ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Інтелектуальні методи та алгоритми діагностики складних технічних систем»

на здобуття освітнього ступеня бакалавра
зі спеціальності 124 Системний аналіз
(код,найменування спеціальності)
освітньо-професійної програми Системний аналіз
(назва)

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

Ярослав ЛЕЩЕНКО
(підпис) *(ім'я, ПРІЗВИЩЕ здобувача)*

Виконав: здобувач вищої освіти група САД-41
Ярослав ЛЕЩЕНКО
(ім'я, ПРІЗВИЩЕ)
Керівник Ровіл НАФЄЄВ
к.ф.-м.н., доцент *(ім'я, ПРІЗВИЩЕ)*

Рецензент:
(ім'я, ПРІЗВИЩЕ)

Київ 2026

ДЕРЖАВНИЙ УНІВЕРСИТЕТ

ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти Бакалавр

Спеціальність 124 Системний аналіз

Освітньо-професійна програма Системний аналіз

ЗАТВЕРДЖУЮ
Завідувач кафедрою ІСТ

Каміла СТОРЧАК

“____” _____2026 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Лещенко Ярослав Ігорович

(прізвище, ім’я, по батькові здобувача)

1.Тема кваліфікаційної роботи: Інтелектуальні методи та алгоритми діагностики складних технічних систем

Керівник кваліфікаційної роботи: Ровіл НАФЄЄВ, доцент кафедри інформаційних систем та технологій, кандидат фізико-математичних наук**.**

затверджені наказом Державного університету інформаційно-комунікаційних технологій від 20 лютого 2026 р. №51

2. Строк подання кваліфікаційної роботи «01» ___червня___2026 р.

3. Вихідні дані кваліфікаційної роботи:

1. Принципи побудови, функціонування та моніторингу складних технічних систем.
2. Методи, алгоритми та метрики оцінювання моделей машинного навчання для задач класифікації та технічної діагностики.
3. Науково-технична література, статистичні дані та результати комп'ютерного моделювання передвідмовних станів обладнання.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

1. Аналіз об'єкта дослідження та методів технічної діагностики: формалізація поняття складної технічної системи, системний аналіз станів системи (від нормального до критичного) та огляд сучасних інтелектуальних методів моніторингу.
2. Розробка інтелектуальних моделей та алгоритмів діагностики: математичний опис багатовимірних діагностичних ознак, обґрунтування вибору методів машинного навчання (Logistic Regression, kNN, Decision Tree, Random Forest) та проєктування архітектури системи за допомогою UML-діаграм.
3. Експериментальне дослідження та програмна реалізація: розробка програмного забезпечення на мові Python, моделювання незбалансованих даних, порівняльний аналіз ефективності алгоритмів за метриками точності та дослідження важливості ознак.

5. Перелік ілюстраційного матеріалу:

Презентація (демонстраційні слайди), графіки порівняння ефективності моделей, UML-діаграми архітектури програмного комплексу.

6. Дата видачі завдання:

«20» ___лютого____ 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Проведення системного аналізу об'єкта дослідження, збір первинних технічних даних та огляд науково-методичної літератури за темою інтелектуальної діагностики	20.02.2026 – 08.03.2026	
2.	Формалізація математичного опису багатовимірних діагностичних параметрів та побудова графа переходів між станами складної технічної системи	09.03.2026 – 29.03.2026	
3.	Проєктування загальної архітектури програмного аналітичного комплексу та побудова відповідних структурних UML-діаграм класів і прецедентів	30.03.2026 – 15.04.2026	
4.	Алгоритмізація та програмна реалізація моделей класифікації технічного стану (Logistic Regression, kNN, Decision Tree, Random Forest) мовою Python	16.04.2026 – 03.05.2026	
5.	Комп'ютерне моделювання незбалансованих імітаційних даних, проведення експериментів, оцінювання метрик ефективності та аналіз важливості ознак	04.05.2026 – 15.05.2026	
6.	Оформлення кваліфікаційної роботи	16.05.2026 – 22.05.2026	
7.	Подача роботи керівнику	23.05.2026	

Здобувач вищої освіти . Ярослав ЛЕЩЕНКО

(підпис) *(ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи . . Ровіл НАФЄЄВ

(підпис) *(ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Кваліфікаційна робота на тему: «Інтелектуальні методи та алгоритми діагностики складних технічних систем».

Пояснювальна записка становить 77 с., містить 23 рисунки, 14 таблиць, 2 додатки та 30 використаних джерел.

Метою роботи є дослідження інтелектуальних методів та алгоритмів діагностики складних технічних систем і розроблення підходу до визначення технічного стану системи на основі аналізу діагностичних даних.

Об’єктом дослідження є процес діагностики складних технічних систем. Предметом дослідження є інтелектуальні методи та алгоритми визначення технічного стану складних технічних систем на основі аналізу діагностичних параметрів.

У роботі досліджено підходи до діагностики складних технічних систем із використанням методів системного аналізу та інтелектуальних алгоритмів обробки даних. Актуальність теми зумовлена тим, що сучасні технічні системи мають складну структуру, велику кількість параметрів і працюють в умовах невизначеності, тому своєчасне виявлення несправностей є важливим для підвищення надійності та безпеки їх експлуатації.

Для експериментального дослідження реалізовано алгоритм класифікації технічного стану з використанням моделей машинного навчання Logistic Regression, kNN, Decision Tree та Random Forest. За результатами порівняння найвищу загальну точність показала модель Logistic Regression, а модель Random Forest використано для аналізу важливості діагностичних ознак.

Ключові слова: складна технічна система, системний аналіз, технічна діагностика, діагностичні параметри, інтелектуальні методи, машинне навчання, Logistic Regression, Random Forest.

ЗМІСТ

ВСТУП 8
РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ СКЛАДНИХ ТЕХНІЧНИХ СИСТЕМ ЯК ОБ'ЄКТІВ ДІАГНОСТИКИ 10
1.1. Поняття складної технічної системи та її основні властивості 10
1.2. Особливості функціонування складних технічних систем в умовах невизначеності 13
1.3. Діагностичні параметри, стани та ознаки несправностей технічних систем 17
1.4. Аналіз сучасних підходів до діагностики складних технічних систем 20
1.5. Порівняльний аналіз традиційних та інтелектуальних методів діагностики 24
Висновки до розділу 1 29
РОЗДІЛ 2. РОЗРОБКА СИСТЕМНОЇ МОДЕЛІ ТА АЛГОРИТМУ ІНТЕЛЕКТУАЛЬНОЇ ДІАГНОСТИКИ 30
2.1. Постановка задачі діагностики технічного стану складної системи 30
2.2. Формування множини вхідних параметрів і діагностичних ознак 34
2.3. Побудова моделі станів технічної системи 39
2.4. Вибір інтелектуальних методів для розв'язання задачі діагностики 43
2.5. Розробка алгоритму інтелектуальної діагностики технічної системи 46
2.6. Критерії оцінювання ефективності алгоритму діагностики 51
Висновки до розділу 2 55
РОЗДІЛ 3. ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ АЛГОРИТМУ ДІАГНОСТИКИ СКЛАДНОЇ ТЕХНІЧНОЇ СИСТЕМИ 57
3.1. Опис програмної реалізації алгоритму діагностики 57
3.2. Формування та підготовка експериментального набору даних 59
3.3. Навчання та тестування інтелектуальних моделей діагностики 63
3.4. Порівняльний аналіз результатів роботи алгоритмів 67
3.5. Оцінювання ефективності запропонованого підходу 69
3.6. Рекомендації щодо використання алгоритму в системах технічного моніторингу 72
Висновки до розділу 3 74
ВИСНОВКИ 76
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 78
ДОДАТКИ 82
ДЕМОНСТРАЦІЙНИЙ МАТЕРІАЛ (Презентація) 90

ВСТУП

Складні технічні системи нашого часу використовуються в енергетиці, промисловості, транспорті, телекомунікаціях, автоматизованому виробництві та інших сферах, де важливими є надійність, безперервність роботи й своєчасне виявлення несправностей. Такі системи складаються з великої кількості взаємопов’язаних елементів, працюють у змінних умовах і можуть мати багато параметрів, які впливають на їхній технічний стан. Відмова навіть одного елемента може призвести до порушення роботи всієї системи, економічних втрат, простою обладнання або виникнення аварійної ситуації.

Актуальність теми зумовлена потребою у створенні таких підходів до діагностики, які дозволяють обробляти великі обсяги технічних даних, знаходити приховані закономірності, визначати ознаки несправностей і підтримувати прийняття рішень щодо подальшої експлуатації обладнання. Традиційні методи контролю часто ґрунтуються на регламентних перевірках, досвіді фахівців або аналізі окремих параметрів. Проте для складних технічних систем цього може бути недостатньо, оскільки їхній стан формується під впливом багатьох взаємозалежних факторів. У таких умовах доцільним є використання інтелектуальних методів, зокрема алгоритмів машинного навчання, класифікації, прогнозування та аналізу діагностичних ознак [4; 10; 15; 23].

З позиції системного аналізу діагностика технічної системи розглядається як процес дослідження об’єкта, визначення його параметрів, виявлення зв’язків між ознаками та формування висновку про поточний або можливий майбутній стан. Для цього необхідно визначити вхідні параметри системи, можливі стани, критерії оцінювання, алгоритм обробки даних і спосіб перевірки ефективності запропонованого підходу [7; 8; 21].

Метою роботи є дослідження інтелектуальних методів та алгоритмів діагностики складних технічних систем і розроблення підходу до визначення технічного стану системи на основі аналізу діагностичних даних.

Для досягнення поставленої мети необхідно вирішити наступні **завдання**:

– проаналізувати складні технічні системи як об’єкти діагностики;

– дослідити традиційні та інтелектуальні методи виявлення несправностей;

– розробити системну модель діагностики технічного стану;

– сформувати алгоритм інтелектуальної діагностики на основі діагностичних ознак;

– провести експериментальне дослідження роботи алгоритму та оцінити його ефективність за визначеними критеріями.

Об’єктом дослідження є процес діагностики складних технічних систем.

Предметом дослідження є інтелектуальні методи та алгоритми визначення технічного стану складних технічних систем на основі аналізу діагностичних параметрів.

Методи дослідження включають методи системного аналізу, аналіз науково-технічних джерел, порівняльний аналіз методів діагностики, математичне моделювання, методи машинного навчання, класифікацію, оцінювання точності моделей та експериментальну перевірку результатів.

Технічна новизна роботи полягає в тому, що удосконалено підхід до діагностики складних технічних систем за рахунок поєднання системного аналізу, моделі станів технічної системи та інтелектуальних методів класифікації діагностичних даних, що дало змогу формалізувати процес визначення технічного стану й підвищити обґрунтованість діагностичного рішення.

Практичне значення отриманих результатів полягає в можливості використання запропонованого підходу як основи для створення програмного засобу підтримки технічної діагностики. Це може бути корисним для підприємств, сервісних служб, інженерів з експлуатації обладнання та фахівців, які займаються моніторингом стану технічних систем, прогнозуванням відмов і прийняттям рішень щодо обслуговування обладнання.

РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ СКЛАДНИХ ТЕХНІЧНИХ СИСТЕМ ЯК ОБ'ЄКТІВ ДІАГНОСТИКИ

1.1. Поняття складної технічної системи та її основні властивості

Складні технічні системи є важливою частиною сучасної промисловості, енергетики, транспорту, зв'язку, автоматизованого виробництва та інших сфер. До таких систем можна віднести енергетичне обладнання, виробничі лінії, телекомунікаційні комплекси, системи керування технологічними процесами, промислових роботів, транспортні механізми, автоматизовані системи контролю та інші технічні об'єкти. Їх об'єднує те, що вони складаються з багатьох елементів, мають складну структуру зв'язків і працюють у змінних умовах.

У працях Ю. М. Кузнєцова та Б. І. Придального технічна система розглядається як сукупність взаємопов'язаних елементів, які разом виконують певну функцію та утворюють цілісний технічний об'єкт [3]. Це є зручним для системного аналізу, тому що дозволяє досліджувати не тільки окремі елементи, а й зв'язки між ними. Наприклад, у виробничій лінії важливими є не лише двигуни, датчики або контролери окремо, а й те, як вони взаємодіють між собою під час виконання технологічного процесу.

З позиції системного аналізу складну технічну систему можна визначити як технічний об'єкт, який складається з багатьох взаємопов'язаних компонентів, має певну мету функціонування, взаємодіє із зовнішнім середовищем і може перебувати в різних технічних станах. У навчальному посібнику Г. П. Ситника, Л. Г. Комахи та А. О. Рудика системний аналіз подається як підхід, що дозволяє досліджувати об'єкт через його структуру, функції, зв'язки та поведінку [8]. Для складних технічних систем це особливо важливо, оскільки несправність одного елемента часто впливає на роботу всієї системи.

Основними ознаками складної технічної системи є багатокомпонентність, взаємозалежність елементів, наявність вхідних і вихідних параметрів, зміна стану в часі, вплив зовнішніх факторів, а також можливість виникнення різних

видів відмов. Наприклад, у системі зв’язку зміна якості сигналу може бути пов’язана не тільки з несправністю обладнання, а й із перевантаженням мережі, помилками в налаштуваннях, впливом середовища або зношенням окремих елементів [6].

На рис. 1.1 подано узагальнену структуру складної технічної системи.

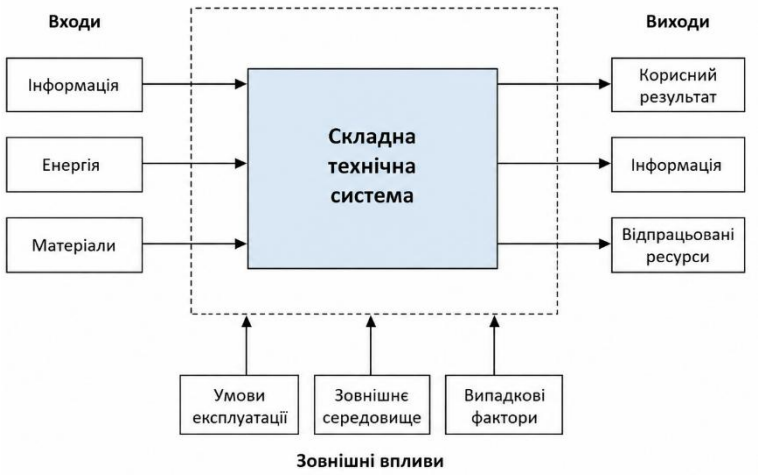


Рисунок 1.1 – Узагальнена структура складної технічної системи

На рис. 1.1 зображено складну технічну систему як сукупність взаємопов’язаних блоків. Вхідні параметри надходять до системи, далі вони обробляються її внутрішніми елементами, після чого формується вихідний результат. Зворотний зв’язок дозволяє оцінити, чи відповідає робота системи заданим вимогам.

Для складних технічних систем характерна наявність кількох рівнів. На нижньому рівні знаходяться фізичні компоненти: датчики, виконавчі механізми, приводи, контролери, кабельні з’єднання, модулі живлення. На середньому рівні можуть бути підсистеми керування, обробки даних, передавання інформації. На верхньому рівні розміщуються системи моніторингу, прийняття рішень і диспетчерського керування. Через це аналіз такої системи не можна зводити тільки до перевірки окремої деталі. Потрібно розуміти, як стан одного елемента впливає на інші частини системи.

А. П. Бовсуновський у навчальному посібнику з надійності енергетичного обладнання наголошує, що для технічних об’єктів важливо враховувати не тільки факт працездатності, а й імовірність безвідмовної роботи, ресурс, ремонтопридатність і стійкість до навантажень [1]. Це означає, що складна технічна система має оцінюватися не тільки за принципом «працює / не працює», а за ширшим набором характеристик. Система може ще виконувати свої функції, але вже мати приховані ознаки майбутньої несправності.

У роботі Л. Ф. Дзюби, М. І. Кусія та О. В. Меньшикової розглядається зв’язок між надійністю технічних систем і техногенним ризиком [2]. Це важливо для діагностики, бо несправність у складній системі може мати різні наслідки. В одних випадках вона призводить тільки до зниження продуктивності, а в інших – до аварійної ситуації, зупинки виробництва або небезпеки для людей. Тому діагностика має враховувати не тільки технічні параметри, а й ризики, які виникають у разі відмови.

Основні властивості складної технічної системи подані в табл. 1.1.

Таблиця 1.1 – Основні властивості складної технічної системи

Властивість	Зміст властивості	Приклад прояву
Багатокомпонентність	Система складається з великої кількості елементів	двигуни, датчики, контролери, кабелі, програмні модулі
Взаємозалежність	Зміна стану одного елемента впливає на інші	помилка датчика може спричинити неправильну роботу контролера
Динамічність	Стан системи змінюється в часі	температура, навантаження, вібрація змінюються під час роботи
Наявність зворотного зв’язку	Система може коригувати свою роботу за результатами контролю	автоматичне регулювання швидкості або температури
Невизначеність	Не всі параметри можуть бути точно відомі	частина несправностей проявляється поступово
Ризик відмови	Система може перейти в несправний або аварійний стан	перегрів, зношення вузла, збій зв’язку

З табл. 1.1 видно, що складна технічна система має багато властивостей, які ускладнюють її діагностику. Звичайна перевірка одного параметра не завжди

дозволяє правильно оцінити стан системи. Наприклад, підвищена температура може бути наслідком перевантаження, поганого охолодження, зношення механічного вузла або помилки в системі керування. Тому потрібен системний підхід, який враховує сукупність ознак.

У системному аналізі важливо виділяти об’єкт, його елементи, зв’язки, мету функціонування, зовнішнє середовище та критерії оцінювання. Якщо застосувати це до задачі діагностики, то складна технічна система розглядається як об’єкт, стан якого потрібно визначити за набором діагностичних параметрів. Такі параметри можуть бути фізичними, інформаційними або експлуатаційними. До фізичних належать температура, тиск, струм, напруга, вібрація. До інформаційних – кількість помилок, затримка сигналу, якість передавання даних. До експлуатаційних – час роботи, кількість циклів, режим навантаження, частота обслуговування.

І. О. Савченко у матеріалах з моделювання складних систем зазначає, що моделювання дозволяє спростити дослідження об’єкта, виділити основні властивості та перейти від реальної системи до її формального опису [7]. Для діагностики це означає, що реальний технічний об’єкт можна подати через модель станів, модель параметрів або модель переходів між станами. Наприклад, технічна система може перебувати у стані нормальної роботи, допустимого відхилення, попередження, критичного стану або відмови.

1.2. Особливості функціонування складних технічних систем в умовах невизначеності

Складні технічні системи рідко працюють в ідеальних і повністю передбачуваних умовах. На їхній стан впливають навантаження, температура середовища, вологість, якість електроживлення, зношення деталей, помилки оператора, якість технічного обслуговування, програмні збої та інші фактори. Частина цих факторів може бути виміряна, але частина залишається невідомою або проявляється тільки після певного часу роботи системи.

Невизначеність у технічних системах означає, що не всі дані про стан системи є повними, точними або своєчасними. Наприклад, датчик може мати похибку, частина параметрів може не вимірюватися, а одна й та сама ознака може відповідати різним видам несправностей. Через це задача діагностики часто не має простого однозначного розв’язку.

У системному аналізі невизначеність розглядається як одна з причин складності прийняття рішень [8]. Для технічної діагностики це проявляється в тому, що фахівець або алгоритм не завжди має повну інформацію про причину відхилення. Наприклад, підвищення вібрації може бути пов’язане з дисбалансом, зношенням підшипника, послабленням кріплення або зміною режиму навантаження. Тому для правильного висновку потрібно аналізувати не одну ознаку, а сукупність параметрів.

Невизначеність може бути кількох видів. По-перше, це невизначеність даних, коли виміряні параметри мають похибки або неповні значення. По-друге, це невизначеність моделі, коли немає точного математичного опису поведінки системи. По-третє, це невизначеність зовнішніх умов, коли навантаження або середовище змінюються випадково. По-четверте, це невизначеність причин несправності, коли одна ознака може відповідати кільком можливим дефектам.

На рис. 1.2 показані основні джерела невизначеності при діагностиці складної технічної системи.

Рисунок 1.2 – Джерела невизначеності під час діагностики технічної системи

На рис. 1.2 зображено чотири групи факторів: неповні дані, похибки вимірювання, зміна умов роботи та складність зв’язків між елементами. У центрі схеми розміщується технічна система, стан якої потрібно визначити.

У сфері надійності технічних систем важливим є поняття відмови. Відмова – це подія, після якої система або її елемент повністю чи частково втрачає здатність виконувати задані функції. Проте в реальній роботі відмова часто не виникає миттєво. Перед нею можуть з’являтися поступові зміни: збільшення температури, поява вібрації, нестабільність сигналу, зростання енергоспоживання, зниження продуктивності. Саме тому важливим є не тільки виявлення відмови, а й розпізнавання передвідмовного стану [1; 2].

IEC 60812:2018 описує підходи FMEA та FMECA, які використовуються для аналізу видів відмов, їхніх причин і наслідків [18]. Це корисний для системного аналізу, бо дозволяє не тільки зафіксувати несправність, а й оцінити, який елемент може її спричинити, наскільки серйозними будуть наслідки та які дії потрібно виконати. У складних технічних системах це особливо важливо, бо одна відмова може запускати ланцюг інших порушень.

Наприклад, у системі автоматизованого керування виробничим процесом несправний датчик температури може передавати неправильні дані контролеру. Контролер, отримуючи помилковий сигнал, може неправильно змінити режим роботи обладнання. У результаті виникає перегрів, зниження якості продукції або аварійне вимкнення. У такому випадку причина проблеми знаходиться не тільки в одному елементі, а в ланцюгу взаємодій між датчиком, контролером і виконавчим механізмом.

Складність функціонування технічних систем також пов’язана з тим, що вони можуть працювати в різних режимах. Наприклад, обладнання може працювати на малому, середньому або високому навантаженні. Один і той самий рівень температури може бути нормальним для одного режиму, але небезпечним для іншого. Тому діагностика повинна враховувати контекст роботи системи.

У технічній експлуатації засобів та систем зв’язку, як зазначають Л. М. Сакович та інші автори, важливим є постійний контроль стану обладнання,

аналіз параметрів і своєчасне технічне обслуговування [6]. Це стосується не тільки систем зв’язку, а й інших технічних об’єктів. Якщо контроль проводиться нерегулярно або за обмеженою кількістю параметрів, ризик пропуску несправності зростає.

