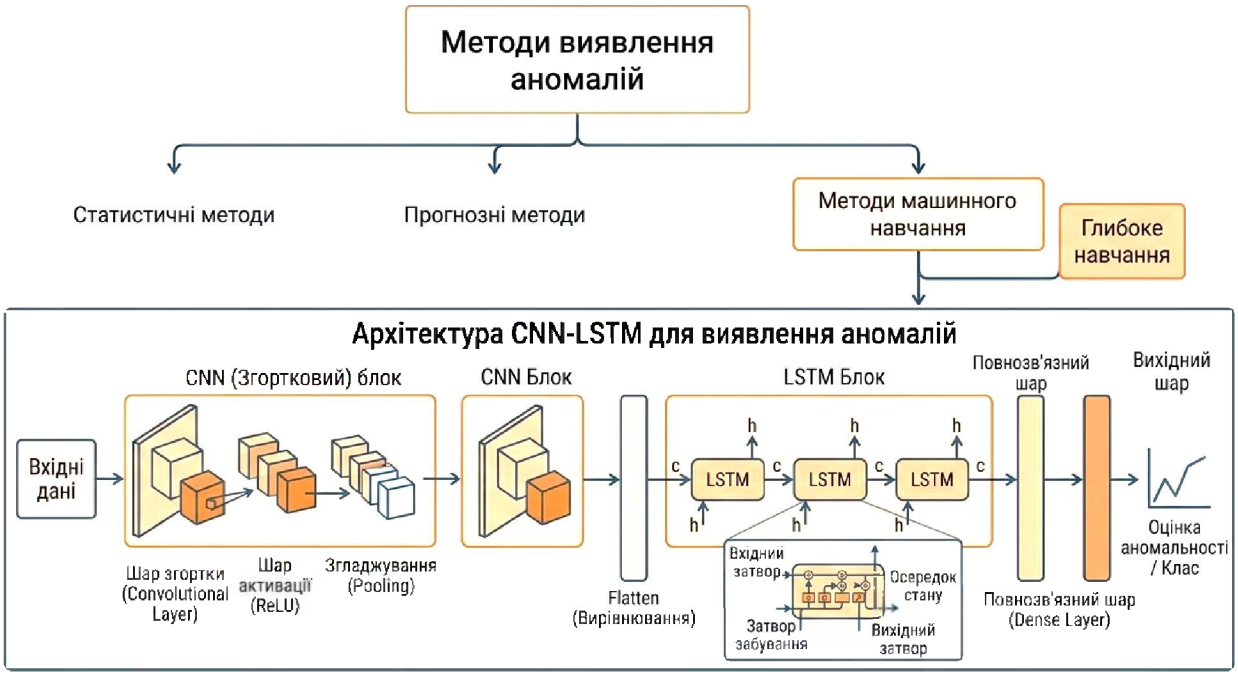


Рисунок 3.1 – Архітектура гібридної моделі CNN-LSTM для аналізу часових рядів.

Таблиця 3.2.

Основні шари нейронної мережі CNN-LSTM

Шар	***Тип шару***	***Призначення***
Input Layer	Вхідний шар	Прийом багатовимірного часового ряду
Conv1D	Згортковий шар	Виділення локальних патернів сигналу
Batch Normalization	Нормалізація	Стабілізація процесу навчання
ReLU	Функція активації	Моделювання нелінійних залежностей
Dropout	Регуляризація	Запобігання перенавчанню
LSTM	Рекурентний шар	Аналіз часових залежностей
Dense	Повно-зв'язаний шар	Формування прогнозу
Output	Вихідний шар	Результат роботи моделі

Як видно з табл. 3.2, сформований набір ознак охоплює статистичні, спектральні та часові характеристики сигналів. Такий комплексний опис дозволяє моделі враховувати як миттєві зміни параметрів, так і довготривалі деградаційні тенденції в роботі обладнання.

Параметри архітектури моделі CNN-LSTM наведено в табл. 3.3.

Таблиця 3.3

Основні гіперпараметри моделі

Параметр	Значення
Розмір вхідного вікна	32 вимірювання
Кількість Conv1D фільтрів	32 – 64
Розмір ядра згортки	3 – 5
Кількість LSTM нейронів	64
Dropout	0.25
Оптимізатор	Adam
Функція втрат	Binary Crossentropy / MSE

Параметри, наведені в табл. 3.3, відображають поєднання згорткових і рекурентних шарів, що дозволяє моделі одночасно виділяти локальні патерни сигналу та враховувати довготривалі часові залежності, характерні для процесів деградації обладнання.

3.3 Навчання та валідація моделі

Після розробки архітектури нейронної мережі проводиться її навчання та перевірка ефективності на незалежних даних. Перед початком навчання виконується попередня обробка даних: сенсорні показники нормалізуються, а часові ряди розбиваються на фрагменти однакової довжини за допомогою методу ковзного вікна. Це дозволяє сформувати навчальні приклади для моделі. Для оцінювання якості прогнозування дані поділяються на навчальну, валідаційну та тестову вибірки.

Навчальна вибірка використовується для навчання моделі, валідаційна – для контролю процесу навчання та підбору параметрів, а тестова – для остаточної перевірки ефективності моделі.

Результати навчання та оцінювання моделі наведено в табл. 3.4.

Таблиця 3.4

Розподіл даних для навчальної моделі

Тип вибірки	Частка даних	Призначення
Навчальна (Train)	80 %	Навчання параметрів моделі
Валідаційна (Validation)	20 %	Не використовується в поточній реалізації
Тестова (Test)	–	Оцінка точності моделі

Таблиця 3.4 відображає розподіл даних між навчальною та валідаційною вибірками у пропорції 80% до 20%, що відповідає реалізації у коді через функцію train_test_split. Такий розподіл забезпечує достатній обсяг навчальних даних для збіжності моделі та незалежну оцінку якості на валідаційній вибірці.

Таблиця 3.5

Основні метрики оцінювання моделей

Метрика	Опис	Призначення
Accuracy	Частка правильних прогнозів	Загальна точність класифікації
Precision	Точність позитивних прогнозів	Оцінка виявлення аномалій
Recall	Повнота позитивних прогнозів	Виявлення всіх аномальних станів
F1-score	Середнє значення Precision і Recall	Комплексна оцінка моделі
MSE	Середньоквадратична похибка	Оцінка точності регресійних прогнозів

Навчання нейронної мережі виконується за допомогою алгоритму зворотного поширення помилки з використанням оптимізатора Adam. Для

запобігання перенавчанню застосовується метод ранньої зупинки (Early Stopping), який зупиняє навчання у випадку відсутності покращення результатів на валідаційній вибірці (табл. 3.5).

Висновок за розділом 3

У третьому розділі було розглянуто процес розробки моделі прогнозування відмов промислового обладнання на основі аналізу вібраційно-температурних часових рядів. Проведено обґрунтований вибір архітектури моделі з урахуванням специфіки промислових сигналів, їхньої нестаціонарності, великої довжини послідовностей та наявності довгострокових часових залежностей.

Встановлено, що для обробки таких даних доцільно застосовувати сучасні архітектури глибокого навчання, здатні ефективно моделювати часову структуру сигналів та виявляти приховані деградаційні патерни. У межах розділу було спроєктовано структуру нейронної мережі, визначено основні компоненти моделі, включаючи вхідний шар, послідовність рекурентних або згорткових блоків для обробки часових залежностей, а також повно-зв'язні шари для формування фінального прогнозу.

РОЗДІЛ 4
ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНІ РЕЗУЛЬТАТИ

4.1 Архітектура програмного веб-застосунку

Для практичної реалізації запропонованої моделі раннього виявлення відмов було розроблено веб-застосунок, який об’єднує функціональні компоненти для обробки даних, взаємодії з моделлю глибокого навчання та відображення результатів аналізу. Програмне рішення побудоване за клієнт-серверною архітектурою.

У його структурі передбачено окремі модулі, що відповідають за завантаження та попередню обробку вхідних даних, виконання обчислень із використанням моделі глибокого навчання, а також формування та візуалізацію отриманих результатів. Взаємодія з користувачем реалізується через веб-інтерфейс.

Загальну структуру програмного рішення наведено на рис. 4.1.

