

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра інженерії програмного забезпечення

Пояснювальна записка

до магістерської роботи
на ступінь вищої освіти магістр

на тему: «**ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ПРОМИСЛОВИХ
ПІДПРИЄМСТВ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ**»

Виконав: студент 6 курсу, групи ПДМ–61
спеціальності

121 Інженерія програмного забезпечення
(шифр і назва спеціальності/спеціалізації)

Кривов'яз Ю. О.

(прізвище та ініціали)

Керівник Щербина І. С.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Київ – 2022

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти – «Магістр»

Спеціальність підготовки – 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

Негоденко О.В.

“ _____ ” _____ 2022 року

З А В Д А Н Н Я НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТА

КРИВОВ'ЯЗУ ЮРІО ОЛЕКСАНДРОВИЧУ

(прізвище, ім'я, по батькові)

1. Тема роботи: «Підвищення ефективності промислових підприємств на основі нейронних мереж»

Керівник роботи: Щербина І. С., к.т.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом закладу вищої освіти «11» жовтня 2021 року №170.

2. Строк подання студентом роботи _____

3. Вхідні дані до роботи

Показники ефективності роботи обладнання промислового підприємства

Методи підвищення ефективності промислових підприємств

Технології нейронних мереж та методи машинного навчання

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

4.1 Наукова проблематика. Завдання прогнозного технічного обслуговування.

4.2 Методи машинного навчання для прогнозування несправностей обладнання.

4.3 Алгоритм прогнозного технічного обслуговування.

4.4 Застосування методів машинного навчання для прогнозування несправностей обладнання.

5. Перелік демонстраційного матеріалу

1. Мета, об'єкт та предмет дослідження

2. Класифікація стратегій технічного обслуговування
3. Вплив впровадження стратегії прогнозного технічного обслуговування
4. Метод машинного навчання: Випадковий ліс
5. Метод машинного навчання: Градієнтний бустинг
6. Нейронна мережа
7. Алгоритм прогнозного технічного обслуговування
8. Характеристика набору даних
9. Результати методів машинного навчання
10. Архітектура системи прогнозування несправностей для прогнозного обслуговування
11. Апробація роботи
12. Висновки

6. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури		Виконано
2	Дослідження методів машинного навчання		Виконано
3	Розробка алгоритму прогнозного технічного обслуговування та архітектури програмного забезпечення		Виконано
4	Створення та навчання моделі для прогнозування несправностей промислового устаткування		Виконано
5	Оформлення розділів дипломної роботи		Виконано
6	Вступ, висновки, реферат		Виконано
7	Розробка обов'язкових демонстраційних матеріалів		Виконано
8	Перевірка на антиплагіат		Виконано
9	Попередній захист роботи		Виконано
10	Проходження нормконтролю		Виконано
11	Здача роботи		

Студент _____
(підпис)

Кривов'яз Ю. О.
(прізвище та ініціали)

Керівник роботи _____
(підпис)

Щербина І. С.
(прізвище та ініціали)

РЕФЕРАТ

Текстова частин магістерської роботи с. 97, рис. 53, 59 джерел.

ТЕХНІЧНЕ ОБСЛУГОВУВАННЯ, ПРОГНОЗНЕ ОБСЛУГОВУВАННЯ, ПРОГНОЗУВАННЯ НЕСПРАВНОСТЕЙ, МЕТОДИ МАШИННОГО НАВЧАННЯ, РОЗРОБКА ОЗНАК, ВИПАДКОВИЙ ЛІС, НЕЙРОННІ МЕРЕЖІ, ГРАДІЄНТНИЙ БУСТИНГ, МАТРИЦЯ НЕВІДПОВІДНОСТЕЙ, ХМАРНІ ОБЧИСЛЕННЯ, AMAZON WEB SERVICES

Об'єкт дослідження – устаткування промислових виробництв.

Предмет дослідження – прогнозування несправностей промислового устаткування на основі методів машинного навчання.