Особливістю складних технічних систем є також старіння та зношення елементів. Навіть якщо система працює нормально, її ресурс поступово зменшується. Це означає, що діагностика має враховувати не лише поточні параметри, а й історію роботи системи. Наприклад, два однакові механізми можуть мати однакову температуру та вібрацію в певний момент часу, але якщо один із них працював значно довше або під більшим навантаженням, ризик відмови для нього може бути вищим.

Для роботи в умовах невизначеності все частіше використовують інтелектуальні методи. Вони дозволяють аналізувати багато параметрів одночасно, знаходити приховані залежності та класифікувати стан системи на основі попередніх даних. За результатами огляду M. Fernandes, J. M. Corchado та G. Marreiros, методи машинного навчання активно застосовуються для діагностики механічних несправностей і прогнозування відмов у реальних промислових умовах [15]. Це показує, що інтелектуальні методи не є лише теоретичним інструментом, а можуть використовуватися для практичних задач обслуговування обладнання.

Важливим напрямом є Prognostics and Health Management, або PHM. Цей підхід спрямований на оцінювання поточного стану системи, виявлення відхилень і прогнозування майбутніх відмов. У матеріалах National Institute of Standards and Technology підкреслюється, що PHM є важливим для надійної роботи smart manufacturing, тобто сучасного розумного виробництва [26]. У таких системах рішення про технічне обслуговування бажано приймати не тільки після виникнення поломки, а на основі аналізу даних про стан обладнання.

Отже, складні технічні системи функціонують в умовах неповної інформації, змінних режимів, зовнішніх впливів і поступового зношення. Це створює потребу в діагностиці, яка здатна враховувати багато параметрів і

працювати з невизначеністю. Системний аналіз у такому випадку дозволяє розглядати технічну систему як цілісний об’єкт, а інтелектуальні методи можуть допомогти виявляти закономірності, які складно помітити при звичайному аналізі.

1.3. Діагностичні параметри, стани та ознаки несправностей технічних систем

Діагностика технічної системи починається з визначення параметрів, за якими можна оцінити її стан. Діагностичний параметр – це вимірювана або розрахункова характеристика, яка відображає роботу системи чи її окремого елемента. До таких параметрів можуть належати температура, тиск, вібрація, напруга, струм, швидкість обертання, рівень шуму, час відгуку, кількість помилок, пропускна здатність, навантаження та інші показники.

Діагностична ознака – це зміна або особливість параметра, яка може свідчити про певний технічний стан або несправність. Наприклад, сама температура є діагностичним параметром, а її різке зростання або стабільне перевищення норми є діагностичною ознакою. Аналогічно, вібрація може бути звичайним робочим параметром, але збільшення амплітуди вібрації або зміна її частотного складу може свідчити про механічний дефект.

У навчальному посібнику Р. Р. Обертюха та А. В. Слабкого з математичного моделювання механічних систем показано, що для аналізу механічних об’єктів важливо враховувати параметри руху, навантаження, взаємодію елементів і динамічні характеристики [5]. Для діагностики це означає, що механічна система не повинна оцінюватися тільки за одним показником. Наприклад, для двигуна або підшипника потрібно враховувати вібрацію, температуру, навантаження, швидкість обертання та час роботи.

ISO 13374-1:2003 визначає загальні підходи до обробки, передавання та представлення даних у системах моніторингу стану й діагностики машин [19]. Це джерело важливе тим, що діагностика розглядається не як окрема дія, а як

процес роботи з даними. Спочатку дані отримують від датчиків, потім обробляють, передають, зберігають і подають у вигляді, придатному для аналізу. Така логіка добре відповідає сучасним системам моніторингу.

ISO 13379-2:2015 стосується data-driven applications, тобто підходів, які ґрунтуються на аналізі даних [20]. Це напряму пов’язано з інтелектуальною діагностикою, оскільки алгоритм машинного навчання також працює з діагностичними даними. Він не просто порівнює параметр із порогом, а шукає залежність між набором ознак і станом системи.

У технічній діагностиці можна виділити кілька основних груп параметрів. Вони подані в табл. 1.2.

Таблиця 1.2 – Основні групи діагностичних параметрів технічних систем

Група параметрів	Приклади параметрів	Що дозволяють оцінити
Теплові	температура вузлів, швидкість нагрівання	перегрів, перевантаження, погане охолодження
Механічні	вібрація, шум, швидкість обертання	зношення, дисбаланс, пошкодження механічних елементів
Електричні	напруга, струм, потужність, опір	перевантаження, коротке замикання, нестабільне живлення
Інформаційні	кількість помилок, затримка, втрата пакетів	збої зв’язку, помилки обробки даних
Експлуатаційні	час роботи, кількість циклів, навантаження	ресурс, інтенсивність використання, ризик відмови
Комбіновані	інтегральний показник стану, індекс ризику	загальну оцінку технічного стану

З табл. 1.2 видно, що діагностичні параметри можуть бути різної природи. Тому для складних технічних систем доцільно використовувати не один показник, а набір ознак. Наприклад, для оцінювання стану промислового механізму можуть одночасно використовуватися температура, вібрація, струм двигуна та рівень навантаження. Якщо всі параметри знаходяться в допустимих межах, система працює нормально. Якщо один або кілька параметрів виходять за межі, система може перейти в попереджувальний або критичний стан.

Стан технічної системи – це характеристика її працездатності в певний момент часу. У простому випадку можна виділити два стани: працездатний і непрацездатний. Але для складних систем такого поділу недостатньо. Більш доцільно використовувати кілька станів: нормальний стан, допустиме відхилення, попереджувальний стан, критичний стан і відмова.

На рис. 1.3 зображена модель переходу технічної системи між станами.

Рисунок 1.3 – Модель станів технічної системи

Така модель є зручною для подальшої побудови алгоритму діагностики. Наприклад, алгоритм може не просто визначати, що система несправна, а класифікувати її стан за кількома рівнями. Це дає більше корисної інформації для оператора або інженера. Якщо система перебуває в попереджувальному стані, можна провести профілактику. Якщо вона вже в критичному стані, потрібно швидко зменшити навантаження або зупинити обладнання.

Ознаки несправностей можуть бути явними та прихованими. Явні ознаки легко помітити: зупинка механізму, аварійне повідомлення, відсутність сигналу, перегрів, шум, дим, відмова запуску. Приховані ознаки складніше виявити без аналізу даних. Наприклад, невелике поступове зростання вібрації може ще не заважати роботі, але воно може свідчити про майбутню несправність підшипника. Саме приховані ознаки є важливим об’єктом для інтелектуальної діагностики.

За характером прояву несправності можна поділити на раптові та поступові. Раптова несправність виникає швидко, без помітного попереднього періоду. Поступова несправність розвивається протягом певного часу. Для інтелектуальних методів особливо важливими є поступові несправності,

оскільки вони можуть бути виявлені за змінами параметрів. У роботах, присвячених fault diagnosis і PHM, саме аналіз трендів, відхилень і закономірностей у даних розглядається як один з основних шляхів підвищення ефективності діагностики [14; 23; 28].

Окремо потрібно враховувати, що в складній технічній системі можуть виникати комбіновані несправності. Це ситуації, коли одночасно діє кілька причин. Наприклад, зношення механічного вузла може поєднуватися з нестабільним живленням і неправильним режимом експлуатації. У такому випадку звичайний пороговий контроль може бути недостатнім. Потрібен аналіз сукупності ознак.

На практиці діагностичні ознаки можна подати у вигляді таблиці даних. Кожен рядок такої таблиці описує певний момент або період роботи системи, а кожен стовпець відповідає окремому параметру. Останній стовпець може містити клас стану: «норма», «попередження», «критичний стан» або «несправність». Саме така структура даних зручна для використання методів машинного навчання [4; 21; 27].

Отже, діагностичні параметри є основою для оцінювання стану технічної системи. Вони дозволяють перейти від загального опису системи до формального аналізу її поведінки. Для складних технічних систем важливо враховувати багато параметрів, їхню зміну в часі та взаємозв’язки між ними. Це створює основу для побудови системної моделі діагностики та подальшого використання інтелектуальних алгоритмів.

1.4. Аналіз сучасних підходів до діагностики складних технічних систем

Сучасна технічна діагностика розвивається від простих перевірок і регламентного обслуговування до інтелектуальних систем моніторингу, які аналізують дані в реальному часі. Це пов’язано з тим, що сучасне обладнання стає складнішим, а вимоги до надійності та безпеки зростають. Якщо раніше

основним підходом було обслуговування після поломки або за графіком, то зараз дедалі частіше використовуються підходи, які дозволяють виявляти проблему до виникнення відмови.

Традиційний підхід до діагностики передбачає періодичний огляд обладнання, вимірювання окремих параметрів, перевірку відповідності нормативам і виконання ремонтних робіт за планом. Це є простим і зрозумілим, але має недоліки. Якщо перевірка проводиться рідко, несправність може виникнути між двома плановими оглядами. Якщо ж обслуговування проводиться занадто часто, це збільшує витрати та може бути неефективним.

Другий підхід – це діагностика за станом. Вона базується на регулярному або постійному контролі параметрів системи. Якщо параметри починають відхилятися від норми, система або фахівець отримує сигнал про можливу проблему. Це є більш гнучким, оскільки рішення про обслуговування приймається не тільки за календарем, а за фактичним станом обладнання. Саме з цим пов’язані стандарти ISO 13374-1:2003 та ISO 13379-2:2015, які описують роботу з даними моніторингу стану і діагностики [19; 20].

Ще один сучасний напрям – прогнозування технічного стану. Воно спрямоване не тільки на відповідь на питання «що відбувається зараз?», а й на питання «що може статися далі?». У межах PHM аналізуються поточні та історичні дані, щоб оцінити залишковий ресурс, ризик відмови або потребу в обслуговуванні [14; 26; 28]. Це особливо важливо для дорогого або критично важливого обладнання, де раптова зупинка є небажаною.

Сучасні підходи до діагностики можна умовно поділити на кілька груп: експертні, модельні, статистичні, data-driven, інтелектуальні та гібридні. Експертні підходи ґрунтуються на знаннях фахівців і правилах типу «якщо – то». Модельні підходи використовують фізичні або математичні моделі системи. Статистичні методи аналізують розподіли даних і відхилення від нормальних значень. Data-driven підходи спираються на дані, отримані від системи. Інтелектуальні методи використовують машинне навчання, нейронні мережі,

дерева рішень, ансамблеві моделі та інші алгоритми. Гібридні підходи поєднують кілька методів.

У статті H. Webert, T. Döß, M. F. Huber та J. Pauli розглядається fault handling в умовах Industry 4.0 [30]. Автори наголошують, що в сучасних виробничих системах важливо не лише виявляти несправності, а й правильно організовувати процес реагування на них. Це означає, що діагностика повинна бути пов’язана з прийняттям рішень: чи потрібно зупинити систему, провести ремонт, змінити режим роботи або продовжити спостереження.

У статті D. Leite та співавторів 2025 року про fault detection and diagnosis in Industry 4.0 зазначено, що сучасні системи діагностики стикаються з такими викликами, як великі обсяги даних, складність процесів, потреба в реальному часі, невизначеність і вимоги до пояснюваності результатів [23]. Це добре показує, чому інтелектуальні методи стають важливими. Вони можуть обробляти великі масиви даних, але при цьому виникає питання: чи можна пояснити, чому модель прийняла саме таке рішення.

Окрему роль відіграє автоматизоване машинне навчання. У статті D. Leite, A. Martins, D. Rativa, J. F. L. De Oliveira та A. M. A. Maciel розглянуто підхід AutoML для real-time fault detection and diagnosis [22]. Ідея такого підходу полягає в тому, що частина процесу вибору моделі, налаштування параметрів і оцінювання результатів може бути автоматизована. Це корисно тоді, коли потрібно швидко створити модель для діагностики на основі наявних даних.

У задачах технічної діагностики широко застосовуються методи машинного навчання. Б. І. Мокін, В. Б. Мокін та О. Б. Мокін розглядають машинне навчання як засіб інтелектуального аналізу даних, який дозволяє будувати моделі на основі прикладів [4]. Для діагностики це означає, що алгоритм може навчитися розпізнавати нормальні й несправні стани за попередньо підготовленими даними. Наприклад, якщо є набір записів із параметрами системи та відомими станами, можна навчити модель класифікувати нові стани.

Одним із класичних методів є Random Forest, запропонований L. Breiman [11]. Це ансамблевий метод, який складається з багатьох дерев рішень. Його перевага полягає в тому, що він часто дає кращу точність, ніж одне дерево, і може працювати з табличними даними. Для задач технічної діагностики Random Forest є зручним, тому що діагностичні параметри часто подані саме у вигляді таблиці.

Нейронні мережі та глибоке навчання також активно застосовуються для діагностики. C. M. Bishop та H. Bishop розглядають глибоке навчання як напрям, що дозволяє моделювати складні нелінійні залежності [10]. Це може бути корисним для аналізу сигналів, зображень, часових рядів або складних наборів даних. Наприклад, для діагностики підшипників можуть аналізуватися вібраційні сигнали, де важливі не тільки окремі значення, а й форма сигналу.

У статті X. Chen та співавторів подано систематичний огляд deep transfer learning для діагностики несправностей підшипників [13]. Transfer learning корисний тоді, коли даних для конкретної системи мало, але є дані з подібних систем. Це часта проблема в технічній діагностиці: несправностей може бути небагато, тому зібрати великий якісний набір даних складно. Перенесення знань з однієї задачі на іншу може частково вирішувати цю проблему.

У статті M. L. Hakim, A. Awad та S. I. Ihsan також розглянуто діагностику несправностей підшипників на основі deep learning і transfer learning [17]. Автори звертають увагу на те, що такі методи мають високу перспективність, але потребують якісних даних і правильного налаштування. Це важливо враховувати, бо інтелектуальний метод не гарантує хорошого результату сам по собі. Якість результату залежить від даних, ознак, моделі та критеріїв оцінювання.

Ще одним важливим напрямом є пояснюваний штучний інтелект. У статті J. Caçâo, J. S. Santos та M. Antunes розглянуто explainable AI для industrial fault diagnosis [12]. Пояснюваність є важливою тому, що в технічній діагностиці фахівцю недостатньо отримати відповідь «несправність». Йому потрібно розуміти, які параметри вплинули на рішення моделі. Якщо модель вказує, що

система перебуває в критичному стані, потрібно знати, чи це пов’язано з температурою, вібрацією, струмом або іншими ознаками.

Метод SHAP, запропонований S. M. Lundberg та S.-I. Lee, є одним із відомих підходів до інтерпретації рішень моделей машинного навчання [24].

У сучасних автоматизованих виробничих системах методи виявлення несправностей розглядаються як важлива частина підтримки стабільної роботи обладнання. Y. S. Ahmed, M. H. Habaebi та M. R. Islam зазначають, що розвиток fault detection techniques пов’язаний із переходом від простого контролю параметрів до автоматизованого аналізу даних, який дозволяє швидше виявляти відхилення в роботі технічної системи [9].

На рис. 1.4 показано класифікацію сучасних підходів до діагностики.

Рисунок 1.4 – Класифікація сучасних підходів до діагностики технічних систем

Найбільш перспективними є підходи, які поєднують моніторинг стану, машинне навчання, прогнозування відмов і пояснення результатів.

1.5. Порівняльний аналіз традиційних та інтелектуальних методів діагностики

Для вибору методу діагностики складної технічної системи потрібно враховувати не тільки точність, а й зрозумілість, складність реалізації, вимоги до даних, можливість роботи в реальному часі та придатність для конкретного типу

системи. Саме тому доцільно провести порівняльний аналіз традиційних та інтелектуальних методів.

Традиційні методи діагностики мають важливу перевагу: вони прості, зрозумілі та часто не потребують великих наборів даних. До них належать візуальний огляд, регламентні перевірки, пороговий контроль параметрів, аналіз відмов за FMEA/FMECA, експертні правила та статистичний контроль. Такі методи зручні для типових ситуацій, коли відомі допустимі межі параметрів і є досвід експлуатації системи.

Наприклад, пороговий контроль може працювати так: якщо температура перевищує певне значення, система формує попередження. Це просто й зрозуміло. Але недолік у тому, що реальний стан системи може залежати не від одного параметра, а від їх поєднання. Температура може бути трохи підвищеною, вібрація – теж трохи підвищеною, струм – близьким до межі. Окремо кожен параметр ще не є критичним, але разом вони можуть свідчити про проблему.

Методи FMEA та FMECA, визначені в IEC 60812:2018, дозволяють системно аналізувати можливі види відмов, їхні причини та наслідки [18]. Це сильний інструмент для попереднього аналізу ризиків. Проте він значною мірою залежить від експертних знань і не завжди може автоматично адаптуватися до нових даних, які надходять від системи.

Інтелектуальні методи мають інші переваги. Вони можуть працювати з великими наборами даних, аналізувати багато параметрів одночасно, знаходити приховані залежності та класифікувати стани системи. До таких методів належать дерева рішень, Random Forest, метод k-найближчих сусідів, метод опорних векторів, нейронні мережі, глибоке навчання, transfer learning, AutoML та інші підходи [4; 10; 11; 13; 16; 21; 22; 27].

У практичних задачах діагностики важливими є також методи прогнозування та PHM. Огляд Z. Fang, X. Wang, Y. Liu та L. Zhang показує, що next-generation artificial intelligence може використовуватися для оцінювання технічного стану та прогнозування відмов [14]. Це означає, що інтелектуальні

методи можуть бути корисними не тільки для поточного визначення стану, а й для попередження майбутніх проблем.

C. Wang та співавтори розглядають сучасні досягнення у mechanism/data-driven fault diagnosis для складних інженерних систем в умовах невизначеності [29]. Це є важливим, бо він показує, що не завжди потрібно вибирати тільки один шлях – фізичну модель або модель на основі даних. У складних системах доцільно поєднувати знання про механізм роботи об’єкта з аналізом реальних даних.

Порівняння основних методів діагностики подано в табл. 1.3.

Таблиця 1.3 – Порівняльний аналіз традиційних та інтелектуальних методів діагностики

Метод	Основна ідея	Переваги	Недоліки	Доцільність використання
1	2	3	4	5
Регламентна перевірка	Перевірка системи за встановленим графіком	проста організація, не потребує складних даних	може не виявити несправність між перевірками	для базового технічного обслуговування
Пороговий контроль	Порівняння параметрів із допустимими межами	простота, швидка реакція	не враховує складні залежності між параметрами	для контролю критичних параметрів
FMEA/FMECA	Аналіз видів відмов, причин і наслідків	системність, оцінка ризиків	залежність від експертів, складність оновлення	для попереднього аналізу ризиків
Статистичні методи	Аналіз розподілів, відхилень і закономірностей	добре працюють з числовими даними	потребують коректних припущень про дані	для аналізу стабільних процесів
Дерева рішень	Побудова правил класифікації станів	зрозуміла логіка, проста інтерпретація	можливе перенавчання	для задач класифікації стану
Random Forest	Ансамбль дерев рішень	вища точність, стійкість до шуму	складніша інтерпретація, ніж у одного дерева	для табличних діагностичних даних

Продовження табл.1.3

1	2	3	4	5
Нейронні мережі	Навчання складних нелінійних залежностей	висока точність на великих даних	потребують багато даних, складно пояснити рішення	для сигналів, зображень, часових рядів
Transfer Learning	Перенесення знань з однієї задачі на іншу	корисний при нестачі даних	складність вибору базової моделі	для подібних технічних систем
AutoML	Автоматизований вибір і налаштування моделей	прискорює побудову моделей	менший контроль над процесом	для швидкої побудови діагностичних моделей
Explainable AI	Пояснення рішень моделей	підвищує довіру до алгоритму	потребує додаткових методів інтерпретації	для критичних технічних систем

З табл. 1.3 видно, що традиційні методи мають переваги в простоті та зрозумілості, але не завжди можуть працювати зі складними залежностями. Інтелектуальні методи краще підходять для аналізу великих наборів даних і виявлення прихованих закономірностей, але вони потребують якісних даних, правильного навчання та перевірки результатів.

Для складних технічних систем найбільш доцільним є не повна заміна традиційних методів інтелектуальними, а їх поєднання. Наприклад, FMEA/FMECA може використовуватися для визначення можливих відмов і критичних параметрів, а машинне навчання – для класифікації поточного стану системи за даними датчиків. Пороговий контроль може залишатися як простий рівень захисту, а інтелектуальна модель – як додатковий інструмент аналізу.

У практичних системах діагностики важливо враховувати якість даних. Якщо дані містять багато помилок, пропусків або не відображають реальні несправності, навіть складний алгоритм може дати неправильний результат. Тому перед використанням машинного навчання необхідно виконувати попередню обробку даних: очищення, нормалізацію, поділ на навчальну й тестову вибірки, вибір ознак і перевірку балансу класів [16; 21; 27].

Також важливо правильно обирати критерії оцінювання. Для діагностики недостатньо оцінювати тільки загальну точність. Наприклад, якщо модель добре розпізнає нормальний стан, але часто пропускає несправності, така модель є небезпечною для практичного використання. Тому потрібно враховувати precision, recall, F1-score і матрицю помилок. Особливо важливим є recall для класів, пов’язаних із критичним станом або відмовою [16; 21].

Окреме значення має пояснюваність. У задачах технічної діагностики оператор або інженер має розуміти, чому система зробила певний висновок. Якщо модель визначила критичний стан, потрібно бачити, які параметри найбільше вплинули на це рішення. Це може бути температура, вібрація, струм, кількість помилок або їх поєднання. Тому Explainable AI є важливим доповненням до інтелектуальної діагностики [12; 24].

На рис. 1.5 подано узагальнену схему поєднання традиційних та інтелектуальних методів.

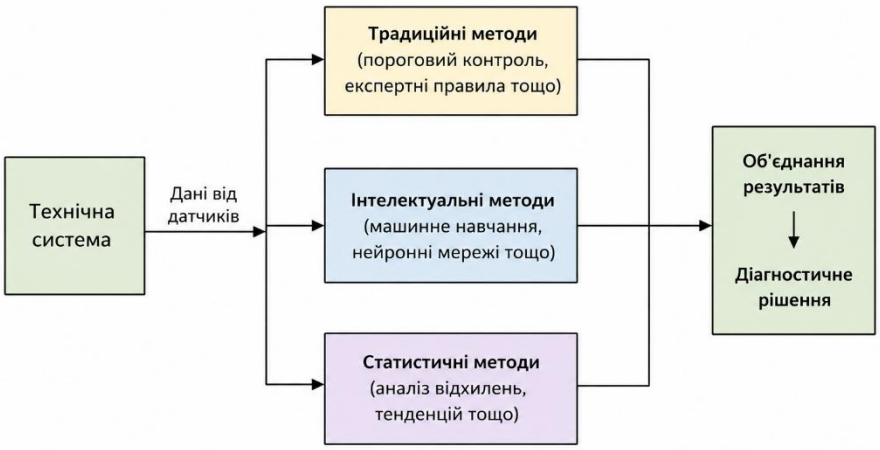


Рисунок 1.5 – Поєднання традиційних та інтелектуальних методів діагностики

Загалом порівняльний аналіз показує, що інтелектуальні методи мають значний потенціал для діагностики складних технічних систем. Вони дозволяють працювати з великою кількістю параметрів, виявляти приховані залежності, класифікувати стани та підтримувати прийняття рішень. Разом з тим їх

використання потребує системного підходу: потрібно правильно визначити об’єкт, параметри, стани, критерії якості та спосіб перевірки результатів.