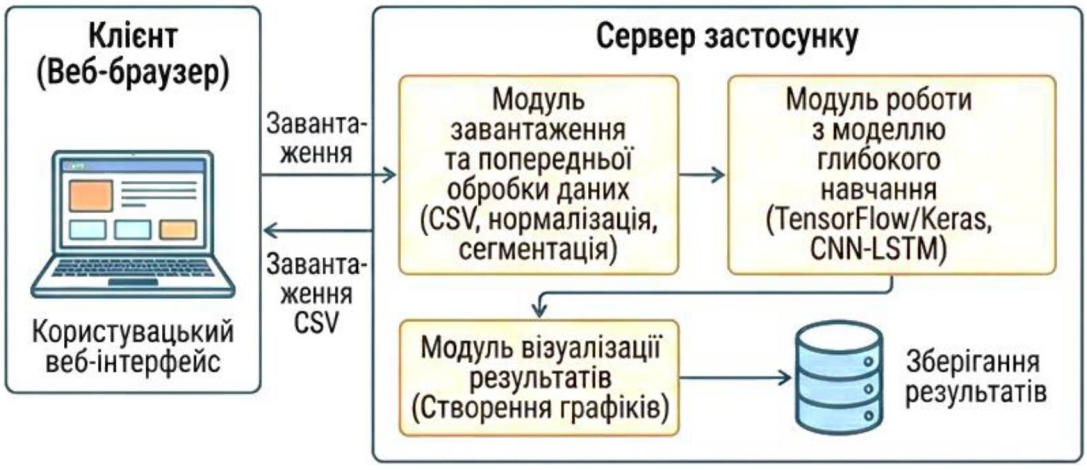


Рисунок 4.1 – Архітектура програмного веб-застосунку.

Як показано на рис. 4.1, програмна реалізація забезпечує послідовну обробку даних від етапу завантаження сигналів до отримання результатів прогнозування технічного стану обладнання.

4.2 Опис програмної реалізації

Програмну частину системи реалізовано мовою Python із використанням сучасного стеку бібліотек для обробки даних, машинного навчання та побудови веб-сервісів. Для розробки та навчання нейронної мережі використано TensorFlow/Keras, що забезпечує гнучкість у побудові архітектури CNN-LSTM та ефективну оптимізацію процесу навчання.

Серверну частину реалізовано на базі FastAPI, який підтримує асинхронну обробку запитів, автоматичну генерацію документації API (OpenAPI/Swagger) та високу продуктивність при роботі з потоками даних. Архітектура програмного модуля побудована за модульним принципом і включає кілька основних компонентів: модуль завантаження даних, модуль попередньої обробки, модуль навчання моделі, модуль прогнозування та веб-інтерфейс взаємодії з користувачем.

На етапі завантаження система приймає часові ряди у форматі CSV. Дані проходять валідацію (перевірка структури, відсутності пропущених значень або їх обробка) та перетворюються у внутрішній формат. Попередня обробка включає нормалізацію (наприклад, MinMaxScaler або стандартизацію), а також формування ковзних вікон (sliding window), що дозволяє перетворити часовий ряд у вибірку для навчання нейронної мережі. Модель CNN-LSTM реалізує комбінований підхід: згорткові шари (CNN) використовуються для виділення локальних патернів у часових вікнах, тоді як рекурентні шари (LSTM) моделюють довгострокові залежності у даних. Навчання моделі виконується з фіксованою кількістю епох (80), що було обрано емпірично як компроміс між якістю та часом навчання.

Параметр batch_size визначається динамічно залежно від розміру датасету, що дозволяє оптимізувати використання пам’яті та стабільність градієнтів під час навчання. CNN-LSTM використовується переважно для офлайн-аналізу, навчання та побудови узагальненої моделі нормальної поведінки системи. Для задач онлайн-виявлення аномалій у реальному часі застосовується алгоритм

IsolationForest, який має меншу обчислювальну складність і не потребує тривалого перенавчання. Це дозволяє швидко обробляти нові вхідні дані та оперативно визначати відхилення від нормального стану. Взаємодія з користувачем реалізована через REST API, що надає можливість завантаження даних, ініціалізації процесу аналізу та отримання результатів. FastAPI забезпечує асинхронну обробку запитів, що особливо важливо при роботі з великими обсягами часових рядів.

Результати обробки представлені у вигляді графіків (наприклад, із використанням Chart.js), де відображається часовий ряд разом із позначеними аномаліями. Користувацький інтерфейс орієнтований на мінімізацію складності взаємодії: користувачу достатньо завантажити файл із даними, після чого система автоматично виконує всі етапи — від попередньої обробки до побудови прогнозу та візуалізації результатів. Такий підхід дозволяє використовувати систему без глибоких знань у галузі машинного навчання. Запропонована програмна реалізація забезпечує чітке розділення між офлайн-моделлю (CNN-LSTM), що відповідає за навчання та аналіз складних залежностей, і онлайн-алгоритмом (IsolationForest), що виконує швидке виявлення аномалій. Це підвищує масштабованість, продуктивність та придатність системи до використання в умовах реальної експлуатації.

4.3 Організація експерименту

Для оцінювання ефективності розробленої моделі було використано набір часових рядів, що поєднують вібраційні та температурні параметри та відтворюють умови функціонування промислового обладнання. Вихідні дані розподілено на навчальну та тестову вибірки у пропорції 80% до 20%. У процесі навчання моделі застосовано функцію втрат Binary Crossentropy та оптимізатор Adam.

Навчання проводилося протягом 80 епох із раннім зупиненням (EarlyStopping, patience=8) та динамічним розміром батчу від 8 до 32 залежно від обсягу навчальної вибірки. Відповідний процес подано на рисунку 4.2. Для

контролю якості навчання проаналізовано зміну значень функції втрат і точності класифікації впродовж навчання, що дозволило оцінити збіжність моделі та стабільність її роботи.

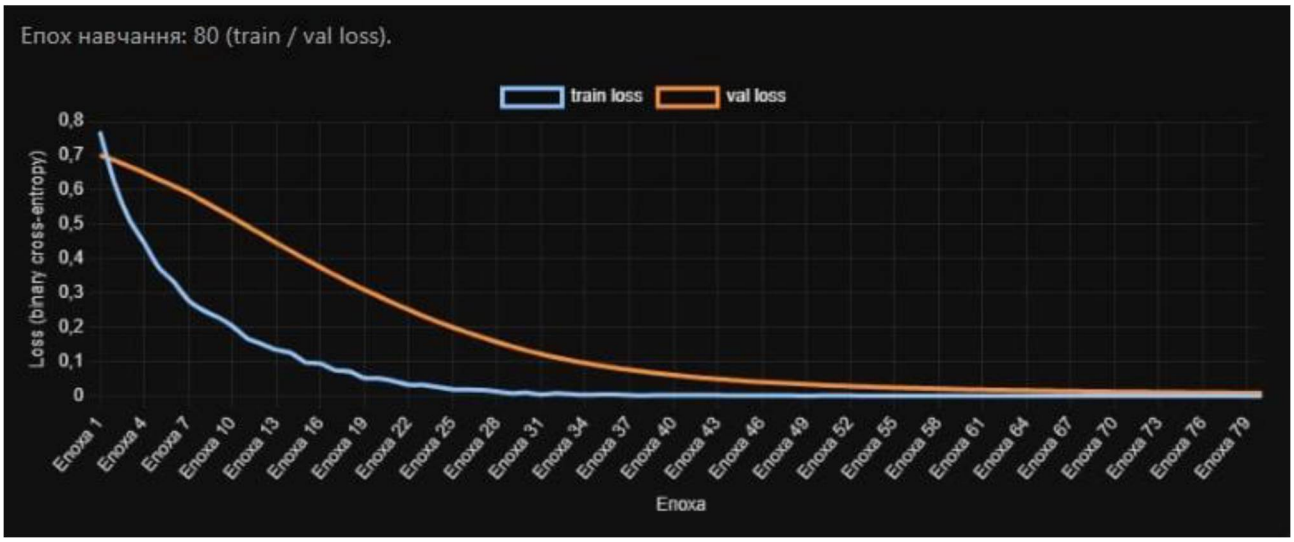


Рисунок 4.2 – Графік відповідності результатів функції втрат та точності класифікації впродовж навчання моделі.

Як видно з рис. 4.2, спостерігається стабільне зменшення значення функції втрат як на навчальній, так і на валідаційній вибірках, що свідчить про відсутність перенавчання моделі.