Мета роботи – підвищення ефективності промислових виробництв за рахунок впровадження прогнозного технічного обслуговування на основі методів машинного навчання.

Методи дослідження – системний аналіз, математичне моделювання.

У роботі проведено аналіз існуючих методів технічного обслуговування для промислового устаткування, таких як реактивне обслуговування, профілактичне обслуговування та прогнозне обслуговування. Було встановлено, що головним недоліком реактивного обслуговування є ремонт промислового устаткування після його поломки, що може приносити збитки виробництву. Профілактичне обслуговування носить строго визначений в часі розпорядок, що також не є гарантом постійної працездатності обладнання, за рахунок недетермінованих поломок. Реактивне та профілактичне обслуговування зазвичай використовуються разом. Прогнозне обслуговування є найкращим типом обслуговування оскільки ресурси виробництва на технічне обслуговування витрачаються тільки на обслуговування прогнозованих поломок, що дозволяє використовувати устаткування більш ефективно та безпечно.

Дослідження методів машинного навчання для прогнозування несправностей показало, що для набору даних, що був використаний в роботі, найкраще підходять наступні методи: випадковий ліс, нейронна мережа, градієнтний бустинг.

Підготовка даних та створення моделі відбувалось за методикою, що запропонована в роботі. Отриману модель необхідно впроваджувати у виробництво використовуючи хмарні обчислення. Архітектура системи для прогнозного обслуговування побудована на основі Amazon Web Services наведена і описана в роботі.

Галузь використання – технічне обслуговування промислового устаткування.

ЗМІСТ

ВСТУП.....	10
1 НАУКОВА ПРОБЛЕМАТИКА. ЗАВДАННЯ ПРОГНОЗНОГО ТЕХНІЧНОГО ОБСЛУГОВУВАННЯ.....	12
1.1 Постановка наукової задачі	12
1.2 Огляд стратегій технічного обслуговування.....	12
1.3 Прогнозне обслуговування та Industry 4.0.....	16
1.4 Обґрунтування доцільності дослідження технічного обслуговування для підвищення ефективності промисловості.....	19
1.5 Використання машинного навчання для прогнозного обслуговування	24
2 МЕТОДИ МАШИННОГО НАВЧАННЯ ДЛЯ ПРОГНОЗУВАННЯ НЕСПРАВНОСТЕЙ ОБЛАДНАННЯ.....	28
2.1 Випадковий ліс (Random Forest)	28
2.2 Підсилення градієнта (Gradient Boosting Tree)	32
2.3 Нейронні мережі (Neural Networks)	35
2.4 Вибір методів машинного навчання та дослідження метрик якості методів машинного навчання.....	52
3 АЛГОРИТМ ПРОГНОЗНОГО ТЕХНІЧНОГО ОБСЛУГОВУВАННЯ ...	55
3.1 Отримання даних (Data Extraction).....	55
3.2 Прогнозування даних (Data Prediction).....	57
4 ЗАСТОСУВАННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ПРОГНОЗУВАННЯ НЕСПРАВНОСТЕЙ ОБЛАДНАННЯ.....	60
4.1 Вибір технічних засобів.....	60
4.2 Характеристика набору даних	60
4.2.1 Вихідні дані обладнання.....	60
4.2.2 Журнал помилок обладнання.....	64
4.2.3 Історія технічного обслуговування	65
4.2.4 Історія поломок обладнання	66
4.2.5 Відомості про обладнання.....	67

4.3	Розробка ознак.....	68
4.4	Застосування методів машинного навчання.....	73
4.5	Архітектура програмного забезпечення для використання моделі машинного навчання для прогнозного обслуговування.....	77
	ВИСНОВКИ	82
	ПЕРЕЛІК ПОСИЛАНЬ.....	85
	ДОДАТОК А.....	90

ВСТУП

Обґрунтування вибору теми та її актуальність: методи технічного обслуговування, такі як, реактивне обслуговування та профілактичне обслуговування, найширше застосовуються в промислових виробництвах, але вони мають свої недоліки, які негативно впливають на прибутковість виробництва та термін експлуатації обладнання.