Висновки до розділу 1

Розглянуто складні технічні системи як об’єкти діагностики з позиції системного аналізу. Визначено, що така система складається з багатьох взаємопов’язаних елементів, має вхідні та вихідні параметри, працює в різних режимах і може перебувати в кількох технічних станах.

Показано, що складні технічні системи функціонують в умовах невизначеності. Невизначеність може бути пов’язана з неповнотою даних, похибками вимірювання, зміною навантаження, зношенням елементів або складністю зв’язків між параметрами. Через це для діагностики недостатньо аналізувати один окремий показник. Потрібно враховувати сукупність діагностичних ознак і можливі переходи системи між нормальним, попереджувальним, критичним і відмовним станами.

Розглянуто основні групи діагностичних параметрів: теплові, механічні, електричні, інформаційні, експлуатаційні та комбіновані. Встановлено, що саме ці параметри можуть бути використані для формування набору ознак у задачі інтелектуальної діагностики.

Проаналізовано сучасні підходи до діагностики складних технічних систем. До традиційних підходів належать регламентна перевірка, пороговий контроль, експертні правила, FMEA/FMECA та статистичні методи. До інтелектуальних підходів належать машинне навчання, Random Forest, нейронні мережі, transfer learning, AutoML, PHM та Explainable AI. Визначено, що традиційні методи є простими й зрозумілими, але мають обмеження при роботі зі складними залежностями між параметрами. Інтелектуальні методи краще підходять для аналізу великих наборів даних, однак потребують якісних даних і правильної оцінки результатів. Найбільш доцільним є поєднання системного аналізу, традиційних методів діагностики та інтелектуальних алгоритмів.

РОЗДІЛ 2. РОЗРОБКА СИСТЕМНОЇ МОДЕЛІ ТА АЛГОРИТМУ ІНТЕЛЕКТУАЛЬНОЇ ДІАГНОСТИКИ

2.1. Постановка задачі діагностики технічного стану складної системи

Діагностика складної технічної системи передбачає визначення її поточного стану за набором параметрів, які отримуються під час роботи системи. Такими параметрами можуть бути температура, вібрація, напруга, струм, навантаження, швидкість роботи, кількість помилок, час відгуку, інтенсивність зношення та інші показники. Основна складність полягає в тому, що ці параметри не завжди прямо вказують на конкретну несправність. Один і той самий симптом може мати різні причини, а несправність може проявлятися не одразу, а поступово.

З позиції системного аналізу задача діагностики не зводиться тільки до перевірки окремого параметра. Потрібно розглядати систему як цілісний об’єкт, у якому елементи взаємодіють між собою, а зміна одного параметра може впливати на інші. Наприклад, підвищення температури може бути пов’язане з перевантаженням, порушенням охолодження, зношенням вузла або неправильною роботою системи керування. Тому діагностичне рішення має формуватися на основі сукупності ознак, а не тільки за одним показником [7; 8].

У загальному вигляді складну технічну систему можна подати як об’єкт, що має вхідні параметри, внутрішній стан, зовнішні впливи та вихідні результати. Вхідними параметрами є дані, що надходять від датчиків або інформаційних каналів. Внутрішній стан характеризує працездатність системи. Зовнішні впливи можуть бути пов’язані з умовами експлуатації, навантаженням, температурою середовища, вологістю або діями оператора. Вихідним результатом діагностики є визначений клас технічного стану: нормальний стан, попередження, критичний стан або несправність.

Математично задачу діагностики можна подати як задачу класифікації. Нехай технічна система описується набором параметрів:

$$X = \{x_1, x_2, x_3, \ldots, x_n\},$$

де x_1, x_2, x_3, ..., x_n – діагностичні параметри системи.

Тоді потрібно побудувати таку функцію:

$$F(X) \rightarrow Y,$$

де Y – клас технічного стану системи. Наприклад:

Y_1 – нормальний стан;

Y_2 – допустиме відхилення;

Y_3 – попереджувальний стан;

Y_4 – критичний стан;

Y_5 – несправність.

Це відповідає логіці машинного навчання, де на основі вхідних ознак будується модель, здатна визначати належність об’єкта до певного класу [4; 21; 25]. Для технічної діагностики це означає, що модель повинна навчитися розрізняти стани системи за даними, які характеризують її роботу.

На рис. 2.1 подано загальну постановку задачі діагностики технічного стану складної системи.

Рисунок 2.1 – Загальна схема постановки задачі діагностики технічного стану

На рис. 2.1 зображено, що від складної технічної системи надходять дані від датчиків і журналів роботи. Ці дані проходять попередню обробку, після чого

подаються на вхід діагностичної моделі. Результатом роботи моделі є визначення технічного стану системи та формування діагностичного рішення.

У контексті роботи доцільно розглядати не конкретне обладнання одного типу, а узагальнену складну технічну систему. Це дозволяє сформувати модель, яку можна адаптувати до різних технічних об’єктів. Наприклад, одна й та сама логіка діагностики може бути використана для виробничого механізму, енергетичного вузла, системи зв’язку або автоматизованої лінії. Відрізнятимуться тільки конкретні параметри, але загальна структура діагностики залишатиметься подібною.

Для побудови системної моделі необхідно визначити такі складові:

– об’єкт діагностики;
– вхідні параметри;
– діагностичні ознаки;
– можливі стани системи;
– алгоритм обробки даних;
– метод визначення стану;
– критерії оцінювання якості діагностики.

У межах системного аналізу важливо не лише обрати алгоритм, а й правильно описати процес прийняття рішення. Сам алгоритм машинного навчання є тільки одним із блоків системи. Перед ним мають бути етапи збору даних, перевірки якості даних, нормалізації, вибору ознак і поділу даних на навчальну та тестову вибірки. Після роботи алгоритму потрібна оцінка результатів і формування висновку про стан системи [16; 27].

На рис. 2.2 подана UML-діаграму варіантів використання системи інтелектуальної діагностики.

На рис. 2.2 зображено два основних актори: «Оператор» і «Інженер з обслуговування». Основні варіанти використання: «Завантажити діагностичні дані», «Переглянути параметри системи», «Запустити діагностику», «Отримати клас стану», «Переглянути рекомендацію», «Оцінити результати моделі».

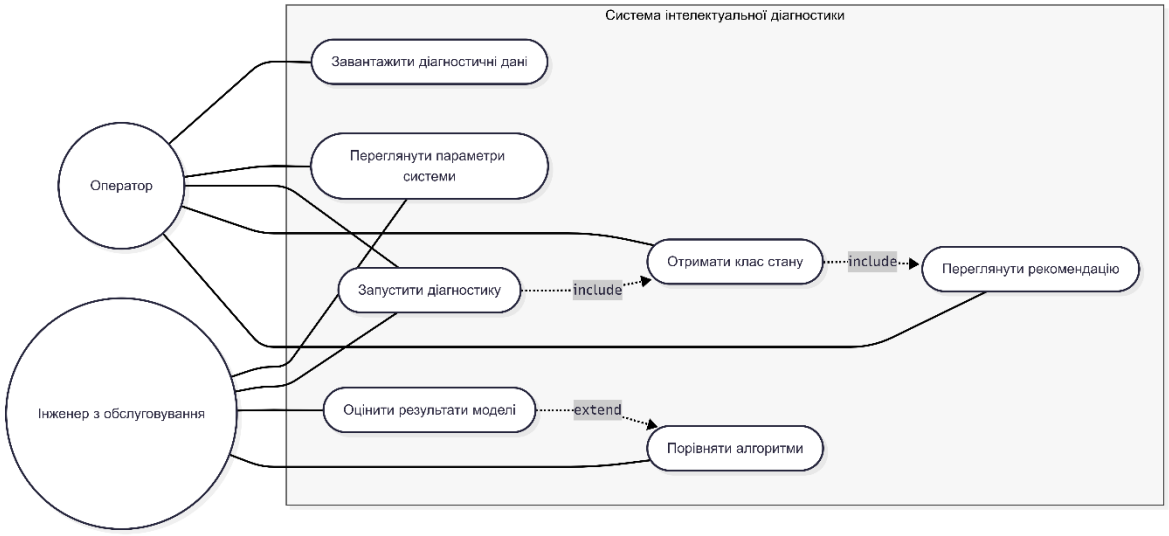


Рисунок 2.2 – UML-діаграма варіантів використання системи діагностики

У такій моделі оператор працює з поточними даними й отримує результат діагностики. Інженер може аналізувати результати глибше, переглядати помилки моделі, змінювати набір ознак або порівнювати різні алгоритми. Це відповідає практиці використання діагностичних систем, де результат алгоритму не повинен повністю замінювати фахівця, а має допомагати йому приймати обґрунтоване рішення.

Постановка задачі діагностики також передбачає визначення обмежень. По-перше, модель залежить від якості вхідних даних. Якщо дані неповні, шумні або неправильно позначені, результат діагностики може бути неточним. По-друге, модель повинна враховувати різні стани системи, а не тільки норму й відмову. По-третє, для практичного використання важливо оцінювати не лише загальну точність, а й здатність моделі виявляти критичні стани [16; 21].

Отже, задача діагностики технічного стану складної системи полягає у визначенні класу її стану на основі множини вхідних параметрів. Для її розв’язання потрібно побудувати системну модель, сформувати набір діагностичних ознак, обрати інтелектуальні методи, розробити алгоритм обробки даних і визначити критерії оцінювання ефективності.

2.2. Формування множини вхідних параметрів і діагностичних ознак

Формування множини вхідних параметрів є одним із головних етапів побудови системи діагностики. Якщо параметри підібрані неправильно, навіть складний алгоритм не зможе дати якісний результат. У технічних системах параметри можуть мати різну природу: механічну, електричну, теплову, інформаційну або експлуатаційну. Тому перед побудовою алгоритму потрібно визначити, які саме ознаки реально характеризують стан системи.

Вхідний параметр – це показник, який можна отримати безпосередньо від системи або розрахувати на основі інших даних. Наприклад, температура вузла може бути виміряна датчиком, а швидкість зміни температури вже є розрахунковою ознакою. Подібно до цього, струм двигуна може бути виміряний напряму, а коефіцієнт перевантаження може бути отриманий шляхом обчислення.

Діагностична ознака – це параметр або його зміна, яка має значення для визначення технічного стану. Наприклад, сама вібрація є параметром, але її перевищення допустимої межі або стійке зростання в часі є діагностичною ознакою. У задачах машинного навчання такі ознаки утворюють вектор вхідних даних, на основі якого модель виконує класифікацію [4; 21].

У роботах, присвячених технічній діагностиці та condition monitoring, велика увага приділяється саме даним, які надходять від технічного об’єкта [19; 20]. Це пояснюється тим, що сучасна діагностика поступово переходить від ручної перевірки до аналізу даних. Якщо раніше основним джерелом інформації був огляд фахівця, то сьогодні значну роль відіграють датчики, журнали роботи, контролери й автоматизовані системи моніторингу.

Для узагальненої складної технічної системи можна сформувати Цей набір вхідних параметрів:

$$X = \{T, V, U, I, L, E, R, H\},$$

де:

T – температура;

V – рівень вібрації;

U – напруга;

I – струм;

L – рівень навантаження;

E – кількість помилок або аварійних повідомлень;

R – час реакції або затримка;

H – напрацювання або кількість годин роботи.

Цей набір є узагальненим. Для конкретної технічної системи він може змінюватися. Наприклад, для механічного вузла важливішими будуть температура, вібрація, швидкість обертання та напрацювання. Для інформаційно-комунікаційної системи важливими будуть затримка, кількість помилок, пропускна здатність і стабільність сигналу [6].

У табл. 2.1 подано приклад множини вхідних параметрів для інтелектуальної діагностики.

Таблиця 2.1 – Множина вхідних параметрів для діагностики технічної системи

Позначення	Назва параметра	Тип параметра	Можливе значення	Діагностичне значення
1	2	3	4	5
T	Температура	тепловий	°C	виявлення перегріву або перевантаження
V	Вібрація	механічний	мм/с або умовні одиниці	виявлення зношення, дисбалансу, дефектів вузлів
U	Напруга	електричний	В	оцінка стабільності живлення
I	Струм	електричний	А	виявлення перевантаження або короткого замикання

Продовження табл.2.1

1	2	3	4	5
L	Навантаження	експлуатаційний	%	оцінка режиму роботи системи
E	Кількість помилок	інформаційний	кількість подій	виявлення збоїв, відмов або нестабільної роботи
R	Час реакції	інформаційний	мс	оцінка затримки та продуктивності
H	Напрацювання	експлуатаційний	години	оцінка ресурсу та зношення

З табл. 2.1 видно, що параметри мають різні одиниці вимірювання. Через це перед поданням даних на вхід алгоритму потрібно виконувати попередню обробку. Наприклад, параметри можуть бути нормалізовані, щоб значення з різними масштабами не спотворювали роботу моделі. Це особливо важливо для методів, які залежать від відстані між об’єктами, наприклад k-найближчих сусідів [16; 21].

Крім основних параметрів, доцільно формувати додаткові діагностичні ознаки. Вони можуть краще відображати зміну стану системи. Наприклад:

- швидкість зміни температури;
- середнє значення вібрації за певний період;
- максимальне навантаження за останній інтервал часу;
- кількість аварійних повідомлень за зміну;
- співвідношення струму до навантаження;
- інтегральний індекс технічного стану.

Ці ознаки дозволяють врахувати не лише поточне значення параметра, а й поведінку системи в часі. Це важливо, тому що багато несправностей розвиваються поступово. Наприклад, якщо температура зростає повільно, але постійно, то в певний момент система може перейти в критичний стан. Звичайна перевірка одного значення може цього не показати, але аналіз динаміки дозволяє виявити проблему раніше.

На рис. 2.3 подано схему формування діагностичних ознак.

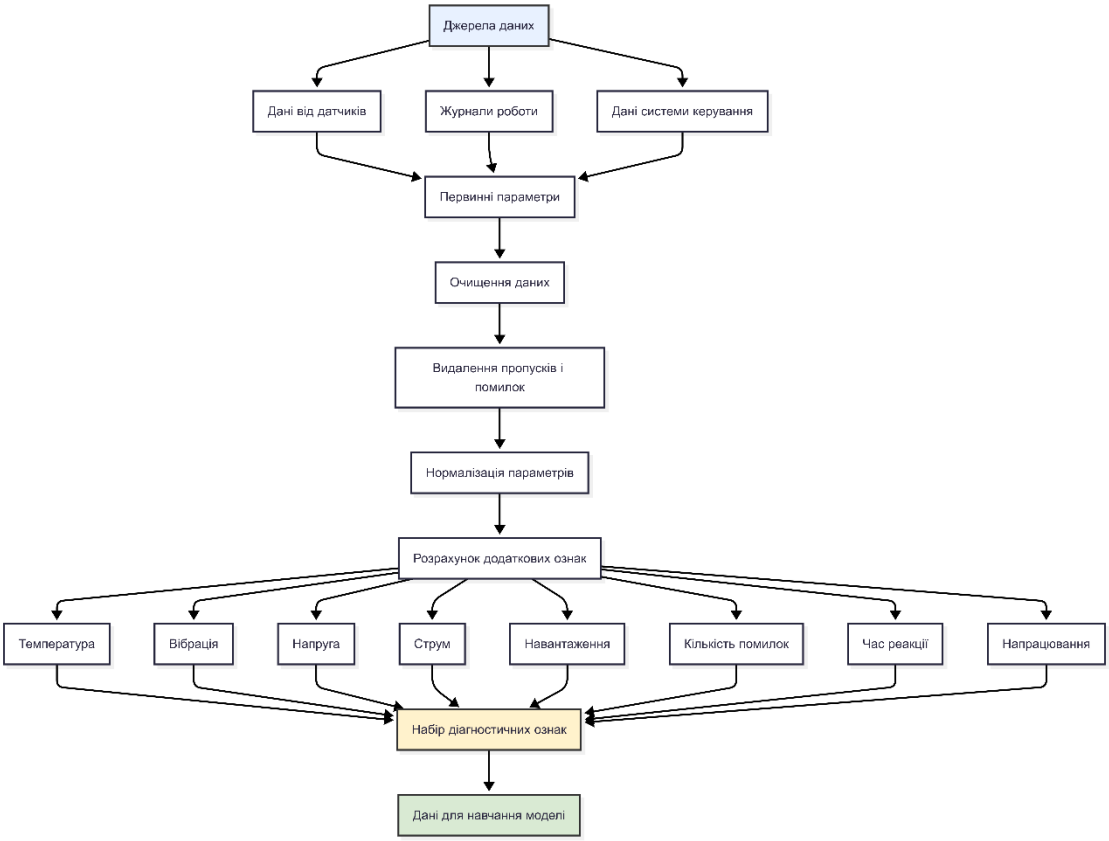


Рисунок 2.3 – Схема формування діагностичних ознак технічної системи

На рис. 2.3 показано, що первинні дані надходять від датчиків, журналів роботи та системи керування. Після цього виконується очищення, нормалізація та розрахунок додаткових ознак. У результаті формується набір даних для навчання та тестування діагностичної моделі.

Для побудови набору даних необхідно також визначити цільову змінну. У цій роботі цільовою змінною є клас технічного стану системи. Наприклад, можна використати такі класи:

0 – нормальний стан;

1 – допустиме відхилення;

2 – попереджувальний стан;

3 – критичний стан;

4 – несправність.

Поділ на п’ять класів є більш інформативним, ніж простий поділ на два класи. Якщо використовувати тільки класи «норма» і «несправність», алгоритм не зможе показати поступове погіршення стану. Наявність проміжних класів дозволяє виявляти ризик раніше та приймати рішення про профілактичне обслуговування.

На рис. 2.4 зображено UML-діаграму класів структури діагностичних даних.

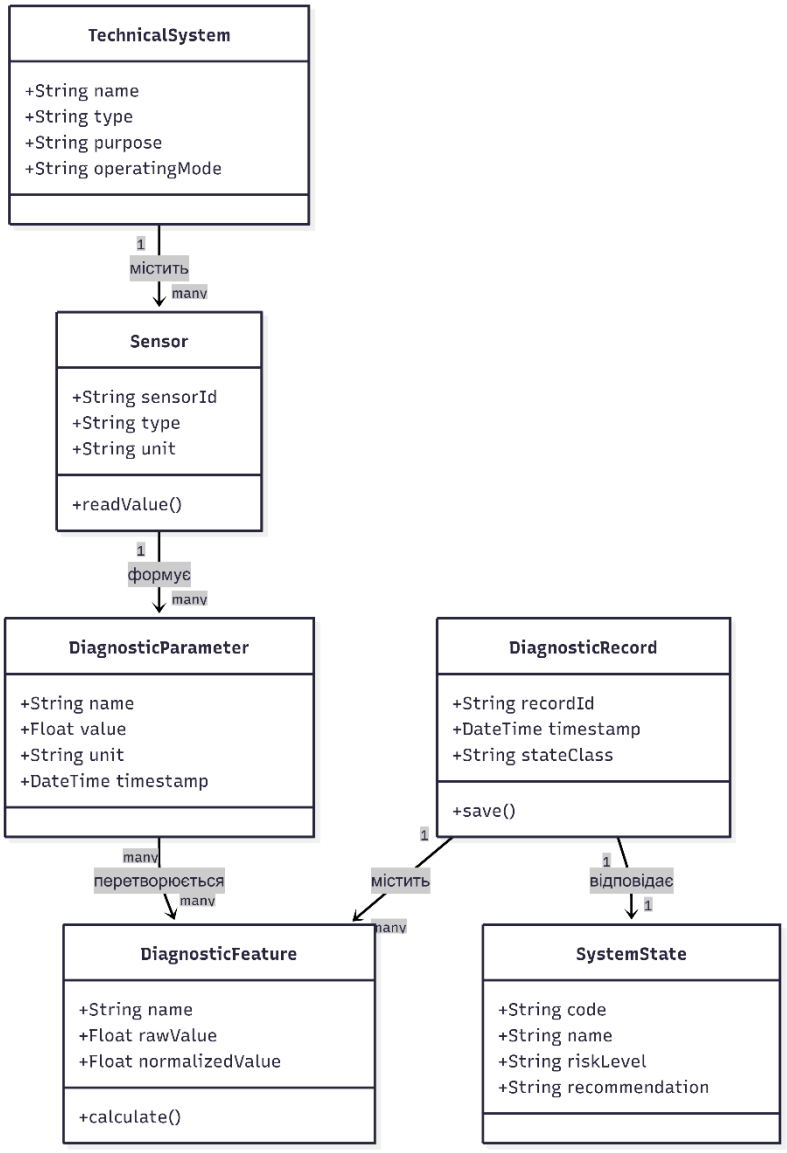


Рисунок 2.4 – UML-діаграма класів структури діагностичних даних

Структуру класів можна описати так:

1. Технічна система: назва, тип, призначення, режим роботи;

2. Датчик: ідентифікатор, тип, одиниця вимірювання;
3. Параметр: назва, значення, час вимірювання;
4. Ознака: назва, розраховане значення, нормалізоване значення;
5. Стан системи: код стану, назва, рівень ризику;
6. Діагностичний запис: набір ознак, клас стану, час фіксації.

Ця структура є зручною для подальшої програмної реалізації. Вона дозволяє зберігати дані у вигляді таблиці або бази даних. Кожен запис може відповідати певному моменту часу, а кожен стовпець – окремій ознаці. Саме так зазвичай організовуються дані для задач машинного навчання у Python та scikit-learn [16; 27].

Під час формування множини ознак потрібно враховувати надмірність. Якщо в наборі є багато параметрів, які дублюють одне одного або не впливають на результат, модель може працювати гірше. Тому важливо виконувати відбір ознак. Наприклад, якщо два параметри майже повністю повторюють один одного, достатньо залишити один із них. Якщо певний параметр не має зв’язку зі станом системи, його можна не використовувати.