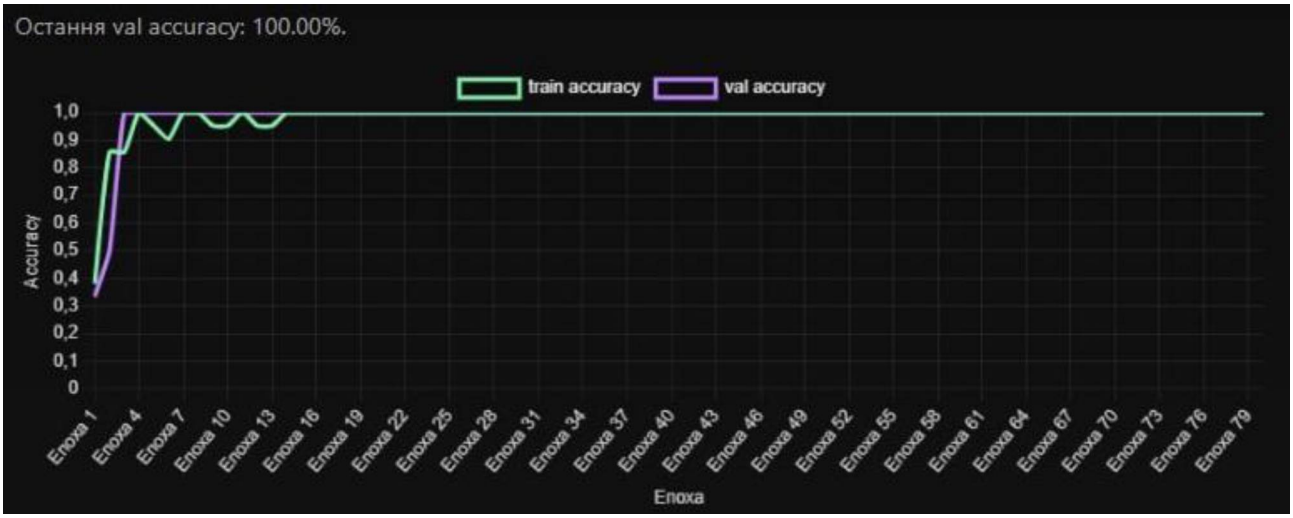


Рисунок 4.3 – Зміна точності класифікації (accuracy) під час навчання моделі CNN-LSTM.

У процесі навчання спостерігається поступове зменшення значення функції втрат як на навчальній, так і на валідаційній вибірках, що свідчить про коректне налаштування моделі та відсутність перенавчання.

На рис. 4.3 відображено динаміку точності класифікації (accuracy) на навчальній та валідаційній вибірках упродовж навчання. Стабільне зростання обох кривих та їх збіжність свідчать про відсутність перенавчання моделі.

Таблиця 4.1

API інтерфейси системи

Метод	Endpoint	Опис
GET	/health	Перевірка стану сервісу
POST	/api/v1/ingest/csv	Завантаження сенсорних даних у форматі CSV
POST	/api/v1/ingest/json	Передача сенсорних даних у форматі JSON
POST	/api/v1/dataset/build	Побудова набору ознак для навчання
POST	/api/v1/train/classifier	Тренування моделі машинного навчання
POST	/api/v1/evaluate	Оцінка точності моделі
POST	/api/v1/predict	Прогнозування технічного стану обладнання

У таблиці 4.1 наведено перелік HTTP-ендпоінтів системи HTTP запитів до нашої моделі.

4.4 Функціональне тестування програмної системи

Для перевірки працездатності розробленої системи проведено функціональне тестування відповідно до «Програми і методики випробувань» (Додаток В). Перевірка охоплювала основні модулі: завантаження та попередню обробку даних, роботу моделі CNN-LSTM і відображення результатів у веб-інтерфейсі.

На початковому етапі оцінювалась коректність зчитування «сирих» даних із CSV-файлів та їх подальша обробка. Система виконує перевірку структури даних, виявлення пропущених значень, нормалізацію та формування часових вікон для подачі в модель.

Результати тестування відображаються у веб-інтерфейсі у вигляді перших рядків набору даних і зведеної інформації про їхню структуру та наявність пропусків.

Приклад виведення результатів наведено на рис. 4.4.

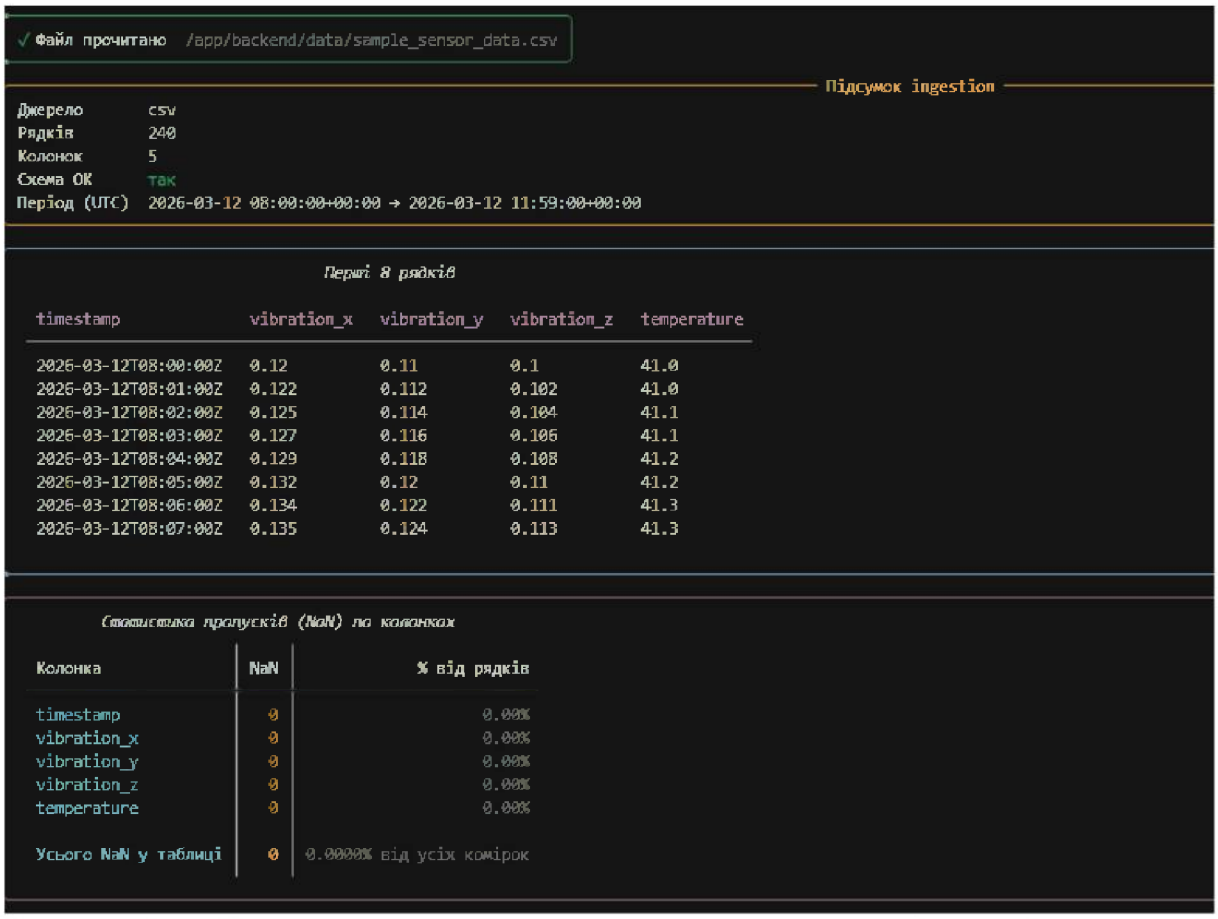


Рисунок 4.4 – Результат завантаження та первинної перевірки вхідних даних датчиків.

Другий етап тестування спрямований на перевірку роботи модуля глибинного навчання CNN-LSTM для аналізу часових рядів. Після підготовки даних система виконує подачу сформованих часових вікон на вхід моделі. У

процесі роботи відображаються графіки зміни функції втрат (loss) та метрики точності (accuracy), що дозволяє оцінити якість навчання моделі.

Після завершення обчислень система формує графік порівняння прогнозованих значень та фактичних показників технічного стану обладнання, що підтверджує адекватність моделі для задачі раннього виявлення аномалій.

Приклад графіку навчання наведено на рис. 4.5.