Для покращення ефективності промислових виробництв, сучасним варіантом вирішення проблем реактивного та профілактичного обслуговування є реалізація прогнозного обслуговування, що завдяки прогностичній моделі дозволяє ремонтувати обладнання перед його поломкою цим самим заощаджуючи кошти на обслуговуванні та заміні обладнання.

Ступінь вивчення проблеми: з появою технологій Індустрії 4.0, все більше промислових виробництв починають більш активно збирати та аналізувати дані, що дозволяє ефективніше налаштовувати процеси виробництва, що позитивно впливає на прибутковість виробництв. Прогнозне обслуговування є одним з основних проблем сучасних промислових виробництв, вирішення якого дозволить більш ефективно використовувати ресурси виробництв.

Основною проблемою прогнозного обслуговування є те, що неможливо розробити єдину систему, яка буде працювати для всього обладнання, тому системи розробляються в основному для кожного виробництва окремо, враховуючи наявне обладнання.

Для прогнозного обслуговування використовуються різні методи машинного навчання, що будуть розглянуті далі в роботі. В різних наукових роботах розглядаються різні галузі промисловості з використанням різних методів машинного навчання для прогнозування несправностей обладнання.

Об'єктом дослідження є устаткування промислових виробництв.

Предметом дослідження є прогнозування несправностей промислового устаткування на основі методів машинного навчання.

Метою роботи є підвищення ефективності промислових виробництв за рахунок впровадження прогнозного технічного обслуговування на основі методів машинного навчання.

Завданням роботи є розробка загальної методики, що дозволить реалізовувати прогнозне технічне обслуговування на основі методів машинного навчання та архітектури системи з використанням хмарних сервісів.

Методика дослідження: Першим кроком для реалізації прогнозування несправностей є розробка алгоритму, на основі якого буде здійснюватися реалізація моделі машинного навчання та подальше її використання. Виконавши всі кроки алгоритму ми отримали модель за допомогою якої ми можемо прогнозувати несправності обладнання. Наступним кроком є розробка архітектури програмного забезпечення з використанням хмарних сервісів для реалізації прогнозного обслуговування. Розроблена архітектура базується на використанні Amazon Web Services та дозволяє отримувати, зберігати та аналізувати дані, прогнозувати несправності обладнання та на основі прогнозування стану обладнання відображати поточний стан обладнання та нотифікувати відповідальний персонал.

Наукова новизна роботи: Таким чином, було реалізовано три методи машинного навчання, що дозволяють прогнозувати несправності для обладнання промислових виробництв на основі розробленої методики впровадження прогнозного технічного обслуговування, та розроблено архітектуру програмного забезпечення на основі Amazon Web Services, що дозволяють використовувати моделі машинного навчання для прогнозування несправностей з хмарними сервісами.

Практична значущість результатів: Дану методику можна застосовувати для різних галузей та обладнання, де потрібно реалізувати прогнозне обслуговування на основі прогнозування несправностей промислового устаткування. Розроблену архітектуру на основі хмарних сервісів можна застосувати для систем прогнозування несправностей в різних галузях промислових виробництв.

1 НАУКОВА ПРОБЛЕМАТИКА. ЗАВДАННЯ ПРОГНОЗНОГО ТЕХНІЧНОГО ОБСЛУГОВУВАННЯ

1.1 Постановка наукової задачі

Вічною виробничою ціллю було вироблення більшої кількості продукції вищої якості за мінімальних витрат.

Революція «розумного виробництва» вже дає змогу виробникам досягнути цієї мети успішніше, ніж будь-коли, і однією з основних технологій цієї нової хвилі інновацій є промисловий штучний інтелект.