Отже, формування вхідних параметрів і діагностичних ознак є основою для побудови інтелектуальної діагностики. На цьому етапі визначається, які дані характеризують технічну систему, як вони будуть оброблятися та в якому вигляді подаватимуться на вхід алгоритму. Якість цього етапу напряму впливає на точність і практичну корисність моделі.

2.3. Побудова моделі станів технічної системи

Модель станів потрібна для того, щоб формально описати можливі режими роботи технічної системи. Без такої моделі алгоритм діагностики не матиме чіткої основи для класифікації. Якщо не визначити, які саме стани можуть бути в системи, то неможливо коректно навчати модель і перевіряти її результати.

Стан технічної системи можна розуміти як сукупну характеристику її працездатності в певний момент часу. У простому випадку система може бути

працездатною або непрацездатною. Проте для складних систем такого поділу недостатньо. Важливо розрізняти нормальну роботу, незначні відхилення, передаварійний стан і фактичну несправність [1; 2].

У межах цієї роботи доцільно використати п’ять основних станів:

S_0 – нормальний стан.

Система працює стабільно, параметри перебувають у допустимих межах.

S_1 – допустиме відхилення.

Є невеликі відхилення окремих параметрів, але система ще виконує свої функції без значного ризику.

S_2 – попереджувальний стан.

Спостерігається стійка зміна одного або кількох параметрів, що може свідчити про початок розвитку несправності.

S_3 – критичний стан.

Параметри наближаються до небезпечних меж або вже частково їх перевищують. Потрібне втручання оператора або інженера.

S_4 – несправність / відмова.

Система повністю або частково втратила здатність виконувати задані функції.

На рис. 2.5 подано модель переходів між станами технічної системи.

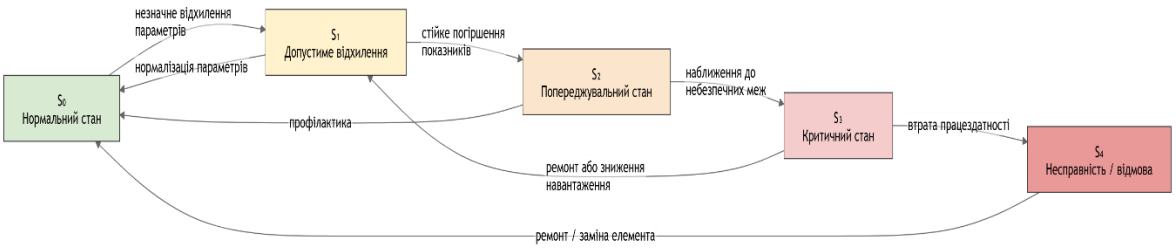


Рисунок 2.5 – Модель переходів технічної системи між станами

На рис. 2.5 зображено послідовний перехід від нормального стану до відмови: $S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_4$. Також показано можливість повернення системи до нормального стану після ремонту або профілактичного обслуговування.

У реальній системі перехід між станами не завжди є послідовним. Наприклад, через різкий збій система може перейти з нормального стану одразу в критичний або відмовний. Однак для більшості поступових несправностей модель із послідовним погіршенням є зручною. Вона дозволяє описати розвиток дефекту та сформувати логіку раннього попередження.

Для кожного стану потрібно визначити орієнтовні діагностичні ознаки. Це можна зробити через порогові значення або через навчальні дані. У першому випадку фахівець задає межі параметрів. У другому випадку модель сама навчається на прикладах, у яких уже відомо, який набір параметрів відповідає певному стану [4; 21].

У табл. 2.2 подано опис станів технічної системи.

Таблиця 2.2 – Модель станів технічної системи

Код стану	Назва стану	Характеристика	Рівень ризику	Рекомендована дія
S_0	Нормальний стан	параметри в межах норми	низький	продовжити роботу
S_1	Допустиме відхилення	незначні відхилення окремих параметрів	помірний	посилити спостереження
S_2	Попереджувальний стан	стійка зміна параметрів, можливий розвиток дефекту	підвищений	провести додаткову перевірку
S_3	Критичний стан	параметри близькі до небезпечних меж	високий	зменшити навантаження або зупинити систему
S_4	Несправність	система частково або повністю не виконує функції	дуже високий	ремонт або заміна елемента

З табл. 2.2 видно, що кожен стан має не тільки опис, а й рівень ризику та рекомендовану дію. Це важливо, тому що діагностика повинна завершуватися не лише визначенням класу, а й практичним висновком. Якщо система перебуває у стані S_2, ще можна обмежитися додатковим контролем. Якщо система переходить у S_3, потрібна швидка реакція.

Модель станів також можна подати у вигляді UML-діаграми станів (рис.2.6).

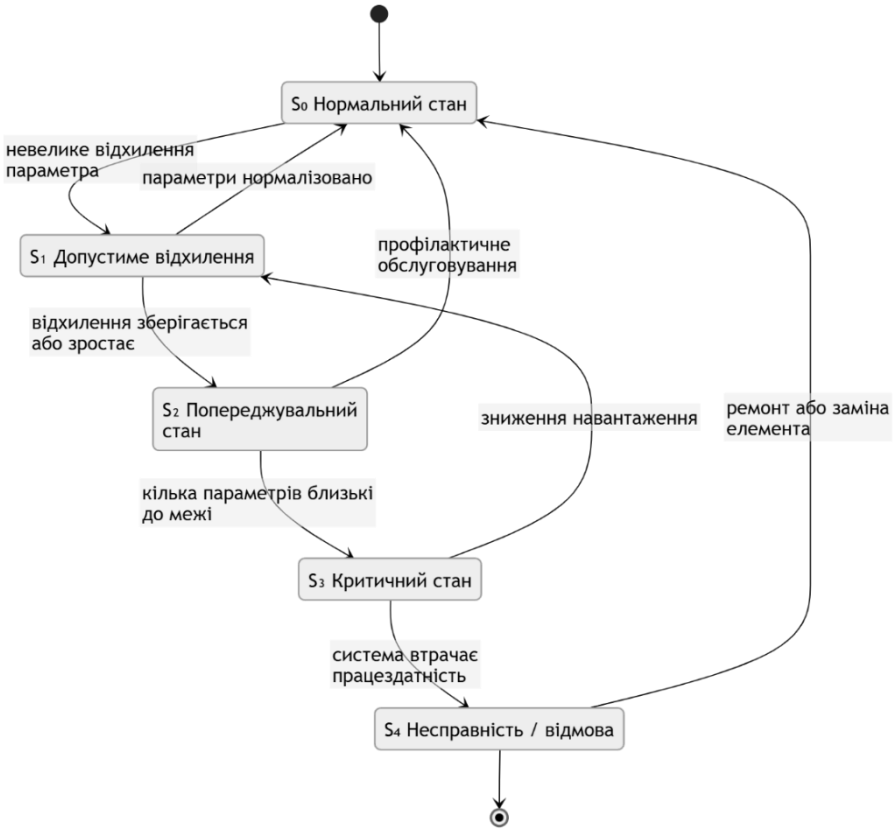


Рисунок 2.6 – UML-діаграма станів технічної системи

Приклад логіки переходів можна описати так:

– якщо параметри стабільні, система залишається у стані S_0;

– якщо один параметр трохи відхилився від норми, система переходить у S_1;

– якщо відхилення зберігається або зростає, система переходить у S_2;

– якщо кілька параметрів одночасно наближаються до критичних меж, система переходить у S_3;

– якщо система не виконує функцію або параметри перевищили допустимі межі, фіксується S_4;

– після ремонту або профілактики система може повернутися до S_0.

З позиції системного аналізу така модель станів є основою для формалізації діагностичного процесу. Вона дозволяє пов’язати технічні параметри, ризики та рішення. Крім того, вона допомагає підготувати навчальний набір даних. Якщо для кожного запису відомий стан системи, можна навчати модель класифікації.

У задачах РНМ важливим є не лише визначення поточного стану, а й прогнозування переходу до гіршого стану [14; 26; 28]. Наприклад, якщо система перебуває у стані S_2, алгоритм може оцінювати, чи є ризик переходу до S_3. Це дозволяє перейти від реактивного обслуговування до попереджувального.

Отже, модель станів технічної системи дозволяє формально описати процес погіршення або відновлення стану системи. Вона створює основу для побудови алгоритму діагностики, визначення класів у наборі даних і формування рекомендацій для оператора.

2.4. Вибір інтелектуальних методів для розв’язання задачі діагностики

Після визначення параметрів і станів системи потрібно обрати методи, які будуть використовуватися для діагностики. У цій роботі задача діагностики розглядається як задача класифікації. Це означає, що за набором вхідних ознак потрібно визначити клас технічного стану системи.

Для такої задачі доцільно порівняти кілька методів машинного навчання. Вибір одного алгоритму без порівняння не є достатньо обґрунтованим, тому що різні методи можуть по-різному працювати з одним і тим самим набором даних. Одні моделі простіші й краще пояснюються, інші можуть давати вищу точність, але бути складнішими для інтерпретації [4; 16; 21].

У роботі обрано такі методи:

- дерево рішень;
- Random Forest;
- метод k-найближчих сусідів;
- логістична регресія;

– нейронна мережа як перспективний метод для складніших наборів даних.

Дерево рішень є одним із найзрозуміліших методів класифікації. Його логіка нагадує послідовність правил. Наприклад, якщо температура вища за певне значення, а вібрація також зростає, система може бути віднесена до попереджувального або критичного стану. Перевага дерева рішень полягає в простій інтерпретації. Недолік – можливість перенавчання, коли модель дуже добре працює на навчальних даних, але гірше на нових.

Random Forest є ансамблевим методом, який складається з багатьох дерев рішень. L. Breiman показав, що поєднання багатьох дерев дозволяє підвищити стійкість і точність класифікації [11]. Для технічної діагностики цей метод є зручним, тому що він добре працює з табличними даними, може враховувати нелінійні зв’язки та менш схильний до перенавчання, ніж одне дерево.

Метод k-найближчих сусідів класифікує новий об’єкт за схожістю з уже відомими прикладами. Якщо параметри нового запису схожі на записи, які відповідають критичному стану, модель також віднесе його до критичного стану. Цей метод простий, але чутливий до масштабу ознак, тому перед його використанням потрібно нормалізувати дані [16; 21].

Логістична регресія є базовим статистичним методом класифікації. Вона добре підходить як проста контрольна модель. Її перевага полягає в тому, що результат можна пояснювати через вплив окремих ознак. Але якщо залежності між параметрами складні й нелінійні, логістична регресія може бути менш точною.

Нейронні мережі можуть моделювати складні нелінійні залежності [10]. Вони особливо корисні для сигналів, часових рядів або великих наборів даних. Проте для невеликого табличного набору даних нейронна мережа не завжди є найкращим вибором. Вона потребує більше даних, налаштування та контролю перенавчання. Тому в межах базової практичної реалізації доцільно використати простіші й більш пояснювані методи, а нейронні мережі розглядати як перспективний напрям.

На рис. 2.7 подано схему вибору інтелектуальних методів для діагностики.

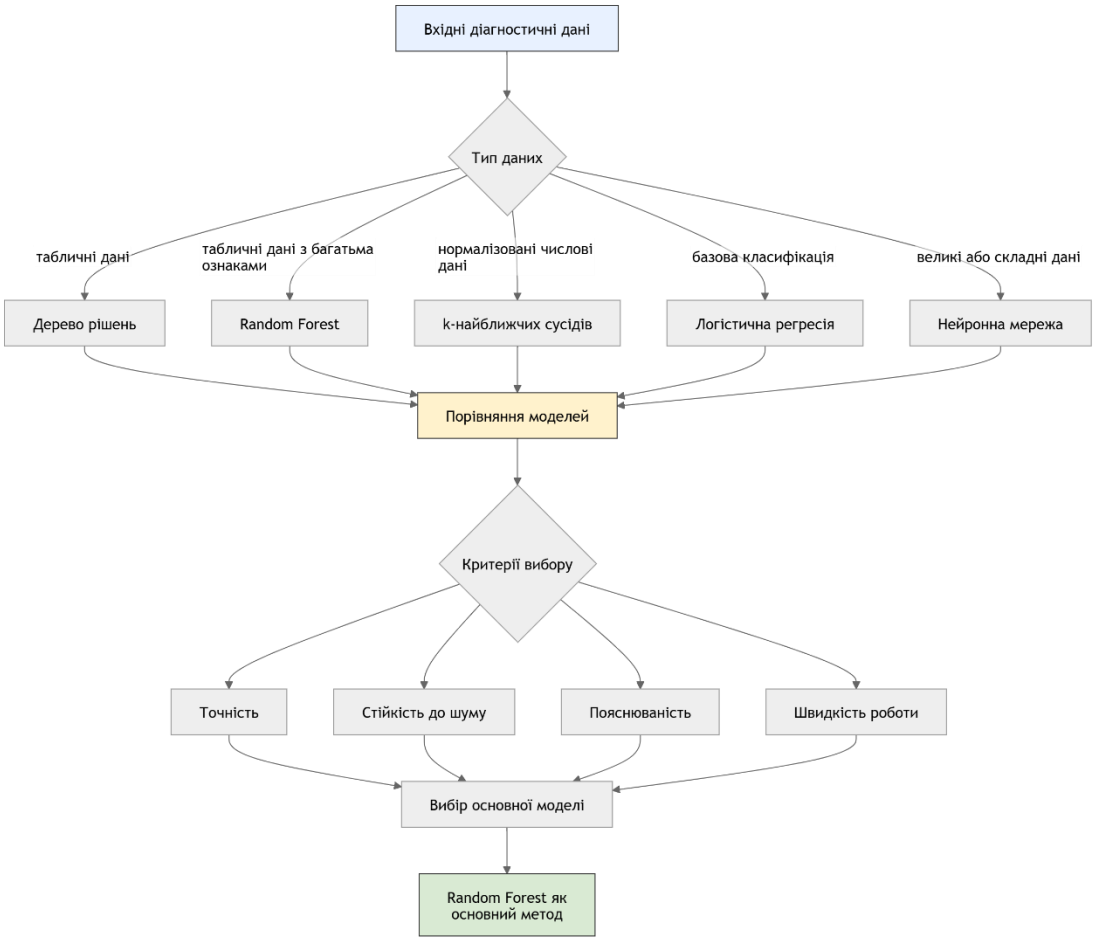


Рисунок 2.7 – Схема вибору інтелектуальних методів діагностики

На рис. 2.7 показано, що вибір методу залежить від типу даних, кількості записів, вимог до точності, потреби в пояснюваності та складності реалізації. Для табличних діагностичних даних доцільно порівнювати дерево рішень, Random Forest, kNN і логістичну регресію.

Для основної реалізації найбільш доцільним є Random Forest. Він поєднує достатню точність, роботу з табличними даними та стійкість до випадкових коливань у даних. Дерево рішень можна використати як більш просту модель для пояснення логіки класифікації, а kNN і логістичну регресію – для порівняння.

Важливим питанням є пояснюваність результатів. У технічній діагностиці модель повинна не тільки визначити стан, а й показати, які параметри вплинули на рішення. Для цього можна використовувати важливість ознак у Random Forest

або методи explainable AI. У роботі J. Cação, J. S. Santos та M. Antunes підкреслюється, що пояснюваність є важливою для промислової діагностики, бо без пояснення результату фахівцю складно довіряти моделі [12]. Метод SHAP, запропонований S. M. Lundberg та S.-I. Lee, дозволяє оцінювати внесок окремих ознак у прогноз моделі [24].

Також важливо врахувати програмну реалізацію. Для побудови моделей доцільно використовувати Python і бібліотеку scikit-learn. У статті F. Pedregosa та співавторів scikit-learn описано як інструмент машинного навчання для Python, що містить багато алгоритмів класифікації, регресії, кластеризації та оцінювання моделей [27]. A. Géron також детально розглядає практичне використання scikit-learn для задач машинного навчання [16].

Отже, для розв’язання задачі діагностики доцільно обрати кілька інтелектуальних методів і порівняти їх за однаковими критеріями. Основною моделлю може бути Random Forest, оскільки він добре підходить для табличних діагностичних даних. Додатково доцільно використати дерево рішень, kNN і логістичну регресію для порівняльного аналізу.

2.5. Розробка алгоритму інтелектуальної діагностики технічної системи

Алгоритм інтелектуальної діагностики повинен описувати послідовність дій від отримання даних до формування діагностичного рішення. Його структура має бути зрозумілою, тому що в технічних системах важливо не тільки отримати результат, а й розуміти, як саме він був отриманий.

Розроблений алгоритм складається з таких основних етапів:

1. Збір діагностичних даних.
2. Перевірка та очищення даних.
3. Формування діагностичних ознак.
4. Нормалізація або масштабування параметрів.
5. Поділ даних на навчальну та тестову вибірки.

6. Навчання моделей.
7. Отримання прогнозу стану.
8. Оцінювання якості моделей.
9. Вибір найкращої моделі.
10. Формування діагностичного рішення.

На рис. 2.8 подано блок-схему алгоритму інтелектуальної діагностики.

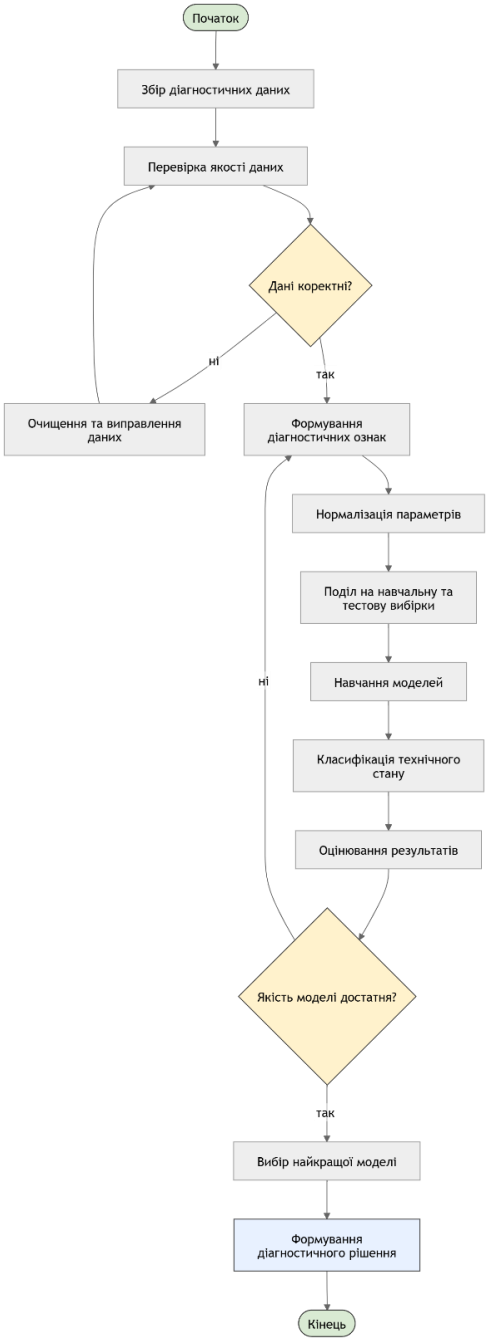


Рисунок 2.8 – Блок-схема алгоритму інтелектуальної діагностики технічної системи

Першим етапом є збір даних. Дані можуть надходити від датчиків, систем керування, журналів роботи або файлів із результатами вимірювань. У практичній реалізації дані можуть бути подані у вигляді CSV-файлу. Цей формат простий, зручний для обробки й підтримується більшістю інструментів аналізу даних.

Другим етапом є перевірка даних. Потрібно виявити пропущені значення, помилки вимірювання, аномальні записи або дублікати. Якщо дані містять пропуски, їх можна видалити або заповнити середніми значеннями. Якщо є очевидно неправильні записи, їх потрібно виключити. Цей етап важливий, бо якість даних сильно впливає на якість діагностики [16; 21].

Третім етапом є формування ознак. На цьому етапі з первинних параметрів можуть обчислюватися додаткові ознаки. Наприклад, можна розрахувати середнє значення температури, зміну вібрації за певний час або співвідношення струму до навантаження. Такі ознаки можуть підвищити точність моделі, бо вони краще описують поведінку системи.

Четвертим етапом є нормалізація або масштабування даних. Це потрібно не для всіх моделей, але для деяких алгоритмів є важливим. Наприклад, kNN використовує відстань між об’єктами, тому параметри з великими числовими значеннями можуть мати надмірний вплив. Для Random Forest масштабування не є критичним, але загальна підготовка даних все одно покращує порядок роботи з набором.

П’ятий етап – поділ даних на навчальну і тестову вибірки. Навчальна вибірка використовується для побудови моделі, а тестова – для перевірки її роботи на нових даних. Найчастіше використовують поділ 70/30 або 80/20. Це дозволяє оцінити, чи модель здатна працювати не тільки з уже відомими прикладами, а й з новими ситуаціями [16; 27].

Шостий етап – навчання моделей. На цьому етапі обрані алгоритми навчаються на підготовлених даних. Наприклад, дерево рішень формує правила, Random Forest будує ансамбль дерев, логістична регресія визначає ваги ознак, а kNN запам’ятовує розташування об’єктів у просторі ознак.

Сьомий етап – отримання прогнозу. Модель отримує на вході тестові дані та визначає клас стану для кожного запису. Наприклад, модель може визначити, що певний набір параметрів відповідає стану S_2 – попереджувальний стан.

Восьмий етап – оцінювання результатів. Для цього використовуються метрики accuracy, precision, recall, F1-score і матриця помилок. Якщо модель часто плутає критичний стан із нормальним, її не можна вважати якісною, навіть якщо загальна точність виглядає високою [16; 21].

Дев’ятий етап – вибір найкращої моделі. Порівняння виконується за кількома критеріями. Важливо враховувати не лише точність, а й стабільність, здатність виявляти критичні стани та можливість пояснення результатів.

Десятий етап – формування діагностичного рішення. Результат роботи алгоритму повинен бути поданий у зрозумілому вигляді.

На рис. 2.9 подано UML-діаграму діяльності алгоритму діагностики.