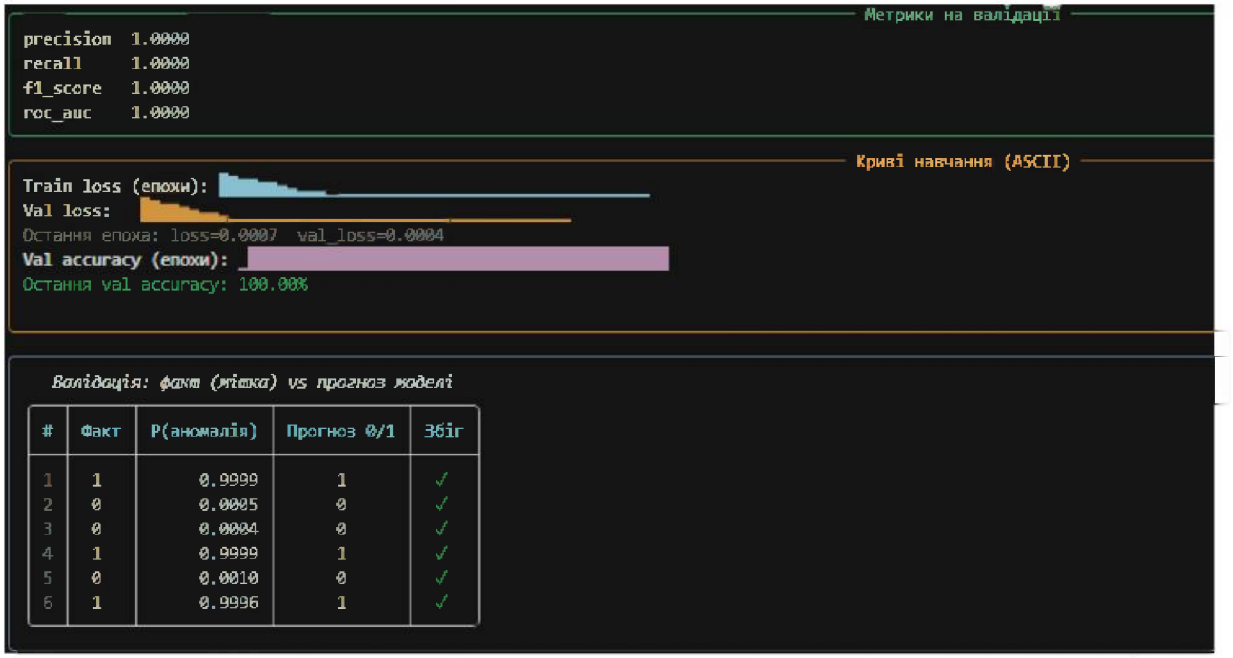


Рисунок 4.5 – Графіки процесу навчання та перевірки моделі CNN-LSTM

Третій етап тестування передбачає перевірку здатності системи виявляти аномальні відхилення у роботі обладнання на основі результатів прогнозування. Після аналізу нових даних система автоматично визначає моменти часу, в яких спостерігаються відхилення від нормального режиму роботи, та формує відповідні повідомлення для користувача.

У веб-інтерфейсі відображається графік часових рядів з позначеними ділянками аномалій, а також текстове повідомлення про потенційну несправність обладнання. Приклад таких повідомлень наведено на рис. 4.6.

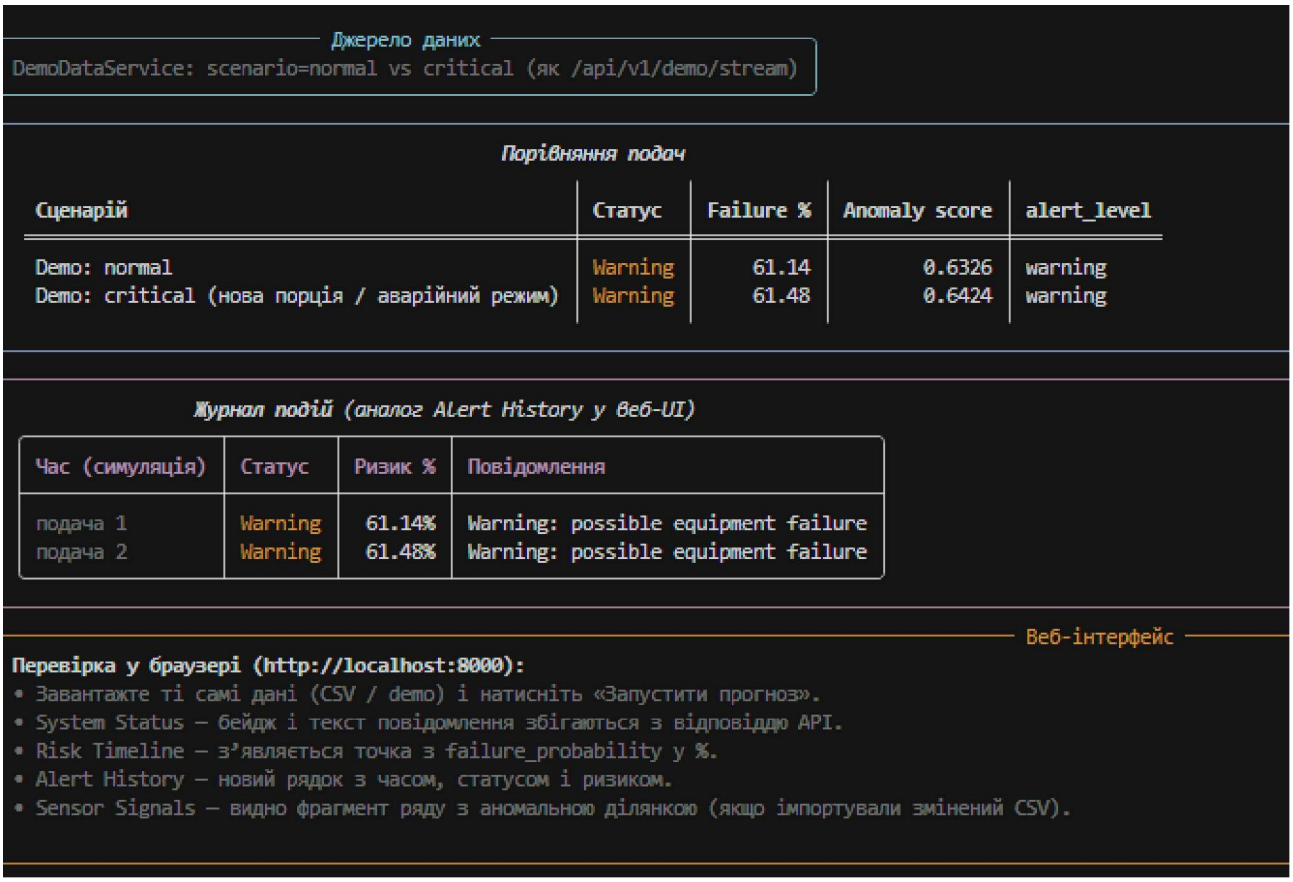


Рисунок 4.6 – Візуалізація виявлених аномалій у роботі обладнання.

Під час тестування перевірено здатність системи коректно обробляти вхідні дані з пропусками та шумами, виконувати аналіз часових вікон, формувати прогноз технічного стану обладнання та візуалізувати результати аналізу для користувача. Результати функціональних тестів узгоджуються з отриманими експериментальними метриками точності та підтверджують працездатність розробленої системи у реальних сценаріях використання.

4.5 Оцінювання якості моделі

Для оцінювання ефективності моделі використано стандартні метрики класифікації: Accuracy, Precision, Recall та F1-score. Отримані результати подано в таблиці 4.2.

Таблиця 4.2.

Результати оцінювання моделі CNN-LSTM

Метрика	***Значення***
Accuracy	**1.00**
Precision	**1.00**
Recall	**1.00**
F1-score	**1.00**
ROC-AUC	1.00

Результати, подані в табл. 4.2, отримані на синтетичних даних із демо-мітками (50% нормальних / 50% аномальних вікон) за допомогою скрипту qa_terminal_test_2_cnn_lstm.py. Значення метрик дорівнюють 1.00, що є очікуваним результатом для ідеально збалансованих синтетичних даних із чітко вираженими класами. Архітектура CNN-LSTM демонструє здатність розрізняти класи; реальні значення метрик залежатимуть від характеристик виробничих даних.

Висновки за розділом 4

У розділі наведено опис програмної реалізації системи раннього виявлення відмов та результати експериментального дослідження. Працездатність системи перевірено шляхом функціонального тестування згідно з «Програмою і методикою випробувань» (Додаток В). У межах тестування оцінено коректність завантаження та обробки вхідних даних, роботу моделі CNN-LSTM, а також формування повідомлень про аномальні стани обладнання.

Розроблений веб-застосунок забезпечує практичне впровадження моделі, а отримані експериментальні результати підтверджують доцільність використання гібридної архітектури CNN-LSTM для аналізу вібраційно-температурних часових рядів.

ВИСНОВКИ

У кваліфікаційній роботі розв'язано актуальну задачу підвищення надійності функціонування промислового обладнання шляхом раннього виявлення відмов на основі інтелектуального аналізу вібраційно-температурних часових рядів із використанням методів глибокого навчання та подальшої програмної реалізації отриманих результатів.

У процесі виконання роботи отримано такі основні результати:

1. Проведено аналіз сучасних підходів до забезпечення надійності промислового обладнання та методів раннього виявлення відмов, що дозволило обґрунтувати доцільність використання моделей глибокого навчання для аналізу даних технічного стану.

2. Досліджено особливості вібраційно-температурних часових рядів, зокрема їх нестаціонарність, шумовий характер та наявність деградаційних процесів, що ускладнюють застосування традиційних методів аналізу.

3. Виконано підготовку та попередню обробку даних: очищення, нормалізацію, синхронізацію та формування навчальних і тестових вибірок, придатних для подальшого використання у моделі.

4. Сформовано інформативний простір ознак на основі статистичних, спектральних і часово-частотних характеристик сигналів, що підвищило якість представлення даних для навчання моделі.

5. Обґрунтовано вибір та побудовано гібридну модель глибокого навчання CNN-LSTM, здатну одночасно виділяти локальні патерни сигналу та враховувати довготривалі часові залежності.