Дані стали дуже цінним ресурсом, і їх дешевше, ніж будь-коли збирати та зберігати. Сьогодні все більше виробників використовують ці дані для значного покращення результату промислового виробництва завдяки штучному інтелекту, зокрема машинному навчанню.

Для багатьох це означає значне підвищення ефективності та потужності виробництва за рахунок усунення основних причин виробничих витрат та інших супутніх витрат.

Перед нами стоїть завдання дослідити методи машинного навчання, що можна використати для прогнозування несправностей машин, компонентів, обладнання. Дані прогнозування можна буде використати для підвищення ефективності технічного обслуговування, що в свою чергу позитивно впливатиме на ефективність промислового підприємства в цілому.

1.2 Огляд стратегій технічного обслуговування

Обслуговування може включати набір заходів з технічного обслуговування: моніторинг, аналіз стану, планове технічне обслуговування, капітальний ремонт, ремонт і відновлення. Відповідно до Європейського стандарту технічне обслуговування визначається як сукупність усіх технічних, адміністративних та управлінських дій протягом життєвого циклу об'єкта, призначених для збереження

або відновлення робочого стану, в якому він може виконувати необхідні функції [1].

На рисунку 1.1 можна побачити, що розрізняється три стратегії технічного обслуговування: коригувальне обслуговування (Corrective Maintenance), профілактичне обслуговування (Preventive Maintenance) і прогнозне обслуговування (Predictive Maintenance).

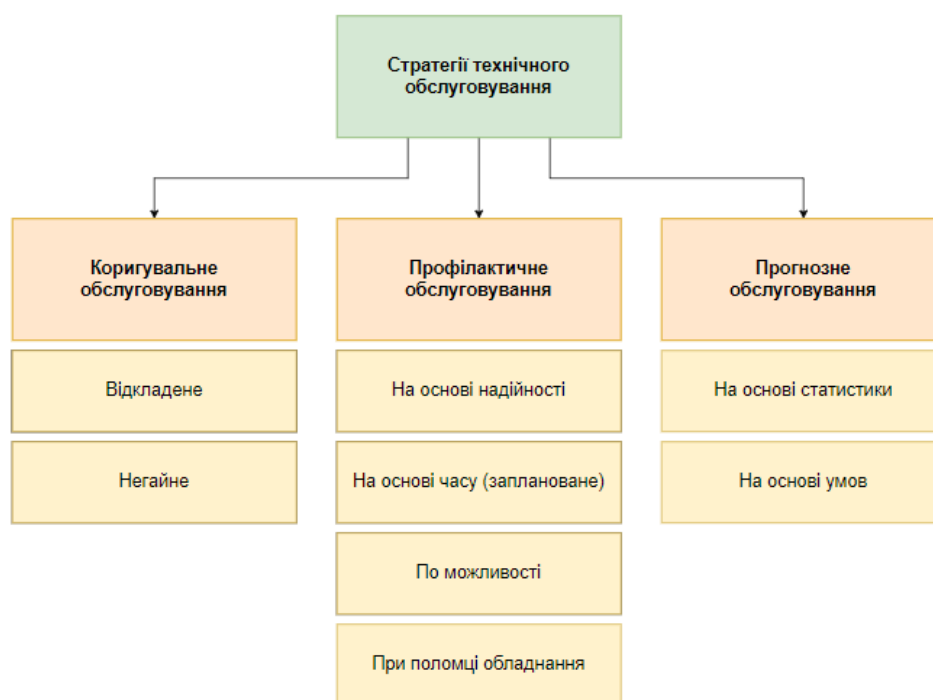


Рисунок 1.1 – Класифікація стратегій технічного обслуговування

Коригувальне обслуговування подібне до ремонтних робіт, які виконуються, коли обладнання має явну помилку або поломку. Також коригувальне обслуговування називають обслуговування від роботи до відмови. Профілактичне обслуговування проводиться через заздалегідь визначені проміжки часу або відповідно до попередньо визначених критеріїв з метою зниження ймовірності збою або погіршення певних функцій. Профілактичне обслуговування планується без будь-яких засобів моніторингу. Розклад може бути заснований на кількості робочих годин або часу. Ці дві стратегії широко застосовуються в багатьох галузях промисловості та часто поєднуються для досягнення кращої ефективності промислових виробництв [2].