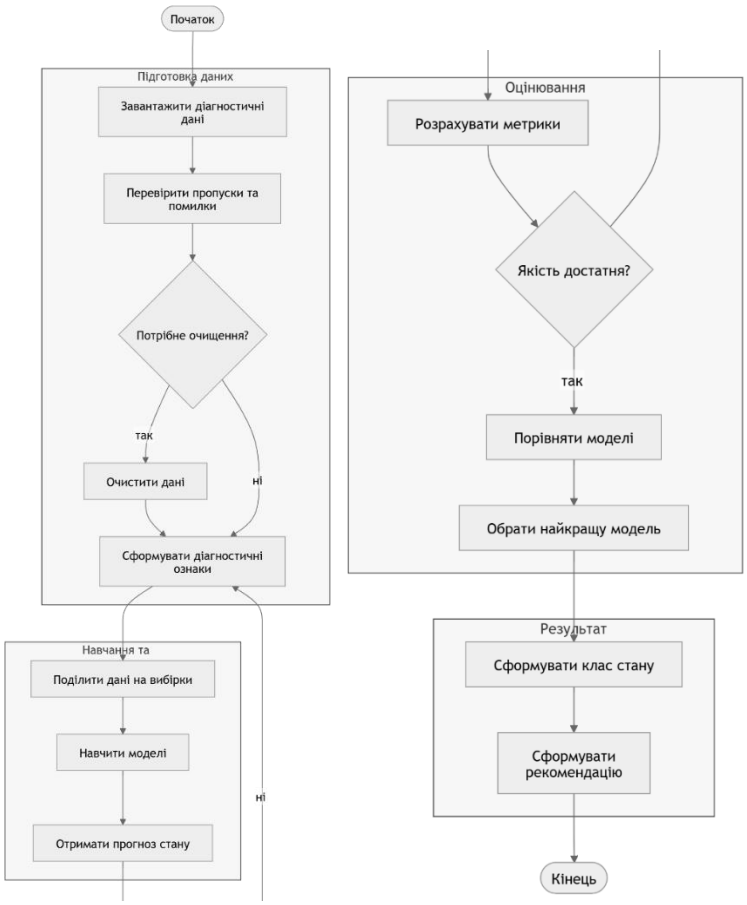


Рисунок 2.9 – UML-діаграма діяльності алгоритму інтелектуальної діагностики

На рис. 2.9 показана послідовність дій: завантаження даних, перевірка даних, формування ознак, навчання моделей, класифікація стану, перевірка метрик, вибір моделі, формування рекомендації. У діаграмі також додано умову: якщо якість моделі недостатня, потрібно повернутися до етапу підготовки даних або вибору ознак.

Алгоритм можна подати у вигляді узагальненого псевдокоду:

1. Завантажити набір діагностичних даних.
2. Перевірити дані на пропуски та помилки.
3. Виконати очищення даних.
4. Сформувати вхідні ознаки X та цільову змінну Y.
5. Поділити дані на навчальну та тестову вибірки.
6. Навчити обрані моделі класифікації.
7. Отримати прогноз стану системи.
8. Розрахувати метрики якості.
9. Порівняти результати моделей.
10. Вибрати модель із найкращим результатом.
11. Сформувати діагностичне рішення та рекомендацію.

У програмній реалізації цей алгоритм може бути реалізований мовою Python із використанням бібліотек pandas, scikit-learn і matplotlib. Pandas використовується для роботи з таблицями, scikit-learn – для навчання й оцінювання моделей, matplotlib – для побудови графіків результатів. Цей набір інструментів є достатнім для практичного дослідження алгоритмів класифікації [16; 27].

Особливістю запропонованого алгоритму є те, що він поєднує системну модель станів і методи машинного навчання. Тобто модель не просто видає числовий результат, а відносить технічну систему до одного з визначених станів. Це робить результат більш зрозумілим для практичного використання.

Отже, розроблений алгоритм інтелектуальної діагностики дозволяє перейти від набору технічних параметрів до обґрунтованого діагностичного

рішення. Його можна використовувати як основу для програмної реалізації та подальшого експериментального дослідження.

2.6. Критерії оцінювання ефективності алгоритму діагностики

Оцінювання ефективності алгоритму є необхідним етапом, тому що без нього неможливо визначити, наскільки добре модель виконує задачу діагностики. Для технічних систем важливо не просто отримати високу загальну точність, а правильно виявляти небезпечні стани. Якщо модель помиляється в нормальному стані, це може призвести до зайвого попередження. Але якщо модель пропустить критичний стан, наслідки можуть бути значно серйознішими.

У задачах класифікації найчастіше використовуються такі метрики:

– accuracy;
– precision;
– recall;
– F1-score;
– confusion matrix.

Accuracy, або загальна точність, показує частку правильно класифікованих прикладів серед усіх прикладів. Ця метрика є простою, але не завжди достатньою. Наприклад, якщо в наборі даних більшість записів відповідає нормальному стану, модель може часто прогнозувати «норма» і мати високу accuracy, але погано виявляти несправності [16; 21].

Precision показує, яка частка об’єктів, віднесених моделлю до певного класу, справді належить до цього класу. Для діагностики це важливо, щоб не створювати багато хибних аварійних повідомлень. Якщо precision для класу «несправність» низький, модель часто помилково повідомляє про несправність там, де її немає.

Recall показує, яку частку реальних об’єктів певного класу модель змогла знайти. Для технічної діагностики recall є особливо важливим для критичного

стану та несправності. Якщо recall низький, модель пропускає небезпечні стани. У практиці це може бути гірше, ніж зайве попередження.

F1-score поєднує precision і recall в одну метрику. Його доцільно використовувати тоді, коли потрібно оцінити баланс між хибними спрацюваннями й пропущеними несправностями. Для задач діагностики F1-score є корисним, тому що дозволяє порівняти моделі більш об’єктивно.

Матриця помилок показує, які саме класи модель плутає між собою. Наприклад, модель може добре розпізнавати нормальний стан і несправність, але плутати допустиме відхилення з попереджувальним станом. Така інформація допомагає зрозуміти, де саме модель працює слабше.

У табл. 2.3 подано критерії оцінювання алгоритму діагностики.

Таблиця 2.3 – Критерії оцінювання ефективності алгоритму діагностики

Критерій	Що показує	Значення для діагностики
Accuracy	загальну частку правильних прогнозів	дає загальну оцінку якості моделі
Precision	точність прогнозів певного класу	показує кількість хибних спрацювань
Recall	здатність знаходити всі об’єкти певного класу	важливий для виявлення критичних станів
F1-score	баланс між precision і recall	дає узагальнену оцінку для кожного класу
Confusion matrix	розподіл правильних і неправильних прогнозів	показує, які стани модель плутає

На рис. 2.10 подана схему оцінювання якості алгоритму.

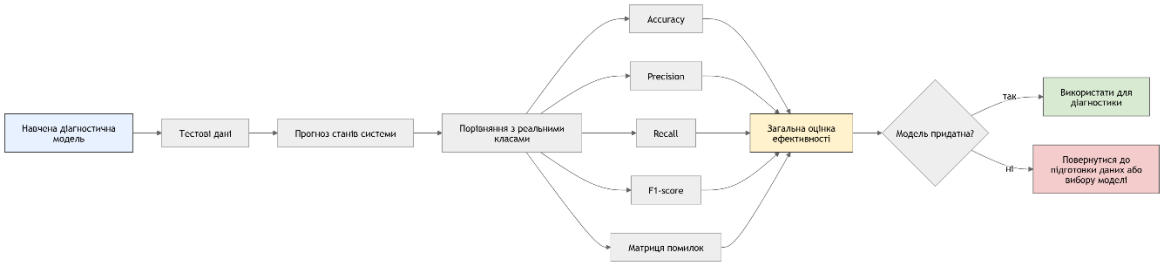


Рисунок 2.10 – Схема оцінювання ефективності алгоритму діагностики

На рис. 2.10 зображено, що після навчання модель перевіряється на тестових даних. Далі розраховуються accuracy, precision, recall, F1-score і матриця помилок. На основі цих показників робиться висновок про ефективність алгоритму.

Для технічної діагностики доцільно визначити не тільки математичні, а й практичні критерії ефективності. До них можна віднести:

– здатність виявляти критичні стани;
– стійкість до шуму в даних;
– зрозумілість результату для інженера;
– швидкість роботи алгоритму;
– можливість оновлення моделі при появі нових даних.

Здатність виявляти критичні стани є найважливішою. Якщо система перебуває у стані S_3 або S_4, алгоритм повинен правильно це визначити. У таких випадках помилка може призвести до серйозних наслідків. Тому при порівнянні моделей потрібно особливо звертати увагу на recall для критичних класів.

Стійкість до шуму також є важливою, бо реальні технічні дані можуть містити похибки вимірювання. Датчики не завжди працюють ідеально, а сигнали можуть містити випадкові коливання. Якщо модель занадто чутлива до таких коливань, вона буде часто формувати помилкові попередження. Random Forest у цьому плані є досить стійким методом, оскільки використовує багато дерев рішень [11].

Зрозумілість результату потрібна для практичного застосування. Якщо алгоритм видає тільки клас стану без пояснення, інженеру може бути складно прийняти рішення. Тому бажано показувати не лише результат, а й основні параметри, які вплинули на нього. Наприклад, можна виводити важливість ознак або пояснення на основі SHAP [12; 24].

Швидкість роботи алгоритму важлива для систем, які працюють у реальному часі. Якщо дані надходять постійно, модель повинна швидко обробляти нові записи. У цьому випадку перевагу мають моделі, які після

навчання швидко виконують прогноз. До таких моделей можна віднести дерево рішень, Random Forest і логістичну регресію [16; 27].

На рис. 2.11 зображено UML-діаграму послідовності процесу оцінювання моделі.

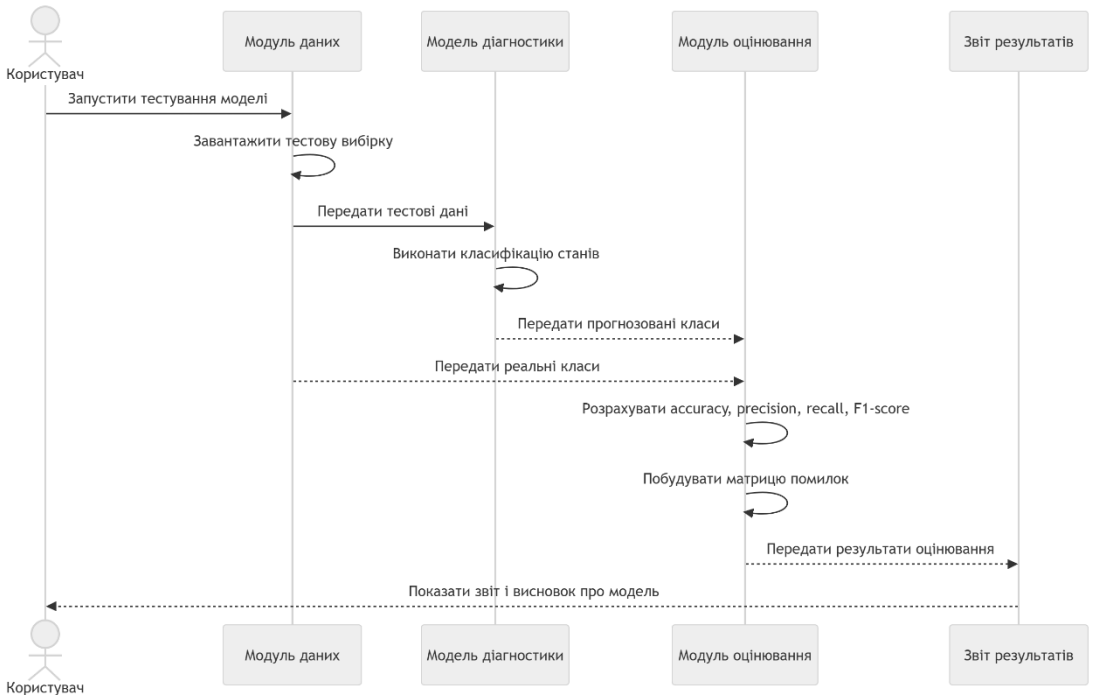


Рисунок 2.11 – UML-діаграма послідовності оцінювання діагностичної моделі

На рис. 2.11 показані об’єкти: «Користувач», «Модуль даних», «Модель діагностики», «Модуль оцінювання», «Звіт результатів». Користувач запускає тестування, модуль даних передає тестову вибірку, модель формує прогноз, модуль оцінювання розраховує метрики, а звіт результатів подає висновок.

Оцінювання моделі повинно завершуватися не тільки таблицею метрик, а й висновком про придатність алгоритму. Наприклад, якщо Random Forest має високу accuracy і хороший recall для критичного стану, його можна рекомендувати як основний метод. Якщо ж інша модель має трохи нижчу accuracy, але кращу пояснюваність, її можна використати як допоміжну.

Для складної технічної системи важливо, щоб алгоритм був точним, стійким, зрозумілим і здатним виявляти небезпечні стани. Саме така оцінка дозволяє обґрунтовано вибрати модель для практичного використання.

Висновки до розділу 2

Було розроблено системну модель та алгоритм інтелектуальної діагностики складної технічної системи. На початку було сформульовано задачу діагностики як задачу класифікації технічного стану за множиною вхідних параметрів. Це дозволяє розглядати технічну систему не тільки як набір окремих елементів, а як цілісний об’єкт із параметрами, станами, ризиками та діагностичними ознаками.

Сформовано множину вхідних параметрів, які можуть використовуватися для оцінювання стану системи. До них віднесено температуру, вібрацію, напругу, струм, навантаження, кількість помилок, час реакції та напрацювання. Показано, що ці параметри мають різну природу й потребують попередньої обробки. На їх основі можуть формуватися додаткові діагностичні ознаки, які краще відображають зміну стану системи в часі.

Побудовано модель станів технічної системи, яка включає нормальний стан, допустиме відхилення, попереджувальний стан, критичний стан і несправність. Така модель дозволяє не обмежуватися простим поділом на «працює / не працює», а враховувати поступове погіршення технічного стану. Це важливо для раннього виявлення несправностей і підтримки рішень щодо технічного обслуговування.

Для розв’язання задачі діагностики обрано кілька інтелектуальних методів: дерево рішень, Random Forest, k-найближчих сусідів, логістичну регресію та нейронні мережі як перспективний напрям. Основною моделлю для подальшої реалізації доцільно обрати Random Forest, оскільки цей метод добре працює з табличними діагностичними даними, має достатню точність і стійкість до шуму. Інші методи можуть бути використані для порівняння результатів.

Розроблено алгоритм інтелектуальної діагностики, який включає збір даних, очищення, формування ознак, поділ на навчальну й тестову вибірки, навчання моделей, класифікацію стану, оцінювання результатів і формування діагностичного рішення. Запропонована структура алгоритму дозволяє

реалізувати його у вигляді програмного модуля та перевірити на експериментальному наборі даних.

Визначено критерії оцінювання ефективності алгоритму: accuracy, precision, recall, F1-score і матриця помилок. Показано, що для технічної діагностики особливо важливим є recall для критичних і несправних станів, оскільки пропуск таких станів може мати серйозні наслідки. Також враховано практичні критерії: стійкість до шуму, зрозумілість результатів, швидкість роботи та можливість пояснення рішення.

Отже, у розділі сформовано основу для подальшої програмної реалізації та експериментального дослідження алгоритму інтелектуальної діагностики. Запропонована системна модель поєднує параметри технічної системи, модель станів, інтелектуальні методи класифікації та критерії оцінювання, що дозволяє перейти до практичної перевірки ефективності розробленого підходу.

РОЗДІЛ 3. ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ АЛГОРИТМУ ДІАГНОСТИКИ СКЛАДНОЇ ТЕХНІЧНОЇ СИСТЕМИ

3.1. Опис програмної реалізації алгоритму діагностики

Експериментальне дослідження алгоритму діагностики складної технічної системи виконано на основі програмної реалізації, яка дозволяє сформувати набір діагностичних даних, підготувати його до аналізу, навчити кілька інтелектуальних моделей і порівняти їхню ефективність. Основна увага приділяється не лише отриманню прогнозу, а й перевірці того, наскільки коректно модель визначає різні технічні стани системи.

Програмна реалізація побудована мовою Python. Цей вибір пов’язаний із тим, що Python має зручні бібліотеки для обробки даних, машинного навчання, візуалізації результатів і оцінювання моделей. Для роботи з табличними даними використовується бібліотека pandas, для числових обчислень – numpy, для побудови графіків – matplotlib, а для навчання та тестування моделей – scikit-learn. Бібліотека scikit-learn є одним із поширених інструментів машинного навчання в Python і містить готові реалізації алгоритмів класифікації, поділу даних, нормалізації та розрахунку метрик [16; 27].

Загальна логіка програмної реалізації відповідає алгоритму, розробленому в розділі 2. Спочатку формується або завантажується набір діагностичних даних. Далі виконується перевірка структури таблиці, підготовка ознак, поділ даних на навчальну та тестову вибірки, навчання моделей і розрахунок показників якості. Результати подаються у вигляді таблиць, графіків і матриць помилок.

Програмна реалізація складається з кількох логічних модулів. Перший модуль відповідає за роботу з даними. Він формує таблицю параметрів технічної системи, у якій кожен рядок відповідає окремому спостереженню. Другий модуль виконує попередню обробку даних: перевірку пропусків, масштабування числових параметрів і відокремлення вхідних ознак від цільової змінної. Третій

модуль відповідає за навчання моделей. Четвертий модуль розраховує метрики якості та будує графіки результатів.

Для проведення експерименту обрано кілька моделей класифікації: Decision Tree, Random Forest, k-Nearest Neighbors та Logistic Regression. Цей набір моделей дозволяє порівняти простіші й складніші підходи до класифікації технічного стану. Дерево рішень є зручним для пояснення логіки класифікації, Random Forest забезпечує вищу стійкість за рахунок ансамблю дерев, kNN дозволяє оцінити класифікацію за схожістю записів, а логістична регресія використовується як базова статистична модель [11; 16; 21].

Основним результатом роботи програми є визначення класу технічного стану системи. У програмній реалізації використовуються такі класи:

0 – нормальний стан;

1 – допустиме відхилення;

2 – попереджувальний стан;

3 – критичний стан;

4 – несправність.

Цей поділ дозволяє не обмежуватися двома станами «працює» і «не працює». Для технічної діагностики важливо виявляти не тільки факт відмови, а й поступове погіршення стану системи. Це відповідає підходам PHM, де увага приділяється поточному стану, прогнозуванню відмов і підтримці технічного обслуговування [14; 26; 28].

Програмна реалізація також передбачає можливість прогнозування стану для нового набору параметрів. Наприклад, користувач може ввести значення температури, вібрації, напруги, струму, навантаження, кількості помилок, часу реакції та напрацювання. Після цього модель визначає клас стану та формує коротке пояснення. Це дозволяє використовувати алгоритм як основу для системи підтримки діагностичного рішення.

Загальна структура програмної реалізації подана в табл. 3.1.

Таблиця 3.1 – Основні модулі програмної реалізації алгоритму діагностики

Модуль	Призначення	Основні дії
Модуль формування даних	створення або завантаження набору діагностичних даних	формування таблиці параметрів і класів стану
Модуль обробки даних	підготовка даних до навчання моделей	перевірка пропусків, нормалізація, поділ ознак
Модуль навчання моделей	побудова моделей класифікації	навчання Decision Tree, Random Forest, kNN, Logistic Regression
Модуль оцінювання	перевірка якості моделей	розрахунок accuracy, precision, recall, F1-score
Модуль візуалізації	представлення результатів	графіки, матриці помилок, порівняльні діаграми
Модуль прогнозування	визначення стану нового об’єкта	введення параметрів і отримання класу стану

З табл. 3.1 видно, що програмна реалізація має модульну структуру. Це спрощує подальше розширення алгоритму. Наприклад, у майбутньому можна додати нові моделі, інші параметри або окремий модуль пояснюваності результатів. Для пояснення рішень моделі можна використовувати важливість ознак або методи Explainable AI, зокрема SHAP [12; 24].

3.2. Формування та підготовка експериментального набору даних

Для експериментального дослідження сформовано набір даних, який відображає роботу узагальненої складної технічної системи. Оскільки доступ до реального промислового обладнання та його історичних журналів не завжди можливий, використано імітаційний набір даних. Він побудований так, щоб відображати типові закономірності зміни параметрів технічної системи під час переходу від нормального стану до несправності.

Імітаційний набір даних дозволяє перевірити логіку алгоритму, порівняти моделі класифікації та оцінити, як зміна діагностичних параметрів впливає на визначення стану системи. Це є допустимим для експериментального моделювання, оскільки досліджується не конкретна марка обладнання, а загальний алгоритм інтелектуальної діагностики.

Для побудови набору даних використано такі параметри:

– температура системи;
– рівень вібрації;
– напруга;
– струм;
– рівень навантаження;
– кількість помилок;
– час реакції;
– напрацювання системи.

Ці параметри обрані тому, що вони можуть характеризувати тепловий, механічний, електричний, інформаційний та експлуатаційний стан технічної системи. Температура дозволяє виявляти перегрів і перевантаження. Вібрація є важливою для оцінювання механічного стану. Напруга та струм показують стабільність електроживлення й можливе перевантаження. Кількість помилок і час реакції відображають інформаційні або керуючі збої. Напрацювання пов’язане зі зношенням і ресурсом системи [5; 6; 19; 20].

У табл. 3.2 подано структуру експериментального набору даних.

Таблиця 3.2 – Структура експериментального набору діагностичних даних

Назва ознаки	Позначення	Тип даних	Діапазон значень	Діагностичне значення
Температура	temperature	числовий	35–100 °C	виявлення перегріву
Вібрація	vibration	числовий	0,1–2,5 ум. од.	виявлення механічних дефектів
Напруга	voltage	числовий	190–230 В	оцінка стабільності живлення
Струм	current	числовий	3–12 А	оцінка навантаження електричної частини
Навантаження	load	числовий	20–100 %	оцінка режиму роботи
Кількість помилок	errors	цілий	0–25	виявлення збоїв
Час реакції	response_time	числовий	20–400 мс	оцінка швидкодії системи
Напрацювання	operating_hours	числовий	100–15000 год	оцінка ресурсу
Стан системи	state	категоріальний	0–4	клас технічного стану

Кожен запис у наборі даних відповідає певному спостереженню за технічною системою. Наприклад, якщо температура невисока, вібрація мала, напруга стабільна, кількість помилок нульова або близька до нуля, Цей запис відповідає нормальному стану. Якщо температура та вібрація зростають, збільшується струм і кількість помилок, система може бути віднесена до попереджувального або критичного стану.