6. Проведено навчання та експериментальне оцінювання моделі з використанням метрик Accuracy, Precision, Recall та F1-score, що підтвердило її високу ефективність для виявлення аномальних станів обладнання.

7. Реалізовано програмний веб-застосунок, який забезпечує практичне використання моделі для аналізу даних і підтримки прийняття рішень інженерно-технічним персоналом.

Отримані результати підтверджують досягнення поставленої мети роботи та доцільність застосування глибокого навчання для задач прогнозного обслуговування промислового обладнання.

Працездатність програмної реалізації підтверджено результатами функціонального тестування відповідно до розробленої програми і методики випробувань

Практична цінність роботи полягає у можливості використання розробленого підходу в системах моніторингу технічного стану обладнання, інтеграції у промислові інформаційні середовища та застосуванні в освітньому процесі за спеціальністю 123 «Комп'ютерна інженерія».

Апробацію результатів дослідження здійснено шляхом участі у науково-практичній конференції, де було представлено доповідь за тематикою роботи, що підтверджує наукову значущість і практичну цінність отриманих результатів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Jardine A. K. S., Lin D., Banjevic D. Maintenance, replacement, and reliability: theory and applications. Boca Raton: CRC Press, 2006. 303 p.
2. Mobley R. K. An introduction to predictive maintenance. 2nd ed. Burlington: Elsevier, 2002. 432 p.
3. Lei Y., Li N., Guo L., Li N., Yan T., Lin J. Machinery health prognostics: A systematic review // Mechanical Systems and Signal Processing. 2018. Vol. 104. P. 799–834.
4. Saxena A., Goebel K. Turbofan Engine Degradation Simulation Data Set. NASA Ames Prognostics Data Repository, 2008. URL: https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository / (дата звернення: 25.03.2026).
5. ISO 10816-1:2016. Mechanical vibration – Evaluation of machine vibration by measurements on non-rotating parts. Geneva: International Organization for Standardization, 2016.
6. ISO 18434-1:2008. Condition monitoring and diagnostics of machines – Thermography. Geneva: International Organization for Standardization, 2008.
7. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge: MIT Press, 2016. 775 p.
8. Chollet F. Deep Learning with Python. New York: Manning Publications, 2017. 384 p.
9. Brownlee J. Deep Learning for Time Series Forecasting. Machine Learning Mastery, 2018. 308 p.
10. Deloitte Insights. Predictive maintenance and the smart factory. 2022. URL: https://www2.deloitte.com (дата звернення: 25.03.2026).
11. Hyndman R. J., Athanasopoulos G. Forecasting: principles and practice. 3rd ed. Melbourne: OTexts, 2021. URL: https://otexts.com/fpp3/ (дата звернення: 25.03.2026).

12. Hochreiter S., Schmidhuber J. Long short-term memory // Neural Computation. 1997. Vol. 9. No. 8. P. 1735–1780.
13. Vaswani A. et al. Attention is all you need // Advances in Neural Information Processing Systems. 2017. Vol. 30.
14. Zhao R., Yan R., Chen Z., Mao K., Wang P., Gao R. X. Deep learning and its applications to machine health monitoring // Mechanical Systems and Signal Processing. 2019. Vol. 115. P. 213–237.
15. Zhang W., Yang D., Wang H. Data-driven methods for predictive maintenance of industrial equipment: A survey // IEEE Systems Journal. 2019. Vol. 13. No. 3. P. 2213–2227.
16. Плескач В. Л., Затонацька Т. Г. Інформаційні системи і технології на підприємствах. Київ: Знання, 2018. 718 с.
17. Кузнєцов М. П., Козлов В. І. Основи технічної діагностики та неруйнівного контролю. Харків: ХНУРЕ, 2019. 312 с.
18. ISO 13374-1:2003. Condition monitoring and diagnostics of machines – Data processing, communication and presentation. Part 1: General guidelines. Geneva: International Organization for Standardization, 2003.
19. IEC 61499-1:2012. Function blocks – Part 1: Architecture. Geneva: International Electrotechnical Commission, 2012.

ДОДАТКИ

Додаток А

Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Навчально-науковий інститут комп'ютерних наук та штучного інтелекту
Кафедра комп'ютерних систем та робототехніки

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних
систем та робототехніки
Максим ХРУСЛОВ
_____
(підпис)
«13» жовтня 2025року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувача першого (бакалаврського) рівня вищої освіти
(першого (бакалаврського) / другого (магістерського))

САВОЧКИ ОЛЕКСАНДРА ІГОРОВИЧА
(прізвище, ім'я, по батькові студента)

Спеціальність: 123 «Комп'ютерна інженерія»
Освітня програма «Комп'ютерна інженерія»
1. Тема роботи **СИСТЕМА РАННЬОГО ВИЯВЛЕННЯ НЕСПРАВНОСТЕЙ У ВИРОБНИЧИХ ПРОЦЕСАХ НА ОСНОВІ ШТУЧНОГО ІНТЕЛЕКТУ**
керівник роботи **Мороз Ольга Юріївна, PhD з комп.наук, доцент зво кафедри,**
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від ***№ 4101-5/1245 від 28.05.2026 року.***
2. Строк подання студентом роботи ***30 травня 2026 року***
3. Вихідні дані до роботи:
   1) технічні характеристики та режими роботи типового обертового обладнання;
   2) методи інженерії ознак для аналізу сигналів: швидке перетворення Фур'є (FFT), вейвлет-перетворення, статистичні характеристики сигналів;
   3) вимоги до архітектури веб-застосунку для відображення стану обладнання та результатів прогнозування;
   4) персональний комп'ютер для розробки, навчання моделі та тестування програмної системи;
   5) критерії оцінки ефективності моделі: точність виявлення аномалій, час реагування, обчислювальні витрати;
   6) обмеження щодо обсягу даних.
4. Перелік питань, які потрібно розробити:
   1) Теоретичний аналіз сучасних стратегій прогнозного обслуговування та методів моніторингу стану промислового обладнання.
   2) Дослідження та підготовка даних з датчиків вібрації та температури, включаючи методи інженерії ознак (FFT, вейвлет-перетворення).
   3) Обґрунтування та розробка моделі машинного навчання для раннього виявлення аномалій та прогнозування відмов.
   4) Програмна реалізація системи моніторингу у вигляді веб-застосунку з відповідною архітектурою та інтерфейсом.
   5) Експериментальна оцінка ефективності моделі на різних наборах даних та тестування швидкодії всієї системи

4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1.	Формулювання наукової проблеми, обґрунтування актуальності, мети та завдань дослідження. Визначення об'єкта, предмета та методів дослідження	02.10.2025 – 18.10.2025
2.	Аналіз наукових досліджень стратегій обслуговування, причин відмов обладнання та методів вібраційного моніторингу	18.10.2025 – 02.12.2025
3.	Огляд і порівняльний аналіз наявних методів машинного навчання та існуючих систем прогнозного обслуговування	02.12.2025 – 20.12.2025
4.	Опис джерел даних, характеристик часових рядів та проведення попередньої обробки сигналів	02.12.2025 – 10.01.2026
5.	Виконання інженерії ознак (FFT, вейвлет-перетворення) та формування навчальних і тестових вибірок	10.01.2026 – 20.01.2026
6.	Обґрунтування вибору архітектури нейронної мережі	20.01.2026 – 08.02.2026
7.	Проєктування архітектури нейронної мережі, вибір функцій втрат та оптимізація гіперпараметрів	12.02.2026 – 18.02.2026
8.	Розробка архітектури програмного застосунку, веб-інтерфейсу та реалізація функціоналу моніторингу	18.02.2026 – 26.02.2026
9.	Тестування моделі на даних, оцінка швидкодії та формування інструкції з розгортання	10.03.2026 – 21.03.2026
10.	Оцінка ефективності розробленої моделі за критеріями	21.03.2026 – 30.03.2026
11.	Формування практичних рекомендацій щодо вдосконалення моделі та її адаптації до умов реального використання	08.04.2026 – 15.04.2026
12.	Оформлення пояснювальної записки кваліфікаційної роботи	15.04.2026 – 30.04.2026
13.	Оформлення звіту про переддипломну практику	30.04.2026 – 10.05.2026
14.	Передзахист кваліфікаційної роботи	10.05.2026 – 20.05.2026
15.	Остаточне оформлення пояснювальної записки кваліфікаційної роботи для проходження перевірки на антиплагіат	до 25.05.2026
16.	Отримання відгуку керівника, підготовка висновків та подання роботи на рецензування	до 30.05.2026

5. Дата видачі завдання ***13 жовтня 2025 року.***

Здобувач вищої освіти _____ Олександр САВОЧКА
підпис ініціали, прізвище

Керівник роботи _____ Ольга МОРОЗ
підпис ініціали, прізвище

Додаток Б

Затверджую

_____

«_____»_____2026 р.

Технічне завдання
на розробку програмного виробу

«Система раннього виявлення несправностей промислового обладнання на основі аналізу вібраційних та температурних часових рядів із використанням глибинного навчання та веб-застосунку»

1. Вступ	**1.1. Назва:** Система раннього виявлення несправностей промислового обладнання на основі аналізу вібраційних та температурних часових рядів із використанням глибинного навчання та веб-застосунку. **1.2. Галузь застосування:** комп'ютерно-інтегровані технології, системи технічної діагностики, промисловий моніторинг (IIoT).
2. Підстава для розробки	2.1. Навчальний план за спеціальністю 123 «Комп'ютерна інженерія». 2.2. Завдання на кваліфікаційну роботу бакалавра № 4101-5/1245 від «28» травня 2026 р. (Додаток А до пояснювальної записки).
3. Призначення розробки	**3.1. Мета розробки:** створення та програмна реалізація інтелектуальної системи раннього виявлення аномалій і прогнозування технічного стану обладнання на основі аналізу часових рядів датчиків із застосуванням моделі глибинного навчання типу CNN-LSTM. **3.2. Призначення розробки:** система призначена для своєчасного виявлення відхилень у роботі промислового обладнання, попередження відмов, зменшення простоїв та підвищення надійності виробничого процесу шляхом автоматизованого аналізу даних датчиків. **3.3. Вихідні дані розробки:** історичні часові ряди показників вібрації та температури з датчиків обладнання у форматі CSV/Excel/JSON.

4. Технічні вимоги до програмного виробу	**4.1. Вимоги до функціональних характеристик** • Імпорт та попередня обробка даних датчиків (очищення, нормалізація, обробка пропусків); • Формування часових вікон (sliding window) для аналізу; • Реалізація моделі глибинного навчання **CNN-LSTM** для аналізу часових рядів; • Виявлення аномалій у роботі обладнання; • Візуалізація результатів аналізу у веб-інтерфейсі (графіки, повідомлення, журнал подій); • Збереження результатів аналізу у файл (CSV/JSON). **4.2. Вимоги до надійності** • Коректна робота за наявності пропусків у даних; • Стійкість до шумів у сигналах датчиків; • Час обробки нової порції даних — не більше 20 секунд. **4.3. Вимоги до умов експлуатації** • ОС: Windows 10/11 або Linux (Ubuntu); • Середовище виконання: Python 3.10+. **4.4. Вимоги до складу і параметрів технічних засобів** • ОЗП: не менше 8 GB; • Вільний дисковий простір: не менше 10 GB. **4.5. Вимоги до інформаційної та програмної сумісності** • Бібліотеки: Pandas, NumPy, TensorFlow/Keras, Scikit-learn, FastAPI; • Формати даних: CSV, XLSX, JSON. **4.6. Вимоги до маркування та упаковки** Не висуваються (електронне розповсюдження). **4.7. Вимоги до транспортування і зберігання** Зберігання у хмарному репозиторії (GitHub/GitLab) або на локальному сервері. **4.8. Спеціальні вимоги** Дотримання конфіденційності виробничих даних.
5. Вимоги до програмної документації	Програмною документацією вважати: 1. Дане технічне завдання (Додаток А); 2. Опис математичної моделі CNN-LSTM та алгоритмів обробки даних (розділ 2); 3. Опис програмної реалізації та інструкцію користувача (розділ 4).
6. Вимоги до техніко-еконо-	1. Зменшення часу простою обладнання за рахунок раннього виявлення несправностей; 2. Підвищення надійності контролю технічного стану;

<table>
<tr><td>мічних показників</td><td colspan="2">3. Використання виключно Open Source програмних засобів.</td></tr>
<tr><td rowspan="12">7. Стадії і етапи розробки</td><td>Дата</td><td>Назва етапу</td></tr>
<tr><td>01.10.2025 – 18.10.2025</td><td>Затвердження теми та аналіз предметної області технічної діагностики</td></tr>
<tr><td>18.10.2025 – 02.12.2025</td><td>Огляд методів аналізу часових рядів та моделей глибинного навчання</td></tr>
<tr><td>02.12.2025 – 20.12.2025</td><td>Обґрунтування вибору моделі CNN-LSTM та підготовка даних</td></tr>
<tr><td>02.12.2025 – 10.01.2026</td><td>Розробка математичної моделі</td></tr>
<tr><td>10.01.2026 – 20.01.2026</td><td>Проєктування архітектури програмної системи</td></tr>
<tr><td>20.01.2026 – 08.02.2026</td><td>Програмна реалізація модулів обробки даних та моделі</td></tr>
<tr><td>10.03.2026 – 21.03.2026</td><td>Реалізація веб-інтерфейсу</td></tr>
<tr><td>21.03.2026 – 30.03.2026</td><td>Тестування системи та оцінка метрик</td></tr>
<tr><td>08.04.2026 – 15.04.2026</td><td>Підготовка тексту пояснювальної записки</td></tr>
<tr><td>15.04.2026 – 30.04.2026</td><td>Оформлення матеріалів роботи</td></tr>
<tr><td>10.05.2026 – 30.05.2026</td><td>Передача роботи керівнику та рецензенту</td></tr>
<tr><td>8. Порядок контролю і приймання програмного продукту</td><td colspan="2">1. Контроль виконання згідно календарного плану;
2. Захист на засіданні Екзаменаційної комісії;
3. Подання пояснювальної записки у паперовому та електронному вигляді.</td></tr>
</table>

Виконавець:

студент групи КІ-41

САВОЧКА О.І._____

Замовник:

PH.D., доцент каф.КСР

МОРОЗ О. Ю_____

Додаток В

Програма і методика випробувань програмного виробу «Система раннього виявлення несправностей промислового обладнання на основі аналізу вібраційних та температурних часових рядів із використанням глибинного навчання та веб-застосунку»

1. Об'єкт випробувань

1. **Назва програмного виробу:** «Система раннього виявлення несправностей промислового обладнання на основі аналізу вібраційних та температурних часових рядів із використанням глибинного навчання та веб-застосунку».
2. **Галузь застосування:** промислові підприємства, системи моніторингу технічного стану обладнання, SCADA/IIoT середовища, інформаційно-вимірювальні системи.
3. Відомості запозичуються з відповідних розділів Технічного завдання (Додаток Б).

2. Мета випробувань

Метою кваліфікаційної роботи є підвищення ефективності та надійності функціонування промислового обладнання шляхом створення системи раннього виявлення відмов на основі аналізу вібраційно-температурних часових рядів із використанням методів глибокого навчання та практичної реалізації результатів у вигляді програмного веб-застосунку.

3. Загальні положення

1. **Підстава для проведення випробувань:** наказ по університету про створення атестаційної комісії з проведення захисту кваліфікаційних робіт бакалаврів від **№ 4101-5/1245 від 28.05.2026 року.**
2. **Місце і тривалість випробувань:** комп'ютерний клас кафедри комп'ютерних систем та робототехніки, **10–20 квітня 2026 р.**
3. **Обсяг випробувань:** у межах даної програми і методики.
4. **Учасники випробувань:** атестаційна комісія за участю розробника та керівника кваліфікаційної роботи.

4. Вимоги до програмного виробу

Програмний виріб повинен:

1. Працювати під керуванням ОС **Windows / Linux**;
2. Коректно зчитувати та обробляти дані з датчиків у форматах CSV/XLSX/JSON;
3. Виконувати попередню обробку часових рядів (нормалізація, усунення пропусків, фільтрація шуму);
4. Реалізовувати модель глибинного навчання типу **CNN-LSTM** для аналізу часових рядів;
5. Забезпечувати виявлення аномалій у режимі, наближеному до реального часу;
6. Відображати результати аналізу у веб-інтерфейсі (графіки, повідомлення про аномалії, журнали подій);
7. Час аналізу нової порції даних не повинен перевищувати **20 секунд**;
8. Мати логічно структуровану модульну архітектуру (препроцесинг, модель, модуль детекції, веб-частина);
9. Відповідати вимогам до технічних засобів, зазначених у ТЗ.

5. Вимоги до програмної документації

До програмної документації належать:

1. Технічне завдання (Додаток А);
2. Програма і методика випробувань (цей додаток);
3. Інструкція користувача (розділ 3 пояснювальної записки).

6. Засоби і порядок випробувань

6.1. Засоби випробувань

Персональний комп'ютер із встановленим:

- Python 3.10+
- Бібліотеки: Pandas, NumPy, TensorFlow/Keras, Scikit-learn, FastAPI
- Середовище виконання: Jupyter Notebook або консоль

6.2. Порядок проведення випробувань

Випробування проводяться у два етапи:

- перевірка документації;
- функціональне тестування програмного виробу.

Запуск системи здійснюється виконанням файлу:

uvicorn app.main:app

або через докер docker compose up --build.