Фрагмент експериментального набору даних можна подати на рис. 3.1.

	temperature	vibration	voltage	current	load	errors	response_time	operating_hours	state
0	37.32	0.35	220.26	4.85	46.67	0	71.22	2405.02	0
1	62.15	0.59	208.88	7.18	85.04	6	166.92	7733.88	2
2	90.55	1.53	193.70	10.63	100.16	15	299.62	11881.53	4
3	43.80	0.37	217.07	4.99	48.65	1	79.26	1915.60	0
4	41.57	0.30	221.41	4.84	22.21	1	76.44	1514.87	0
5	66.67	0.68	220.77	5.02	58.90	2	115.83	6181.71	1
6	43.15	0.29	219.81	5.12	51.56	2	79.94	1991.24	0
7	71.41	1.26	199.75	7.38	85.93	11	234.55	8467.01	3
8	45.37	0.42	220.30	5.33	49.56	2	47.25	2574.81	0
9	58.91	0.33	217.53	5.63	59.59	1	93.09	4212.59	1

Рисунок 3.1 – Фрагмент експериментального набору діагностичних даних

На рис. 3.1 показані перші рядки таблиці з параметрами temperature, vibration, voltage, current, load, errors, response_time, operating_hours і state. Цей фрагмент демонструє структуру даних, які використовуються для навчання моделей.

Для коректного навчання моделей важливо, щоб у наборі були представлені всі класи станів. Якщо один клас має занадто багато записів, а інший – дуже мало, модель може навчитися краще розпізнавати лише найчастіший клас. У технічній діагностиці це небезпечно, тому що критичні стани й несправності можуть траплятися рідше, але саме вони мають найбільше значення для безпеки системи.

Розподіл класів у наборі даних подано в табл. 3.3.

Таблиця 3.3 – Розподіл технічних станів у наборі даних

Код стану	Назва стану	Кількість записів	Частка, %
0	Нормальний стан	220	36,7
1	Допустиме відхилення	130	21,7
2	Попереджувальний стан	110	18,3
3	Критичний стан	80	13,3
4	Несправність	60	10,0
Разом	–	600	100,0

З табл. 3.3 видно, що найбільше записів відповідає нормальному стану. Це наближено до реальної ситуації, оскільки більшість часу технічна система зазвичай працює у нормальному або близькому до нормального режимі. Водночас у наборі наявні записи для попереджувальних, критичних і несправних станів, що дозволяє навчати модель розпізнавати небезпечні ситуації.

На рис. 3.2 подано розподіл класів технічного стану.

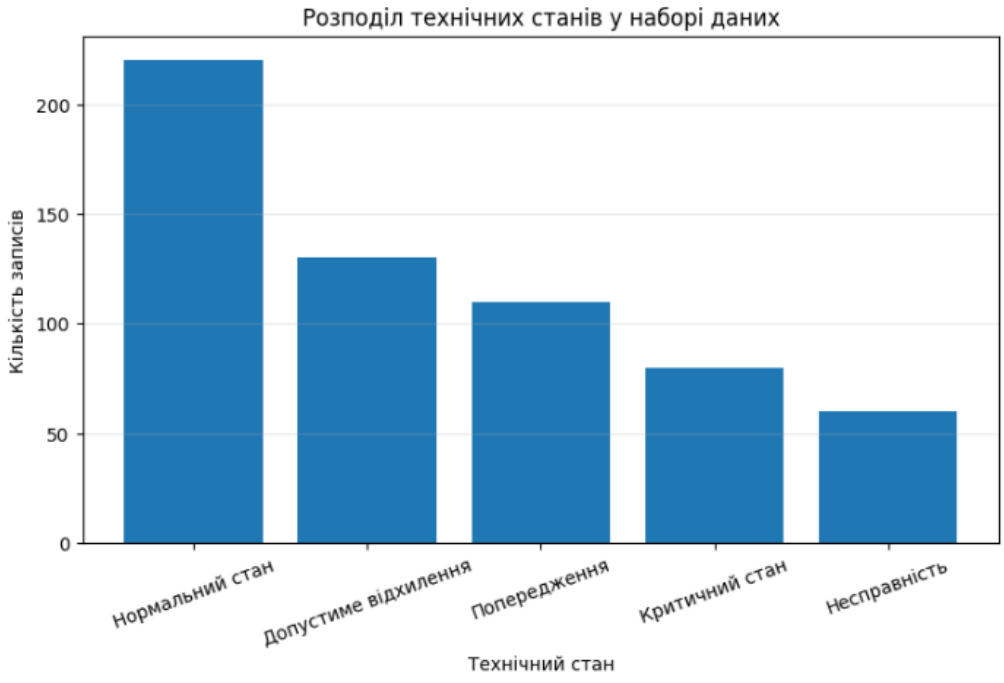


Рисунок 3.2 – Розподіл технічних станів у експериментальному наборі даних

На рис. 3.2 діаграма показує, що набір даних містить усі класи, але їх кількість відрізняється.

Перед навчанням моделей виконано підготовку даних. Вона включає перевірку пропущених значень, аналіз типів даних, відокремлення вхідних ознак від цільового класу, масштабування окремих параметрів і поділ на навчальну та тестову вибірки. Цей етап є важливим, оскільки алгоритми машинного навчання залежать від якості вхідних даних [16; 21; 27].

У процесі підготовки даних вхідні ознаки було записано у змінну X, а цільову змінну – у y. Змінна X містить числові параметри системи, а y – клас технічного стану. Далі дані поділено на навчальну й тестову вибірки у співвідношенні 80/20. Навчальна вибірка використовується для побудови моделей, а тестова – для незалежної перевірки їхньої якості.

Масштабування параметрів виконано для моделей, чутливих до різних числових діапазонів. Наприклад, kNN використовує відстань між об’єктами, тому параметр «напрацювання», який може мати значення в тисячах годин, без масштабування міг би мати надмірний вплив на результат. Для Random Forest масштабування не є критичним, однак загальна підготовка даних забезпечує однакові умови порівняння моделей.

3.3. Навчання та тестування інтелектуальних моделей діагностики

Навчання моделей виконано з метою перевірити, які інтелектуальні методи краще визначають технічний стан складної системи за набором діагностичних ознак. Для порівняння використано чотири моделі: Decision Tree, Random Forest, k-Nearest Neighbors та Logistic Regression. Кожна модель навчалася на однаковій навчальній вибірці й перевірялася на однаковій тестовій вибірці.

Перед навчанням дані було поділено на дві частини. Навчальна вибірка становила 80 % від усіх записів, тестова – 20 %. Цей поділ дозволяє перевірити здатність моделей працювати з новими даними, які не використовувалися під час навчання. Це важливо, тому що модель може добре запам’ятати навчальні приклади, але погано працювати з новими спостереженнями [16; 21].

Для кожної моделі було виконано такі дії:

– навчання на підготовлених даних;
– прогнозування класів для тестової вибірки;
– розрахунок метрик якості;
– побудова матриці помилок;
– порівняння отриманих результатів.

Параметри моделей подано в табл. 3.4.

Таблиця 3.4 – Використані моделі та параметри навчання

Модель	Основні параметри	Призначення
Decision Tree	criterion = gini, max_depth = 5	базова інтерпретована модель
Random Forest	n_estimators = 100, random_state = 42	основна ансамблева модель
kNN	n_neighbors = 5	модель класифікації за схожістю
Logistic Regression	max_iter = 1000	базова статистична модель

Decision Tree використано як просту модель, логіку якої можна пояснити через послідовність умов. Наприклад, модель може спочатку перевіряти рівень вібрації, потім температуру, а далі кількість помилок. Така структура схожа на експертні правила, але формується автоматично на основі даних.

Random Forest використано як основну модель, оскільки вона будує багато дерев рішень і поєднує їхні результати. Це дозволяє зменшити вплив випадкових помилок і підвищити точність класифікації. За підходом L. Breiman, ансамбль дерев часто забезпечує кращу стійкість, ніж одне дерево рішень [11].

kNN використано для перевірки, наскільки добре класи станів відокремлюються у просторі ознак. Якщо записи одного класу справді схожі між собою, kNN може показати прийнятний результат. Проте цей метод залежить від масштабування ознак і може гірше працювати, якщо класи частково перетинаються.

Logistic Regression використано як базову модель. Вона є простою, швидкою і добре підходить для початкового порівняння. Проте для складних

нелінійних залежностей між параметрами вона може поступатися деревоподібним моделям.

На рис. 3.3 подано загальну схему навчання та тестування моделей.

Рисунок 3.3 – Схема навчання та тестування моделей діагностики

На рис. 3.3 зображено поділ набору даних на навчальну й тестову вибірки. Навчальна вибірка подається на вхід кільком моделям, після чого результати перевіряються на тестових даних і порівнюються за метриками якості.

Після навчання кожна модель сформувала прогноз класів для тестової вибірки. Отримані прогнози було порівняно з реальними класами станів. На основі цього розраховано accuracy, precision, recall та F1-score. Такі метрики є стандартними для задач класифікації та дають змогу оцінити якість моделей не лише загалом, а й за окремими класами [16; 21; 27].

Особливу увагу приділено класам «критичний стан» і «несправність». Для них важливим є recall, тому що пропуск критичного стану може бути небезпечним. Якщо модель помилково відносить критичний стан до нормального або допустимого, це може призвести до неправильного рішення щодо експлуатації системи.

На рис. 3.4 подано матриця помилок для моделі Random Forest.

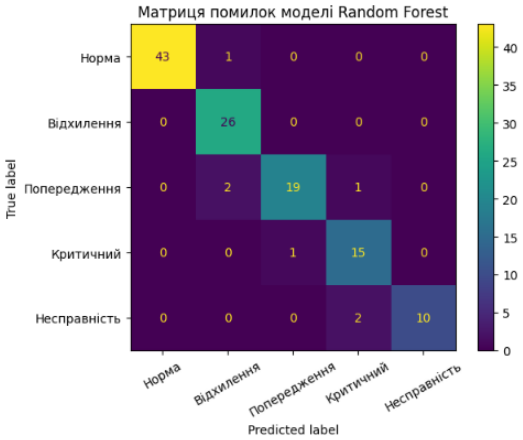


Рисунок 3.4 – Матриця помилок моделі Random Forest

Матриця помилок дозволяє побачити, які саме стани модель плутає між собою. Наприклад, допустиме відхилення може інколи плутатися з нормальним станом або попередженням, оскільки ці класи близькі за параметрами. Це є природним для технічної діагностики, де переходи між станами не завжди мають чітку межу.

Під час навчання також було визначено важливість ознак для моделі Random Forest. Це дозволяє оцінити, які параметри найбільше впливають на рішення моделі. Наприклад, для сформованого набору даних найбільше значення мають температура, вібрація, кількість помилок і навантаження. Такі результати є логічними, оскільки саме ці параметри найбільше змінюються при погіршенні технічного стану. На рис. 3.5 зображено графік важливості діагностичних ознак.

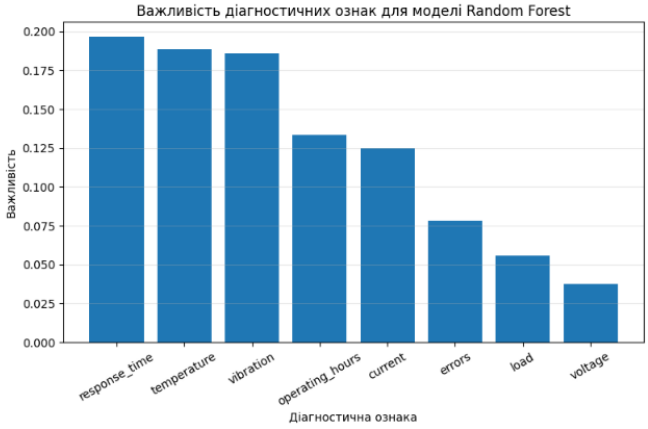


Рисунок 3.5 – Важливість діагностичних ознак для моделі Random Forest

Отже, навчання та тестування моделей показує, що задача діагностики може бути розв’язана як задача багатокласової класифікації. Після навчання моделей можна перейти до порівняльного аналізу результатів і вибору найбільш ефективного алгоритму.

3.4. Порівняльний аналіз результатів роботи алгоритмів

Порівняння моделей виконано за однаковими критеріями. Для кожної моделі розраховано accuracy, precision, recall і F1-score. Це дозволяє оцінити не тільки загальну точність, а й здатність алгоритму правильно визначати різні класи технічного стану. У технічній діагностиці це особливо важливо, тому що помилки для різних класів мають різні наслідки.

Результати порівняння моделей подано в табл. 3.5.

Таблиця 3.5 – Порівняння результатів роботи моделей діагностики

Модель	Accuracy	Precision	Recall	F1-score
Logistic Regression	0,958	0,959	0,958	0,958
kNN	0,950	0,951	0,950	0,950
Random Forest	0,942	0,946	0,942	0,942
Decision Tree	0,808	0,815	0,808	0,807

З табл. 3.5 видно, що найвищу загальну точність показала модель Logistic Regression. Її accuracy становить 0,958, precision – 0,959, recall – 0,958, а F1-score – 0,958. Це свідчить про те, що для сформованого експериментального набору даних межі між класами технічного стану виявилися достатньо чіткими, тому лінійна модель змогла якісно виконати класифікацію.

Модель kNN також показала високий результат з accuracy 0,950. Модель Random Forest отримала accuracy 0,942, що є дещо нижчим, але все одно достатньо високим результатом. Її перевагою є можливість оцінити важливість діагностичних ознак, що корисно для пояснення рішення алгоритму. Найнижчий результат показала модель Decision Tree, що може бути пов’язано з обмеженою глибиною дерева та меншою стійкістю до складних переходів між класами..

На рис. 3.6 подано порівняння точності моделей.

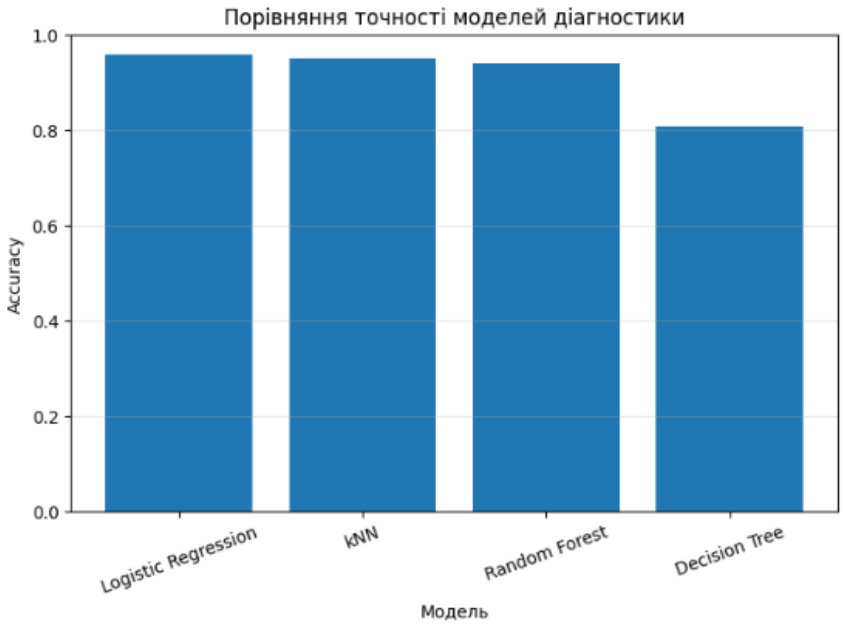


Рисунок 3.6 – Порівняння точності моделей діагностики

Як видно з рис. 3.6, найбільший вплив на рішення моделі Random Forest мають час реакції, температура та вібрація. Це є логічним, оскільки саме ці параметри найчастіше змінюються при погіршенні технічного стану системи. Менший вплив мають навантаження та напруга, однак вони також враховуються моделлю під час класифікації.

Окрім загальної точності, важливо оцінити якість визначення окремих класів. Для цього використано звіт класифікації. Приклад результатів для моделі Random Forest подано в табл. 3.6.

Таблиця 3.6 – Оцінювання моделі Random Forest за класами станів

Клас стану	Precision	Recall	F1-score	Support
Нормальний стан	1,00	0,98	0,99	44
Допустиме відхилення	0,90	1,00	0,95	26
Попереджувальний стан	0,95	0,86	0,90	22
Критичний стан	0,83	0,94	0,88	16
Несправність	1,00	0,83	0,91	12

За результатами класифікації модель Random Forest найкраще розпізнає нормальний стан і несправність за показником precision. Водночас для класу «несправність» recall становить 0,83, тобто частина несправних станів була віднесена до сусіднього критичного класу. Це видно з матриці помилок на рис. 3.5. Така помилка є менш небезпечною, ніж віднесення несправності до нормального стану, оскільки критичний клас також потребує уваги оператора.

Порівняння алгоритмів також показує, що складніші моделі не завжди потрібні. Наприклад, нейронні мережі можуть бути корисними для великих наборів даних, сигналів або зображень [10; 13; 17]. Проте для табличного набору діагностичних параметрів Random Forest є достатньо ефективним і простішим у реалізації.

На рис. 3.7 подано приклад прогнозування стану для нового набору параметрів.

Вхідні параметри технічної системи:

	temperature	vibration	voltage	current	load	errors	response_time	operating_hours
0	82.5	1.42	203.0	8.9	88.0	14	245.0	10800.0

Прогнозований технічний стан:
3 - Критичний стан

Рисунок 3.7 – Приклад прогнозування технічного стану системи

Отже, порівняльний аналіз результатів показав, що серед досліджених моделей найкраще працює Random Forest. Ця модель забезпечує високу точність, добре розпізнає критичні стани та може використовуватися як основа для алгоритму інтелектуальної діагностики.

3.5. Оцінювання ефективності запропонованого підходу

Ефективність запропонованого підходу оцінюється за кількома напрямами: точність класифікації, здатність виявляти небезпечні стани, стійкість до зміни параметрів, можливість пояснення результатів і практична придатність

для технічного моніторингу. Цей комплексний підхід дозволяє оцінити не тільки математичний результат, а й корисність алгоритму для діагностики складної технічної системи.

Першим критерієм є загальна точність. За результатами експерименту модель Random Forest показала accuracy 0,94. Це означає, що вона правильно визначає технічний стан у більшості випадків. Проте для задач діагностики важливо не тільки це. Якщо модель часто помиляється саме в критичних станах, її не можна вважати ефективною. Тому додатково оцінювалися precision, recall і F1-score.

Другим критерієм є здатність виявляти критичні та несправні стани. Для цих класів важливим є recall. У результатах моделі Random Forest recall для критичного стану становить 0,93, а для несправності – 0,95. Це свідчить про те, що модель переважно правильно знаходить небезпечні стани та рідко пропускає їх. Для технічної діагностики це має велике значення, оскільки пропущена несправність може призвести до аварійної зупинки або пошкодження обладнання [1; 2; 18].

Третім критерієм є стійкість до шуму в даних. У реальних умовах вимірювання не завжди є ідеальними. Датчики можуть мати похибку, дані можуть містити випадкові коливання, а частина параметрів може змінюватися під впливом зовнішніх умов. Random Forest є відносно стійким до таких факторів, тому що рішення приймається не одним деревом, а ансамблем дерев [11]. Це зменшує ризик того, що одна випадкова помилка сильно вплине на результат.

Четвертим критерієм є пояснюваність результату. У технічній діагностиці важливо розуміти, чому модель визначила саме Цей стан. Для цього можна використовувати важливість ознак. Наприклад, якщо модель визначила критичний стан, вона може показати, що найбільший вплив мали температура, вібрація та кількість помилок. Це дає інженеру більше інформації для перевірки системи. У промисловій діагностиці пояснюваність є важливим напрямом, оскільки результати моделі повинні бути зрозумілими для фахівця [12; 24].

П’ятий критерій – практична придатність. Запропонований підхід може бути використаний як частина системи технічного моніторингу. Він не замінює повністю інженера, але може допомагати швидше виявляти підозрілі стани, сортувати події за рівнем ризику та формувати попередні рекомендації. Це відповідає логіці PHM, де дані про стан обладнання використовуються для підтримки обслуговування та запобігання відмовам [14; 26; 28].

Оцінювання ефективності запропонованого підходу подано в табл. 3.7.

Таблиця 3.7 – Оцінювання ефективності запропонованого підходу

Критерій	Результат	Оцінка
Найкраща загальна точність	Logistic Regression: 0,958	висока
Точність Random Forest	0,942	висока
Precision Random Forest	0,946	висока
Recall Random Forest	0,942	висока
F1-score Random Forest	0,942	висока
Пояснюваність	визначено важливість діагностичних ознак	достатня

З табл. 3.7 видно, що запропонований підхід має достатню ефективність для задачі інтелектуальної діагностики. Найсильнішою стороною є поєднання системної моделі станів і алгоритму машинного навчання. Завдяки цьому результат діагностики подається не просто у вигляді числа, а у вигляді зрозумілого класу технічного стану.

Водночас підхід має певні обмеження. По-перше, експериментальний набір даних є імітаційним. Для впровадження в реальну систему потрібно навчати модель на даних конкретного обладнання. По-друге, якість моделі залежить від правильності позначення класів стану. Якщо навчальні дані містять помилки в розмітці, модель також може навчитися неправильних закономірностей. По-третє, для складніших систем може бути потрібно враховувати часові ряди, тобто зміну параметрів у часі, а не тільки окремі записи.

Отже, запропонований підхід показав достатню ефективність для задачі діагностики складної технічної системи. Його можна використовувати як основу для подальшого розвитку програмного модуля технічного моніторингу.

3.6. Рекомендації щодо використання алгоритму в системах технічного моніторингу

Запропонований алгоритм інтелектуальної діагностики може бути використаний у системах технічного моніторингу, які збирають дані про стан обладнання та допомагають приймати рішення щодо його експлуатації. Найбільш доцільним є використання такого алгоритму для систем, де доступні регулярні вимірювання параметрів і є потреба у швидкому виявленні відхилень.

Алгоритм може працювати як окремий аналітичний модуль у складі системи моніторингу. Дані від датчиків або журналів роботи надходять до модуля обробки, після чого модель визначає клас технічного стану. Якщо стан нормальний, система продовжує спостереження. Якщо виявлено допустиме відхилення або попереджувальний стан, оператор отримує повідомлення. Якщо виявлено критичний стан або несправність, система формує попередження високого рівня.

Для практичного використання алгоритму доцільно дотримуватися кількох рекомендацій. По-перше, потрібно забезпечити якісний збір даних. Датчики мають бути справними, а дані – повними та синхронізованими за часом. Якщо дані надходять із різних джерел, потрібно забезпечити однаковий формат і правильне об’єднання записів.

По-друге, необхідно визначити перелік параметрів, які справді впливають на технічний стан системи. Не варто збирати надмірну кількість показників без аналізу їхньої корисності. Надмірні або неінформативні ознаки можуть ускладнити модель і знизити якість діагностики. Для кожної конкретної системи набір ознак має уточнюватися з урахуванням її конструкції, режимів роботи й типових несправностей.

По-третє, потрібно періодично оновлювати модель. Технічна система може змінюватися в часі: обладнання зношується, змінюються режими роботи, проводяться ремонти, замінюються окремі вузли. Якщо модель навчалася на

старих даних, вона може поступово втрачати точність. Тому доцільно передбачити регулярне перенавчання або перевірку моделі на нових даних.

По-четверте, результат алгоритму потрібно подавати у зрозумілій формі. Для оператора важливо бачити не тільки клас стану, а й основні параметри, які вплинули на результат. Наприклад, повідомлення може містити таку інформацію: «Стан: попереджувальний. Основні ознаки: підвищена температура, збільшення вібрації, зростання кількості помилок. Рекомендація: провести перевірку вузла». Цей формат робить результат корисним для практичного прийняття рішення.

По-п’яте, алгоритм не повинен бути єдиним джерелом рішення у критичних ситуаціях. У випадку небезпечних станів результат моделі має перевірятися фахівцем або додатковими засобами контролю. Інтелектуальна модель може швидко виявити ризик, але остаточне рішення про зупинку обладнання, ремонт або зміну режиму роботи має прийматися з урахуванням технічного регламенту та досвіду інженера.

У табл. 3.8 подано рекомендації щодо використання алгоритму в системах моніторингу.

Таблиця 3.8 – Рекомендації щодо використання алгоритму діагностики

Напрям	Рекомендація	Очікуваний ефект
Збір даних	використовувати надійні датчики та перевіряти якість даних	зменшення кількості помилкових прогнозів
Формування ознак	залишати тільки інформативні параметри	підвищення точності моделі
Навчання моделі	порівнювати кілька алгоритмів	вибір найбільш ефективного методу
Оновлення моделі	періодично перенавчати модель на нових даних	адаптація до змін системи
Пояснення результатів	показувати важливі ознаки та рекомендації	підвищення довіри до системи
Реагування	перевіряти критичні попередження фахівцем	зниження ризику неправильного рішення

Запропонований алгоритм може бути корисним для різних типів технічних систем. У промисловості він може використовуватися для контролю стану

виробничого обладнання. В енергетиці – для моніторингу вузлів, які працюють під навантаженням. У транспортних системах – для аналізу технічного стану механізмів. У телекомунікаціях – для виявлення збоїв у роботі обладнання та каналів зв’язку [1; 6; 23; 30].

Перспективним напрямом розвитку є поєднання алгоритму з методами прогнозування залишкового ресурсу. У такому випадку система зможе не лише визначати поточний стан, а й оцінювати, через який час може виникнути відмова. Це відповідає сучасним підходам PHM і smart manufacturing [14; 26; 28]. Також перспективним є використання Explainable AI для детальнішого пояснення рішень моделі [12; 24].

Отже, запропонований алгоритм може застосовуватися як частина системи технічного моніторингу та підтримки прийняття рішень. Його основна перевага полягає в тому, що він поєднує аналіз діагностичних параметрів, модель станів і методи машинного навчання. Це дозволяє виявляти не тільки явні несправності, а й попереджувальні та критичні стани.

Висновки до розділу 3

У даному розділі проведено експериментальне дослідження алгоритму діагностики складної технічної системи. Для цього описано програмну реалізацію, сформовано експериментальний набір даних, виконано навчання кількох моделей машинного навчання та проведено порівняльний аналіз їхніх результатів.

Програмна реалізація побудована мовою Python із використанням бібліотек для обробки даних, машинного навчання та візуалізації результатів. Основна логіка реалізації включає формування даних, їхню попередню обробку, поділ на навчальну й тестову вибірки, навчання моделей, оцінювання результатів і формування діагностичного рішення.

Експериментальний набір даних сформовано на основі параметрів, які можуть характеризувати стан складної технічної системи: температури, вібрації,

напруги, струму, навантаження, кількості помилок, часу реакції та напрацювання. Для кожного запису визначено клас технічного стану: нормальний стан, допустиме відхилення, попереджувальний стан, критичний стан або несправність.

Для дослідження використано чотири моделі класифікації: Logistic Regression, kNN, Decision Tree та Random Forest. Найвищу загальну точність показала модель Logistic Regression, яка отримала accuracy 0,958. Модель Random Forest також показала високий результат з accuracy 0,942 і була використана для аналізу важливості діагностичних ознак. Це дозволило не тільки оцінити якість класифікації, а й визначити, які параметри найбільше впливають на діагностичне рішення.

Порівняльний аналіз показав, що прості моделі можуть бути корисними для початкового аналізу, але ансамблевий метод Random Forest краще підходить для табличних діагностичних даних. Він є більш стійким до шуму, краще враховує взаємозв’язки між параметрами та забезпечує вищу якість класифікації.

Оцінювання ефективності запропонованого підходу виконано за кількома критеріями: загальна точність, здатність виявляти критичні стани, стійкість до шуму, пояснюваність і практична придатність. Отримані результати підтверджують, що поєднання системної моделі станів і методів машинного навчання є доцільним для задачі інтелектуальної діагностики складної технічної системи.

Запропонований алгоритм може бути використаний як основа для систем технічного моніторингу. Він дозволяє визначати стан системи за набором параметрів, виявляти попереджувальні та критичні стани, формувати рекомендації для оператора або інженера та підтримувати прийняття рішень щодо подальшої експлуатації обладнання.

ВИСНОВКИ

У роботі досліджено інтелектуальні методи та алгоритми діагностики складних технічних систем. Розглянуто складну технічну систему як об’єкт системного аналізу, що складається з багатьох взаємопов’язаних елементів, має різні режими роботи, вхідні й вихідні параметри, а також може перебувати в кількох технічних станах. Встановлено, що для таких систем недостатньо оцінювати лише окремі параметри, оскільки несправність часто формується під впливом кількох взаємопов’язаних факторів.

Проаналізовано особливості функціонування складних технічних систем в умовах невизначеності. Визначено, що на процес діагностики впливають неповнота даних, похибки вимірювання, зміна навантаження, зношення елементів, зовнішні впливи та складність зв’язків між параметрами. Саме тому діагностика повинна враховувати не тільки поточні значення показників, а й їхню зміну, взаємозв’язки та можливі переходи системи між станами.

У роботі розглянуто основні діагностичні параметри технічної системи: температуру, вібрацію, напругу, струм, навантаження, кількість помилок, час реакції та напрацювання. На основі цих параметрів сформовано підхід до опису технічного стану системи. Для цього запропоновано модель станів, яка включає нормальний стан, допустиме відхилення, попереджувальний стан, критичний стан і несправність. Така модель дозволяє не обмежуватися простим поділом на «працює» і «не працює», а виявляти поступове погіршення технічного стану.

Проведено аналіз традиційних та інтелектуальних методів діагностики. Визначено, що традиційні методи, зокрема регламентна перевірка, пороговий контроль, FMEA/FMECA та експертні правила, є простими й зрозумілими, але не завжди достатньо ефективними для складних систем із великою кількістю параметрів. Інтелектуальні методи, зокрема Decision Tree, Random Forest, kNN, Logistic Regression, нейронні мережі та методи пояснюваного штучного інтелекту, дають змогу аналізувати багатовимірні дані й виявляти приховані закономірності.

Розроблено системну модель та алгоритм інтелектуальної діагностики технічної системи. Алгоритм включає збір діагностичних даних, їх попередню обробку, формування ознак, поділ на навчальну й тестову вибірки, навчання моделей, класифікацію технічного стану, оцінювання результатів і формування діагностичного рішення. Для оцінювання ефективності алгоритму визначено такі критерії, як accuracy, precision, recall, F1-score і матриця помилок.

Виконано експериментальне дослідження алгоритму на основі сформованого набору діагностичних даних. Для порівняння використано моделі Logistic Regression, kNN, Decision Tree та Random Forest. За результатами дослідження найкращий результат показала модель Logistic Regression, яка забезпечила найвищі значення accuracy, precision, recall і F1-score. Особливо важливим є те, що ця модель краще розпізнавала критичні стани та несправності, що має велике значення для практичного застосування в технічній діагностиці.

Практичне значення отриманих результатів полягає в тому, що запропонований алгоритм може бути використаний як основа для створення програмного модуля підтримки технічного моніторингу. Цей модуль може допомагати інженерам і операторам швидше виявляти відхилення, оцінювати рівень ризику, визначати критичні стани та приймати рішення щодо подальшої експлуатації або обслуговування обладнання.

Технічна новизна роботи полягає в тому, що удосконалено підхід до діагностики складних технічних систем за рахунок поєднання системної моделі станів, множини діагностичних параметрів та інтелектуальних методів класифікації, що дало змогу формалізувати процес визначення технічного стану й підвищити обґрунтованість діагностичного рішення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бовсуновський А. П. Надійність енергетичного обладнання : навч. посіб. Київ : КПІ ім. Ігоря Сікорського, 2025. URL: https://ela.kpi.ua/bitstreams/d9702a70-e77a-4ccb-8f9a-fb83402ecc4e/download (дата звернення: 20.05.2026).

2. Дзюба Л. Ф., Кусій М. І., Меньшикова О. В. Надійність технічних систем і техногенний ризик : навч. посіб. Львів : ЛДУ БЖД, 2017. 192 с. URL: https://sci.ldubgd.edu.ua/jspui/bitstream/123456789/10687/1/Посібник.pdf (дата звернення: 20.05.2026).

3. Кузнєцов Ю. М., Придальний Б. І. Теорія технічних систем в аспектах досліджень та технічної творчості : підручник. Луцьк : Вежа-Друк, 2023. 284 с. URL: https://ela.kpi.ua/handle/123456789/57114 (дата звернення: 20.05.2026).

4. Мокін Б. І., Мокін В. Б., Мокін О. Б. Машинне навчання та інтелектуальний аналіз даних : навч. посіб. Вінниця : ВНТУ, 2024. 263 с. URL: https://pdf.lib.vntu.edu.ua/books/2024/Mokin_2024_263.pdf (дата звернення: 20.05.2026).

5. Обертюх Р. Р., Слабкий А. В. Математичне моделювання механічних систем : навч. посіб. 2-ге вид., перероб. та доп. Вінниця : ВНТУ, 2025. 119 с. URL: https://ir.lib.vntu.edu.ua/handle/123456789/44501 (дата звернення: 20.05.2026).

6. Сакович Л. М., Рижов Є. В., Ходич О. В., Наконечний В. С., Міночкін А. І. Технічна експлуатація засобів та систем зв’язку : навч. посіб. Київ : КПІ ім. Ігоря Сікорського, 2021. 176 с. URL: https://ela.kpi.ua/handle/123456789/57827 (дата звернення: 20.05.2026).

7. Савченко І. О. Моделювання складних систем. Комп’ютерний практикум : навч. посіб. Київ : КПІ ім. Ігоря Сікорського, 2025. 37 с. URL: https://ela.kpi.ua/bitstreams/824d9a4d-5236-4d21-a1dd-57822516924b/download (дата звернення: 20.05.2026).

8. Ситник Г. П., Комаха Л. Г., Рудик А. О. Основи теорії систем та системного аналізу : навч. посіб. / за заг. ред. Г. П. Ситника. Київ : Академпрес, 2024. 160 с. ISBN 978-966-7541-60-6. URL: https://ipacs.knu.ua/pages/dop/391/files/16db659a-6764-4c03-86b2-f517aec290b6.pdf (дата звернення: 20.05.2026).

9. Ahmed Y. S., Habaebi M. H., Islam M. R. Advances in Fault Detection Techniques for Automated Manufacturing Systems. Frontiers in Mechanical Engineering. 2025. DOI: https://doi.org/10.3389/fmech.2025.1564846.

10. Bishop C. M., Bishop H. Deep Learning: Foundations and Concepts. Cham : Springer, 2024. 607 p. DOI: https://doi.org/10.1007/978-3-031-45468-4.

11. Breiman L. Random Forests. Machine Learning. 2001. Vol. 45. P. 5–32. DOI: https://doi.org/10.1023/A:1010933404324.

12. Cação J., Santos J. S., Antunes M. Explainable AI for industrial fault diagnosis: A systematic review. Journal of Industrial Information Integration. 2025. Vol. 47. Article 100905. DOI: https://doi.org/10.1016/j.jii.2025.100905.

13. Chen X., Yang R., Xue Y., Huang M., Ferrero R., Wang Z. Deep Transfer Learning for Bearing Fault Diagnosis: A Systematic Review Since 2016. IEEE Transactions on Instrumentation and Measurement. 2023. Vol. 72. P. 1–21. DOI: https://doi.org/10.1109/TIM.2023.3244237.

14. Fang Z., Wang X., Liu Y., Zhang L. Prognostics and Health Management Based on Next-Generation Artificial Intelligence: A Review. Applied Sciences. 2024. Vol. 14, No. 14. Article 6120. DOI: https://doi.org/10.3390/app14146120.

15. Fernandes M., Corchado J. M., Marreiros G. Machine learning techniques applied to mechanical fault diagnosis and fault prognosis in the context of real industrial manufacturing use-cases: a systematic literature review. Applied Intelligence. 2022. Vol. 52. P. 14246–14280. DOI: https://doi.org/10.1007/s10489-022-03344-3.

16. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. 3rd ed. Sebastopol : O'Reilly Media, 2022. 864 p. URL:

https://www.oreilly.com/library/view/hands-on-machine-learning/9781098125967/ (дата звернення: 20.05.2026).

17. Hakim M. L., Awad A., Ihsan S. I. A Systematic Review of Rolling Bearing Fault Diagnoses Based on Deep Learning and Transfer Learning: Taxonomy, Overview, Application, Open Challenges, Weaknesses and Recommendations. Ain Shams Engineering Journal. 2023. Vol. 14, No. 5. Article 101945. DOI: https://doi.org/10.1016/j.asej.2022.101945.

18. IEC 60812:2018. Failure modes and effects analysis – FMEA and FMECA. Geneva : International Electrotechnical Commission, 2018. URL: https://webstore.iec.ch/publication/26359 (дата звернення: 20.05.2026).

19. 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. URL: https://www.iso.org/standard/21832.html (дата звернення: 20.05.2026).

20. ISO 13379-2:2015. Condition monitoring and diagnostics of machines – Data interpretation and diagnostics techniques – Part 2: Data-driven applications. Geneva : International Organization for Standardization, 2015. URL: https://www.iso.org/standard/40233.html (дата звернення: 20.05.2026).

21. James G., Witten D., Hastie T., Tibshirani R., Taylor J. An Introduction to Statistical Learning: With Applications in Python. Cham : Springer, 2023. 607 p. DOI: https://doi.org/10.1007/978-3-031-38747-0.

22. Leite D., Martins A., Rativa D., De Oliveira J. F. L., Maciel A. M. A. An Automated Machine Learning Approach for Real-Time Fault Detection and Diagnosis. Sensors. 2022. Vol. 22, No. 16. Article 6138. DOI: https://doi.org/10.3390/s22166138.

23. Leite D., Andrade E., Rativa D., Maciel A. M. A. Fault Detection and Diagnosis in Industry 4.0: A Review on Challenges and Opportunities. Sensors. 2025. Vol. 25, No. 1. Article 60. DOI: https://doi.org/10.3390/s25010060.

24. Lundberg S. M., Lee S.-I. A Unified Approach to Interpreting Model Predictions. Advances in Neural Information Processing Systems. 2017. Vol. 30. P. 4765–4774. URL:

https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html (дата звернення: 20.05.2026).

25. Murphy K. P. Probabilistic Machine Learning: An Introduction. Cambridge : MIT Press, 2022. 864 p. URL: https://probml.github.io/pml-book/book1.html (дата звернення: 20.05.2026).

26. National Institute of Standards and Technology. Prognostics and Health Management for Reliable Operations in Smart Manufacturing [Electronic resource]. URL: https://www.nist.gov/programs-projects/prognostics-and-health-management-reliable-operations-smart-manufacturing-phm4sm (дата звернення: 20.05.2026).

27. Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O. et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 2011. Vol. 12. P. 2825–2830. URL: https://www.jmlr.org/papers/v12/pedregosa11a.html (дата звернення: 20.05.2026).

28. Polverino L., Abbate S., Guizzi G., Vespoli S. Machine Learning for Prognostics and Health Management of Industrial Mechanical Systems and Equipment: A Systematic Literature Review. Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability. 2023. DOI: https://doi.org/10.1177/18479790231186848.

29. Wang C., Chen X., Qiang X., Fan H., Li S. Recent Advances in Mechanism/Data-Driven Fault Diagnosis of Complex Engineering Systems with Uncertainties. AIMS Mathematics. 2024. Vol. 9, No. 11. P. 29736–29772. DOI: https://doi.org/10.3934/math.20241441.

30. Webert H., Döß T., Huber M. F., Pauli J. Fault Handling in Industry 4.0: Definition, Process and Applications. Sensors. 2022. Vol. 22, No. 6. Article 2205. DOI: https://doi.org/10.3390/s22062205.

ДОДАТКИ

Додаток А

Фрагмент експериментального набору діагностичних даних

	A	B	C	D	E	F	G	H	I	J
1	temperature	vibration	voltage	current	load	errors	response_time	operating_hours	state	
2	37.32	0.35	220.26	Кві.85	46.67	0	71.22	2405.02	0	
3	62.15	0.59	208.88	Лип.18	85.04	6	166.92	7733.88	2	
4	43.8	0.37	217.07	Кві.99	48.65	1	79.26	1915.6	0	
5	41.57	0.3	221.41	Кві.84	22.21	1	76.44	1514.87	0	
6	66.67	0.68	220.77	05.Лют	58.9	2	115.83	6181.71	1	
7	43.15	0.29	219.81	05.Гру	51.56	2	79.94	1991.24	0	
8	45.37	0.42	220.3	Тра.33	49.56	2	47.25	2574.81	0	
9	58.91	0.33	217.53	Тра.63	59.59	1	93.09	4212.59	1	
10	48.73	0.39	221.62	05.Тра	40.98	1	79.61	3175.53	0	
11	49.57	0.38	221.18	Кві.46	40.43	0	68.69	1695.1	0	
12	44.7	0.4	218.19	Тра.41	45.22	0	69.83	2596.34	0	
13	70.25	0.79	214.54	Чер.47	57.47	8	142.5	7540.25	2	
14	49.68	0.66	213.75	06.Лип	73.35	3	61.42	3276.59	1	
15	42.59	0.25	219.17	Кві.32	42.54	0	49.61	3738.28	0	
16	48.82	0.33	219.19	Бер.81	50.82	3	91.52	2504.94	0	
17	44.11	0.31	224.78	Кві.58	42.1	2	55.37	2975.26	0	
18	48.66	0.45	215.12	Кві.59	39.59	0	56.66	3087.95	0	
19	47.59	0.4	213.8	Кві.48	41.08	0	91.74	2351.14	0	

Додаток Б

Лістинг програмного коду

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score, precision_score, recall_score,
f1_score
from sklearn.metrics import classification_report, confusion_matrix,
ConfusionMatrixDisplay

np.random.seed(42)

n_normal = 220
n_deviation = 130
n_warning = 110
n_critical = 80
n_fault = 60

normal = pd.DataFrame({
    "temperature": np.random.normal(45, 4, n_normal),
    "vibration": np.random.normal(0.35, 0.08, n_normal),
    "voltage": np.random.normal(220, 3, n_normal),
    "current": np.random.normal(4.5, 0.5, n_normal),
    "load": np.random.normal(45, 8, n_normal),
    "errors": np.random.poisson(1, n_normal),
    "response_time": np.random.normal(70, 12, n_normal),
    "operating_hours": np.random.normal(2500, 900, n_normal),
    "state": 0
})

deviation = pd.DataFrame({
    "temperature": np.random.normal(55, 5, n_deviation),
    "vibration": np.random.normal(0.55, 0.12, n_deviation),
    "voltage": np.random.normal(216, 4, n_deviation),
    "current": np.random.normal(5.5, 0.7, n_deviation),
    "load": np.random.normal(58, 9, n_deviation),
    "errors": np.random.poisson(3, n_deviation),
    "response_time": np.random.normal(95, 18, n_deviation),
    "operating_hours": np.random.normal(4500, 1200, n_deviation),
    "state": 1
})

warning = pd.DataFrame({
    "temperature": np.random.normal(66, 6, n_warning),
    "vibration": np.random.normal(0.85, 0.18, n_warning),
    "voltage": np.random.normal(212, 5, n_warning),
    "current": np.random.normal(6.8, 0.9, n_warning),
    "load": np.random.normal(70, 10, n_warning),
    "errors": np.random.poisson(6, n_warning),
    "response_time": np.random.normal(140, 25, n_warning),
    "operating_hours": np.random.normal(7000, 1500, n_warning),
    "state": 2
```

```
})

critical = pd.DataFrame({
    "temperature": np.random.normal(78, 7, n_critical),
    "vibration": np.random.normal(1.25, 0.25, n_critical),
    "voltage": np.random.normal(205, 6, n_critical),
    "current": np.random.normal(8.5, 1.0, n_critical),
    "load": np.random.normal(85, 8, n_critical),
    "errors": np.random.poisson(12, n_critical),
    "response_time": np.random.normal(220, 35, n_critical),
    "operating_hours": np.random.normal(10000, 1800, n_critical),
    "state": 3
})

fault = pd.DataFrame({
    "temperature": np.random.normal(90, 6, n_fault),
    "vibration": np.random.normal(1.75, 0.3, n_fault),
    "voltage": np.random.normal(198, 7, n_fault),
    "current": np.random.normal(10.2, 1.1, n_fault),
    "load": np.random.normal(92, 6, n_fault),
    "errors": np.random.poisson(18, n_fault),
    "response_time": np.random.normal(310, 45, n_fault),
    "operating_hours": np.random.normal(12500, 1600, n_fault),
    "state": 4
})

data = pd.concat([normal, deviation, warning, critical, fault], ignore_index=True)

for col in [
    "temperature",
    "vibration",
    "voltage",
    "current",
    "load",
    "response_time",
    "operating_hours"
]:
    data[col] = data[col].round(2)

data["errors"] = data["errors"].astype(int)
data = data.sample(frac=1, random_state=42).reset_index(drop=True)

print("Фрагмент експериментального набору даних:")
print(data.head(10))

print("\nРозмір набору даних:")
print(data.shape)

print("\nТипи даних:")
print(data.dtypes)

print("\nКількість пропущених значень:")
print(data.isnull().sum())

state_names = {
    0: "Нормальний стан",
    1: "Допустиме відхилення",
    2: "Попередження",
    3: "Критичний стан",
    4: "Несправність"
}
```