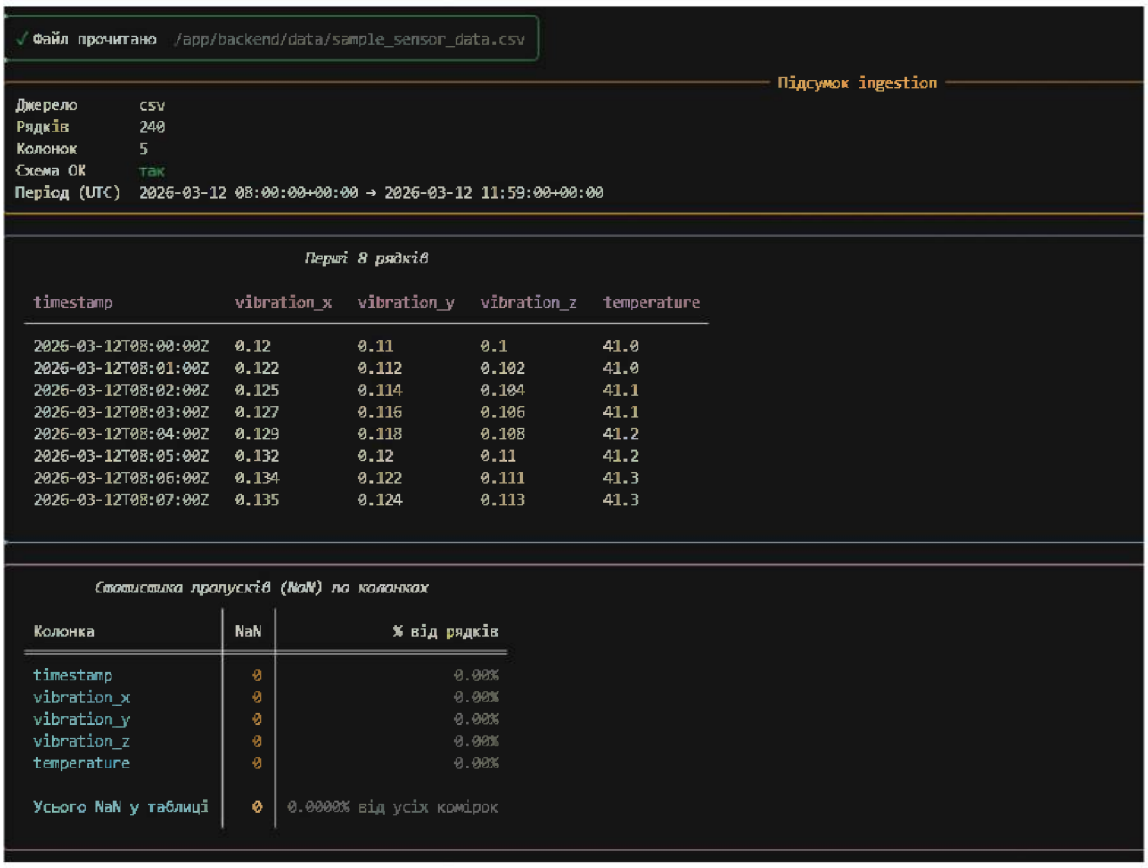
Функціональні тести

Тест 1. Завантаження та підготовка даних

Мета: перевірка коректності зчитування та обробки «сирих» даних датчиків.

Дія: користувач вказує шлях до файлу sensor_data.csv.

Очікуваний результат: система повідомляє про успішне завантаження, відображає перші рядки таблиці та статистику пропусків.



✓ Файл прочитано /app/backend/data/sample_sensor_data.csv

Підсумок ingestion

Джерело	csv
Рядків	240
Колонок	5
Схема ОК	так
Період (UTC)	2026-03-12 08:00:00+00:00 → 2026-03-12 11:59:00+00:00

Перші 8 рядків

timestamp	vibration_x	vibration_y	vibration_z	temperature
2026-03-12T08:00:00Z	0.12	0.11	0.1	41.0
2026-03-12T08:01:00Z	0.122	0.112	0.102	41.0
2026-03-12T08:02:00Z	0.125	0.114	0.104	41.1
2026-03-12T08:03:00Z	0.127	0.116	0.106	41.1
2026-03-12T08:04:00Z	0.129	0.118	0.108	41.2
2026-03-12T08:05:00Z	0.132	0.12	0.11	41.2
2026-03-12T08:06:00Z	0.134	0.122	0.111	41.3
2026-03-12T08:07:00Z	0.135	0.124	0.113	41.3

Статистика пропусків (NaN) по колонках

Колонка	NaN	% від рядків
timestamp	0	0.00%
vibration_x	0	0.00%
vibration_y	0	0.00%
vibration_z	0	0.00%
temperature	0	0.00%
Усього NaN у таблиці	0	0.0000% від усіх комірок

Рисунок В.1 – Результат виконання тесту 1.

Тест 2. Робота моделі CNN-LSTM

Мета: перевірка коректності роботи модуля глибинного навчання.

Дія: запуск процедури аналізу на історичних даних.

Очікуваний результат: відображення графіка «Прогноз / Факт», значень Loss та метрик точності.

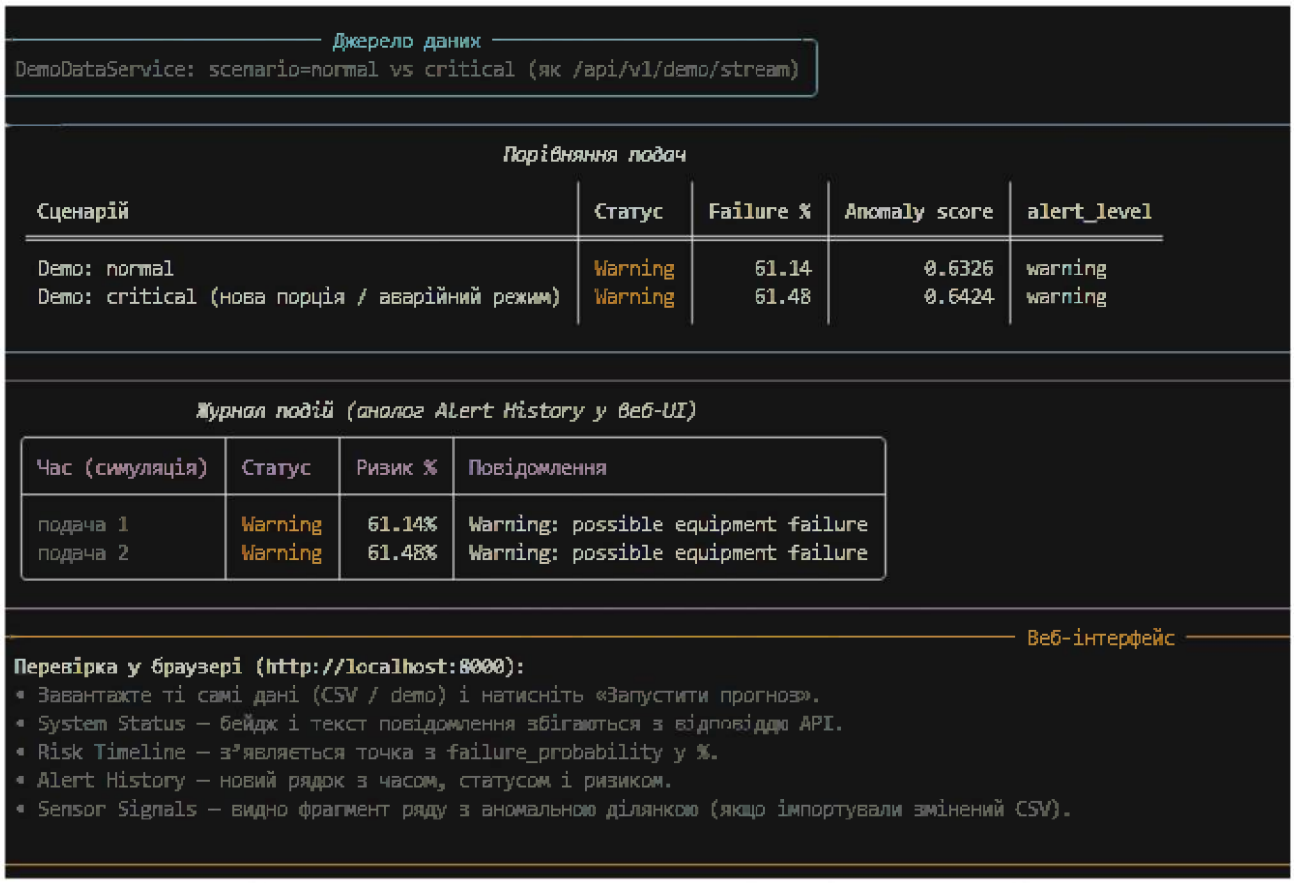


Рисунок В.2 – Результат виконання тесту 2 (Робота моделі CNN-LSTM).