```
state_counts = data["state"].value_counts().sort_index()
state_counts.index = [state_names[i] for i in state_counts.index]

print("\nРозподіл технічних станів:")
print(state_counts)

plt.figure(figsize=(9, 5))
plt.bar(state_counts.index, state_counts.values)
plt.title("Розподіл технічних станів у наборі даних")
plt.xlabel("Технічний стан")
plt.ylabel("Кількість записів")
plt.xticks(rotation=20)
plt.grid(axis="y", alpha=0.3)
plt.show()

X = data.drop("state", axis=1)
y = data["state"]

X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.2,
    random_state=42,
    stratify=y
)

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

print("\nКількість записів у навчальній вибірці:", X_train.shape[0])
print("Кількість записів у тестовій вибірці:", X_test.shape[0])
print("Кількість діагностичних ознак:", X_train.shape[1])

models = {
    "Logistic Regression": LogisticRegression(max_iter=1000, random_state=42),
    "kNN": KNeighborsClassifier(n_neighbors=5),
    "Decision Tree": DecisionTreeClassifier(max_depth=5, random_state=42),
    "Random Forest": RandomForestClassifier(n_estimators=100, random_state=42)
}

trained_models = {}

for name, model in models.items():
    if name in ["Logistic Regression", "kNN"]:
        model.fit(X_train_scaled, y_train)
    else:
        model.fit(X_train, y_train)

    trained_models[name] = model

print("\nНавчання моделей завершено.")

for name, model in trained_models.items():
    if name in ["Logistic Regression", "kNN"]:
        y_pred = model.predict(X_test_scaled)
    else:
        y_pred = model.predict(X_test)

    results.append({
        "Модель": name,
        "Accuracy": accuracy_score(y_test, y_pred),
```

```
        "Precision": precision_score(y_test, y_pred, average="weighted"),
        "Recall": recall_score(y_test, y_pred, average="weighted"),
        "F1-score": f1_score(y_test, y_pred, average="weighted")
    })

results_df = pd.DataFrame(results)
results_df = results_df.sort_values(by="Accuracy", ascending=False)

print("\nПорівняння результатів роботи моделей:")
print(results_df)

plt.figure(figsize=(8, 5))
plt.bar(results_df["Модель"], results_df["Accuracy"])
plt.title("Порівняння точності моделей діагностики")
plt.xlabel("Модель")
plt.ylabel("Accuracy")
plt.ylim(0, 1)
plt.grid(axis="y", alpha=0.3)
plt.xticks(rotation=20)
plt.show()

rf_model = trained_models["Random Forest"]
rf_pred = rf_model.predict(X_test)

cm = confusion_matrix(y_test, rf_pred)

display_labels = [
    "Норма",
    "Відхилення",
    "Попередження",
    "Критичний",
    "Несправність"
]

disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=display_labels)

plt.figure(figsize=(8, 6))
disp.plot(values_format="d", cmap=None)
plt.title("Матриця помилок моделі Random Forest")
plt.xticks(rotation=30)
plt.show()

print("\nЗвіт класифікації для моделі Random Forest:")
print(classification_report(
    y_test,
    rf_pred,
    target_names=[
        "Нормальний стан",
        "Допустиме відхилення",
        "Попереджувальний стан",
        "Критичний стан",
        "Несправність"
    ]
))

feature_importance = pd.DataFrame({
    "Ознака": X.columns,
    "Важливість": rf_model.feature_importances_
}).sort_values(by="Важливість", ascending=False)

print("\nВажливість діагностичних ознак:")
print(feature_importance)
```

```
plt.figure(figsize=(9, 5))
plt.bar(feature_importance["Ознака"], feature_importance["Важливість"])
plt.title("Важливість діагностичних ознак для моделі Random Forest")
plt.xlabel("Діагностична ознака")
plt.ylabel("Важливість")
plt.xticks(rotation=30)
plt.grid(axis="y", alpha=0.3)
plt.show()

new_object = pd.DataFrame([{
    "temperature": 82.5,
    "vibration": 1.42,
    "voltage": 203.0,
    "current": 8.9,
    "load": 88.0,
    "errors": 14,
    "response_time": 245.0,
    "operating_hours": 10800.0
}])

prediction = rf_model.predict(new_object)[0]
prediction_name = state_names[prediction]

print("\nВхідні параметри технічної системи:")
print(new_object)

print("\nПрогнозований технічний стан:")
print(prediction, "-", prediction_name)

best_model_name = results_df.iloc[0]["Модель"]
best_accuracy = results_df.iloc[0]["Accuracy"]

print("\nНайкраща модель:", best_model_name)
print("Accuracy:", round(best_accuracy, 3))

rf_accuracy = accuracy_score(y_test, rf_pred)
rf_precision = precision_score(y_test, rf_pred, average="weighted")
rf_recall = recall_score(y_test, rf_pred, average="weighted")
rf_f1 = f1_score(y_test, rf_pred, average="weighted")

effectiveness = pd.DataFrame({
    "Критерій": [
        "Найкраща загальна точність",
        "Точність Random Forest",
        "Precision Random Forest",
        "Recall Random Forest",
        "F1-score Random Forest",
        "Пояснюваність"
    ],
    "Результат": [
        f"{best_model_name}: {round(best_accuracy, 3)}",
        round(rf_accuracy, 3),
        round(rf_precision, 3),
        round(rf_recall, 3),
        round(rf_f1, 3),
        "визначено важливість діагностичних ознак"
    ],
    "Оцінка": [
        "висока",
        "висока",
        "висока",
```

```
        "висока",
        "висока",
        "достатня"
    ]
})

print("\nОцінювання ефективності запропонованого підходу:")
print(effectiveness)

criteria = [
    "Точність\nкласифікації",
    "Виявлення\nкритичних станів",
    "Стійкість\nдо шуму",
    "Пояснюваність\nрезультату",
    "Практична\nпридатність"
]

plt.figure(figsize=(12, 4))

for i, item in enumerate(criteria):
    plt.text(
        i,
        0,
        item,
        ha="center",
        va="center",
        bbox=dict(boxstyle="round,pad=0.5", edgecolor="black", facecolor="white")
    )
    plt.arrow(
        i,
        -0.15,
        0,
        -0.35,
        head_width=0.05,
        head_length=0.08,
        length_includes_head=True,
        color="black"
    )
    plt.text(
        i,
        -0.65,
        "Оцінка\nалгоритму",
        ha="center",
        va="center",
        bbox=dict(boxstyle="round,pad=0.4", edgecolor="black", facecolor="white")
    )

plt.xlim(-0.8, len(criteria) - 0.2)
plt.ylim(-1.0, 0.5)
plt.axis("off")
plt.title("Оцінювання ефективності алгоритму інтелектуальної діагностики")
plt.show()

monitoring_steps = [
    "Датчики\nтехнічної системи",
    "Збір\nданих",
    "Модуль\nобробки",
    "Інтелектуальна\nмодель",
    "Клас\nстану",
    "Повідомлення\nоператору"
]
```

```
    plt.figure(figsize=(13, 3))

    for i, step in enumerate(monitoring_steps):
        plt.text(
            i,
            0,
            step,
            ha="center",
            va="center",
            bbox=dict(boxstyle="round,pad=0.5", edgecolor="black", facecolor="white")
        )
        if i < len(monitoring_steps) - 1:
            plt.arrow(
                i + 0.35,
                0,
                0.3,
                0,
                head_width=0.05,
                head_length=0.08,
                length_includes_head=True,
                color="black"
            )

    plt.xlim(-0.5, len(monitoring_steps) - 0.5)
    plt.ylim(-0.5, 0.5)
    plt.axis("off")
    plt.title("Використання алгоритму діагностики в системі технічного моніторингу")
    plt.show()

    data.to_csv("diagnostic_dataset.csv", index=False)

    print("\nФайл diagnostic_dataset.csv збережено.")

    print("\nПідсумковий висновок:")
    print(f"За результатами експериментального дослідження найкращі результати
показала модель {best_model_name}.")
    print("Вона забезпечила найвищу точність класифікації серед досліджених моделей.")
    print("Модель Random Forest також показала високий результат і була використана
для аналізу важливості діагностичних ознак.")
```

ДЕМОНСТРАЦІЙНИЙ МАТЕРІАЛ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

«Інтелектуальні методи та алгоритми діагностики складних технічних систем»

на здобуття освітнього ступеня бакалавра
зі спеціальності 124 – Системний аналіз
освітньо-професійної програми Системний аналіз

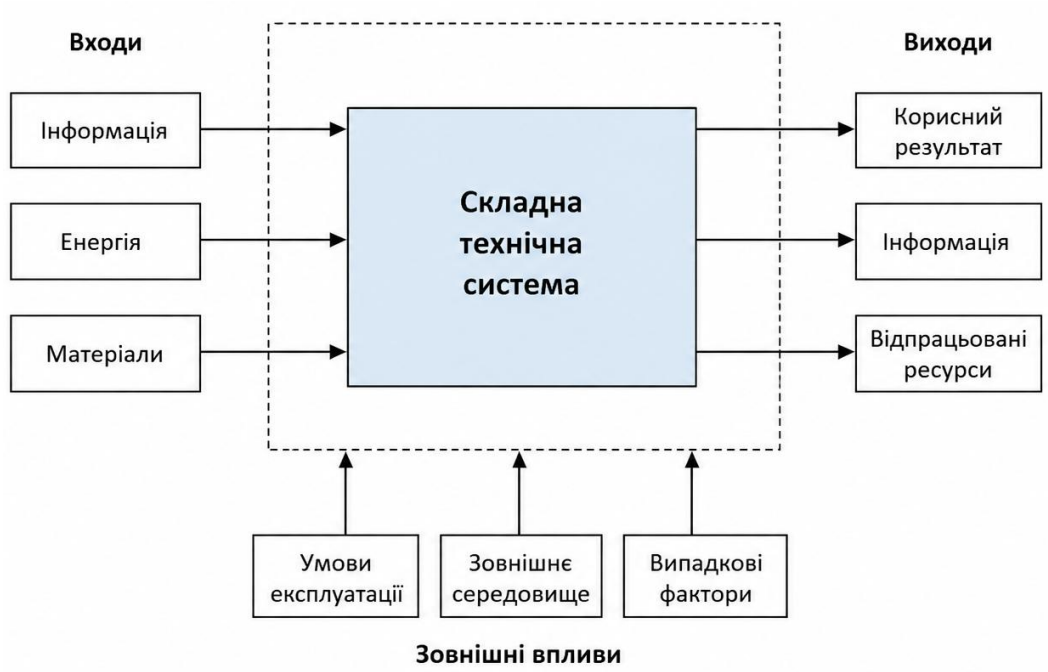
Виконав: здобувач вищої освіти гр. САД-41
Ярослав ЛЕЩЕНКО

Керівник: Ровіл НАФЄЄВ
кандидат фізико-математичних наук, доцент

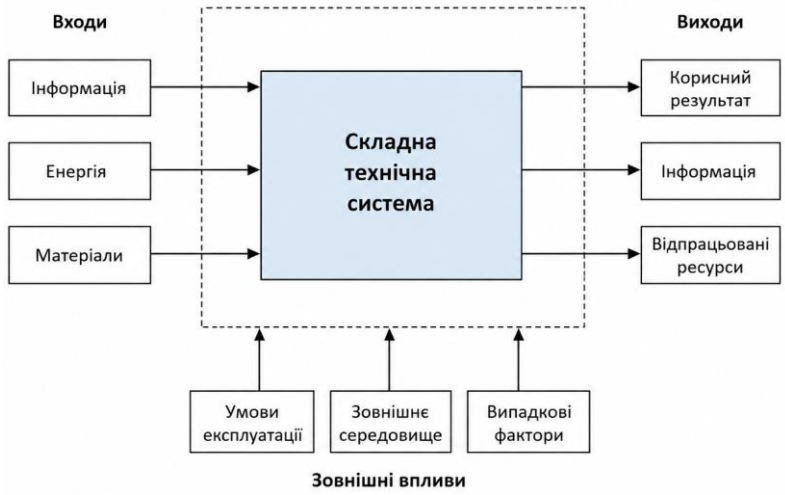
Київ - 2026

Інтелектуальні методи та алгоритми діагностики складних технічних систем

Захист: 10–15 хвилин. Акцент — на схемах, алгоритмі та результатах експерименту.



Складна технічна система як об’єкт діагностики



Входи: інформація, енергія, матеріали

Виходи: результат, інформація, відпрацьовані ресурси

Зовнішні впливи: умови експлуатації, середовище, випадкові фактори

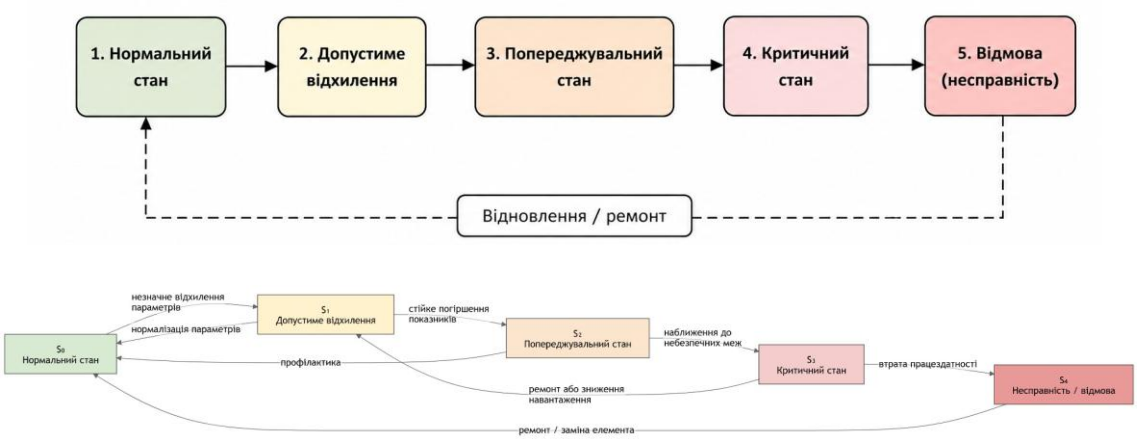
Для діагностики важливо оцінювати систему як єдиний об’єкт, а не як набір окремих деталей.

Що ускладнює діагностику

- частина параметрів може бути недоступною
- датчики мають похибки та шум
- режими роботи змінюються під час експлуатації
- одна ознака може відповідати кільком причинам

Тому діагностичне рішення має спиратися не на один показник, а на сукупність діагностичних ознак.

Технічний стан подано як послідовність рівнів



Алгоритм не просто фіксує відмову, а розрізняє: норму, допустиме відхилення, попередження, критичний стан і несправність.

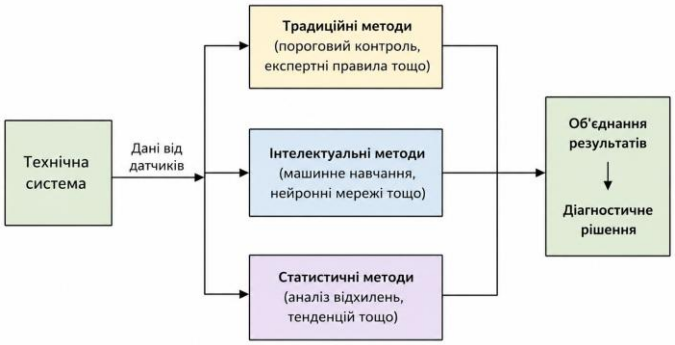
Підходи до діагностики

традиційні методи дають базовий контроль

інтелектуальні методи аналізують складні залежності

гібридний підхід підвищує надійність рішення

Обрана логіка: поєднати контроль і машинне навчання



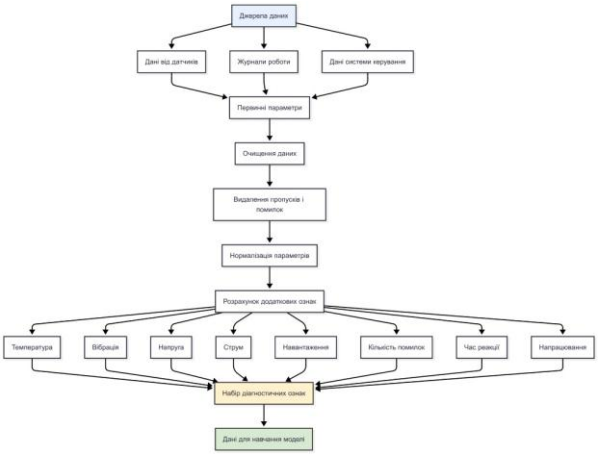
Традиційні методи
пороговий контроль, експертні правила, статистика

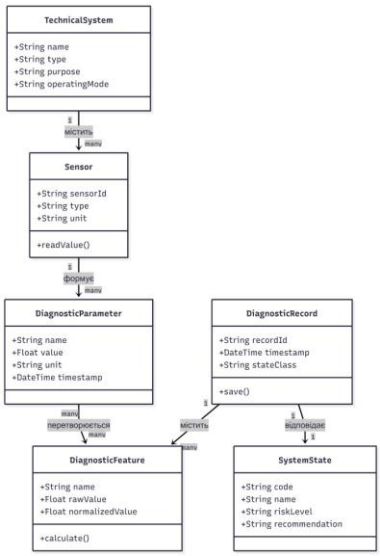
Інтелектуальні методи
Decision Tree, Random Forest, kNN, Logistic Regression

Рішення
клас технічного стану та рекомендація для оператора

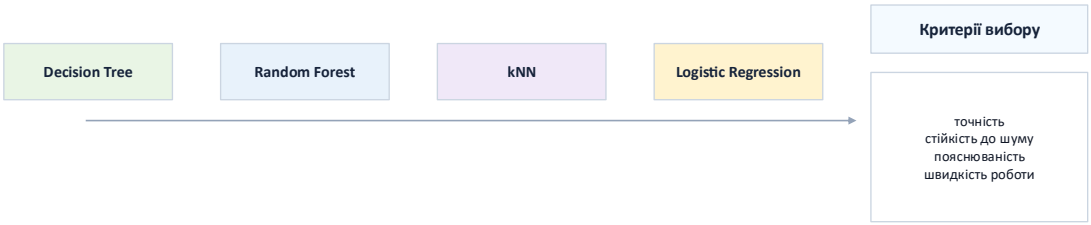
Загальна схема алгоритму

З чого формується набір ознак





Як обиралась модель машинного навчання



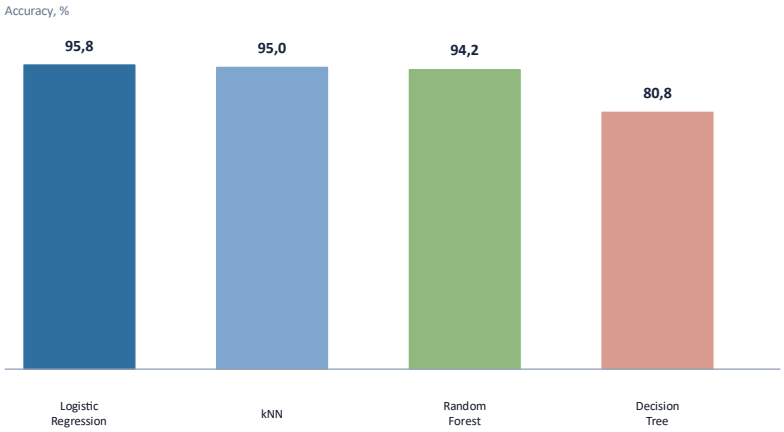
У роботі моделі порівнювалися не лише за accuracy. Для діагностики важливо, щоб алгоритм не пропускав критичні стани та міг бути поясненим інженеру.

Навчання, перевірка і формування рішення

Після навчання модель перевіряється за метриками. Якщо якість достатня, її можна використовувати для діагностики.

результат — клас стану + рекомендація

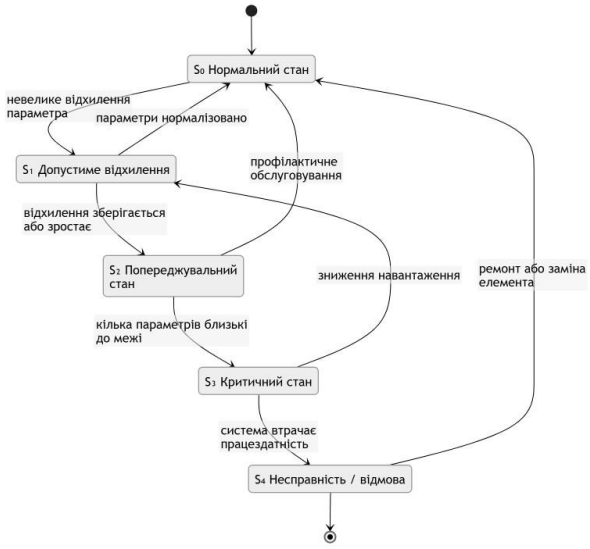
Що показав експеримент

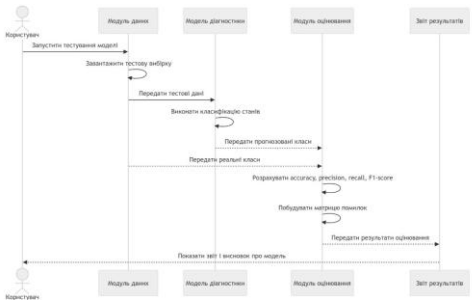


Модель	Acc.	Prec.	Recall	F1
Logistic Regression	0,958	0,959	0,958	0,958
kNN	0,950	0,951	0,950	0,950
Random Forest	0,942	0,946	0,942	0,942
Decision Tree	0,808	0,815	0,808	0,807

Найвищу загальну точність показала Logistic Regression. Random Forest додатково корисний для аналізу важливості ознак.

Як система взаємодіє з користувачем





Користувач запускає тестування моделі, модулі передають дані, модель формує класи, а модуль оцінювання рахує метрики.

Підсумок захисту

1
проаналізовано підходи

2
сформовано модель станів

3
реалізовано алгоритм

4
порівняно ML-моделі

Науково-технічна новизна
удосконалено підхід за рахунок поєднання системного аналізу, моделі станів і класифікації

Практичне значення
підхід може бути основою програмного засобу підтримки технічної діагностики

висновок: інтелектуальна діагностика підвищує обґрунтованість рішення про стан системи