Тест 3. Виявлення аномалій та відображення у веб-інтерфейсі

Мета: перевірка роботи модуля детекції аномалій.

Дія: подача нової порції даних до системи.

Очікуваний результат: поява повідомлення про аномалію, відображення відповідного фрагмента графіка та запис у журналі подій.

Рисунок В.3 – Результат виконання тесту 3.

Висновки

Тест 1, Тест 2 та Тест 3 пройдені успішно. Програмний виріб функціонує відповідно до вимог Технічного завдання. Випробування вважаються такими, що пройдені успішно.

Виконавець: студент групи КІ-41, **О. І. Савочка**_____

Керівник кваліфікаційної роботи: доц.каф.КСР **О. Ю. Мороз**_____

Додаток Г

Приклад коду програми роботи із моделями глибокого навчання

```
from __future__ import annotations
import os
import numpy as np
from sklearn.metrics import f1_score, precision_score, recall_score, roc_auc_score
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.utils.class_weight import compute_class_weight
from app.schemas import TrainRequest, TrainResponse, TrainingCurves
os.environ.setdefault("TF_CPP_MIN_LOG_LEVEL", "2")
try:
    import tensorflow as tf
    from tensorflow import keras
    from tensorflow.keras import layers
except Exception:  # pragma: no cover
    tf = None  # type: ignore[assignment]
    keras = None  # type: ignore[assignment]
    layers = None  # type: ignore[assignment]
# Порядок каналів у тензорі має збігатися з JSON (вікна з dataset/build).
_SENSOR_ORDER = ("vibration_x", "vibration_y", "vibration_z", "temperature")
class TrainingService:
    """Зберігає останню навчену модель у `model_registry` під ключем
`cnn_lstm:latest`."""
    def __init__(self) -> None:
        self.model_registry: dict[str, object] = {}
        # Завантаження збереженої моделі при старті, якщо існує
        models_dir = os.path.join(os.path.dirname(__file__), "..", "..", "models")
        model_path = os.path.join(models_dir, "cnn_lstm_latest.keras")
        if tf is not None and keras is not None and os.path.exists(model_path):
            try:
                loaded_model = keras.models.load_model(model_path)
                self.model_registry["cnn_lstm:latest"] = {
                    "model": loaded_model,
                    "scaler": None,
                    "sequence_length": None,
                    "n_features": None,
                }
            except Exception:
                pass  # Якщо модель пошкоджена — просто ігноруємо
    def train_classifier(self, request: TrainRequest) -> TrainResponse:
```

```
        if tf is None or keras is None or layers is None:
            raise ValueError("TensorFlow is required for CNN-LSTM training. Install the 'tensorflow' package.")
        tf.keras.utils.set_random_seed(42)
        # x: (N, T, 4) — N зразків, T кроків у вікні, 4 сенсори
        x = self._windows_to_array(request.windows)
        y = np.array(request.labels, dtype=np.float32)
        _, seq_len, n_features = x.shape
        class_counts = np.bincount(y.astype(int), minlength=2)
        if np.any(class_counts == 0):
            raise ValueError("Both classes (0 and 1) must be present in labels")
        if np.any(class_counts < 2):
            raise ValueError("Each class must contain at least 2 samples for a stable split")
        x_train, x_val, y_train, y_val = train_test_split(
            x,
            y,
            test_size=0.2,
            shuffle=True,
            random_state=42,
            stratify=y.astype(int),
        )
        # Один scaler на всі кроки: fit лише на train, той самий перетворення для val
        scaler = StandardScaler()
        x_train_2d = x_train.reshape(-1, n_features)
        scaler.fit(x_train_2d)
        x_train = scaler.transform(x_train_2d).reshape(x_train.shape[0], seq_len, n_features)
        x_val = scaler.transform(x_val.reshape(-1, n_features)).reshape(x_val.shape[0], seq_len, n_features)
        # Збалансування без oversampling послідовностей — лише ваги класів у fit()
        class_weight = None
        if request.balancing == "class_weight":
            classes = np.array([0, 1])
            weights = compute_class_weight("balanced", classes=classes, y=y_train.astype(int))
            class_weight = {int(c): float(w) for c, w in zip(classes, weights)}
        # CNN витягує локальні шаблони по часу; LSTM — часову динаміку; sigmoid — P(клас 1)
        model = self._build_cnn_lstm((seq_len, n_features))
        callbacks = [
```

```
            keras.callbacks.EarlyStopping(monitor="val_loss", patience=8,
restore_best_weights=True)
        ]
        batch_size = max(8, min(32, len(x_train)))
        fit_result = model.fit(
            x_train,
            y_train,
            validation_data=(x_val, y_val),
            epochs=80,
            batch_size=batch_size,
            class_weight=class_weight,
            callbacks=callbacks,
            verbose=0,
        )
        # Keras History → криві для графіків у відповіді API та на фронті
        training_curves = self._history_to_training_curves(fit_result.history)
        # Метрики на валідації: поріг 0.5 для бінарного рішення
        y_score = model.predict(x_val, verbose=0).reshape(-1)
        y_pred = (y_score >= 0.5).astype(int)
        metrics = {
            "precision": float(precision_score(y_val, y_pred, zero_division=0)),
            "recall": float(recall_score(y_val, y_pred, zero_division=0)),
            "f1_score": float(f1_score(y_val, y_pred, zero_division=0)),
            "roc_auc": float(roc_auc_score(y_val, y_score)) if len(np.unique(y_val)) > 1
else 0.0,
        }
        key = "cnn_lstm:latest"
        self.model_registry[key] = {
            "model": model,
            "scaler": scaler,
            "sequence_length": seq_len,
            "n_features": n_features,
        }
        # Збереження моделі на диск для відновлення після перезапуску
        models_dir = os.path.join(os.path.dirname(__file__), "..", "..", "models")
        os.makedirs(models_dir, exist_ok=True)
        model.save(os.path.join(models_dir, "cnn_lstm_latest.keras"))

        unique, counts = np.unique(y_train.astype(int), return_counts=True)
        distribution = {str(int(label)): int(count) for label, count in zip(unique, counts)}
        return TrainResponse(
            model_type=request.model_type,
            balancing=request.balancing,
```

```
            feature_count=n_features,
            sequence_length=seq_len,
            train_size=int(len(x_train)),
            validation_size=int(len(x_val)),
            class_distribution=distribution,
            metrics=metrics,
            training_curves=training_curves,
        )
    @staticmethod
    def _history_to_training_curves(history: dict) -> TrainingCurves:
        """Перетворює `model.fit(...).history` у структуру для REST (епохи 1..n)."""
        loss = [float(x) for x in history["loss"]]
        val_loss = [float(x) for x in history["val_loss"]]
        acc_key = "accuracy" if "accuracy" in history else "acc"
        val_acc_key = "val_accuracy" if "val_accuracy" in history else "val_acc"
        accuracy = [float(x) for x in history[acc_key]]
        val_accuracy = [float(x) for x in history[val_acc_key]]
        # На випадок рідкісних розбіжностей довжин — беремо спільний префікс
        n = min(len(loss), len(val_loss), len(accuracy), len(val_accuracy))
        epoch = list(range(1, n + 1))
        return TrainingCurves(
            epoch=epoch,
            loss=loss[:n],
            val_loss=val_loss[:n],
            accuracy=accuracy[:n],
            val_accuracy=val_accuracy[:n],
        )
    @staticmethod
    def _windows_to_array(windows: list[list[dict[str, float]]]) -> np.ndarray:
        """Кожне вікно → матриця (timesteps × 4), усі вікна → np.stack → (N, T,
4)."""
        rows: list[np.ndarray] = []
        for window in windows:
            mat = np.array([[float(step[k]) for k in _SENSOR_ORDER] for step in
window], dtype=np.float32)
            rows.append(mat)
        return np.stack(rows, axis=0)
    @staticmethod
    def _build_cnn_lstm(input_shape: tuple[int, int]):
        # input_shape = (T, 4): вісь часу — остання перед LSTM після згорток/pool
        assert keras is not None and layers is not None
        model = keras.Sequential(
            [
```

```
        layers.Input(shape=input_shape),
        layers.Conv1D(32, kernel_size=3, padding="same"),
        layers.BatchNormalization(),
        layers.Activation("relu"),
        layers.Conv1D(64, kernel_size=3, padding="same"),
        layers.BatchNormalization(),
        layers.Activation("relu"),
        layers.MaxPooling1D(pool_size=2),
        layers.LSTM(64),
        layers.Dropout(0.25),
        layers.Dense(32, activation="relu"),
        layers.Dense(1, activation="sigmoid"),
    ]
)
model.compile(
    optimizer=keras.optimizers.Adam(learning_rate=1e-3),
    loss="binary_crossentropy",
    metrics=["accuracy"],
)
return model
```