Будучи однією з найпопулярніших і сучасних стратегій технічного обслуговування, прогнозоване обслуговування вимірює параметри стану обладнання і виконує відповідні завдання для оптимізації терміну служби машин і процесів без збільшення ризику відмови [3]. Розрізняють дві групи прогнозного обслуговування: на основі статистичних даних та на основі умов.

У порівнянні з іншими стратегіями обслуговування, прогнозне обслуговування має деякі переваги: обладнання, яке потребує обслуговування відключається лише перед неминучою поломкою (це дозволяє максимально уникати простоїв виробництва); скорочення загального часу, витраченого на обслуговування обладнання; зниження витрат на технічне обслуговування за рахунок уникнення катастрофічних поломок обладнання; підвищення доступності та надійності обладнання; продовження терміну служби обладнання та процесів. Проте прогнозне обслуговування стикається з кількома проблемами:

Високі вимоги до доступу до даних, якості даних, об'єднання даних з різних джерел. Оскільки ці джерела даних часто працюють у неоднорідному середовищі, інтеграція між системами є проблематичною [4].

Можливість працювати з промисловими великими даними. Щоб використовувати великі дані, промисловим підприємствам потрібно мати здатність підтримувати різні типи інформації, інфраструктура для зберігання масивних наборів даних і гнучкість, щоб використовувати інформацію після її збору та збереження, що дозволяє історичний аналіз критичних тенденцій для прогнозування в реальному часі [5].

Точність прогнозу. Неточна прогнозна інформація може призвести до непотрібного технічного обслуговування, наприклад, до ранньої заміни компонентів, або до простою виробництва через несподівані збої машини. Тому точність прогнозування залишкового терміну використання обладнання, зокрема довгострокового прогнозу, що дає достатньо часу для підготовки до процесу технічного обслуговування, відіграє істотну роль у повній реалізації потенціалу прогнозного технічного обслуговування [6].

На основі проведеного аналізу можна визначити переваги та недоліки кожної стратегії технічного обслуговування.

Оскільки коригувальне характеризується технічним обслуговуванням після збою, він ідеально підходить для обладнання з низьким пріоритетом, тобто устаткування, яке не впливає на функціонування виробництва. Те ж саме стосується обладнання меншої вартості, оскільки робота, необхідна для обслуговування може виявитись дорожчою, ніж ремонт або його заміна, коли воно виходить з ладу. Коригувальне обслуговування має наступні недоліки: основна проблема полягає в застосуванні коригувального обслуговування до устаткування середнього та високого пріоритету, оскільки в стратегії коригувального обслуговування не вживаються запобіжні дії, термін служби обладнання буде меншим, ніж при застосуванні однієї з альтернативних стратегій технічного обслуговування; якщо даний підхід застосувати до устаткування високого пріоритету, то це може призвести до несподіваних зупинок та великих витрат на ремонт обладнання.

Профілактичне обслуговування є життєво необхідним для устаткування високого пріоритету, яке є необхідним для нормального функціонування промислового виробництва. Профілактичне обслуговування дозволяє збільшити термін експлуатації обладнання та скоротити час незапланованих простоїв. Оскільки профілактичне обслуговування не ґрунтується на фактичному стані обладнання, плани обслуговування іноді можуть бути неефективними та призвести до непотрібного технічного обслуговування (включаючи заміну деталей), що вимагає часу та грошей. Коли профілактичне обслуговування застосовується до недорогих деталей або деталей з низьким пріоритетом, технічне обслуговування може виявитись значно дорожчим ніж коригувальний підхід.

Оскільки мета прогнозного обслуговування полягає в прогнозуванні потенційних несправностей, то даний підхід технічного обслуговування завжди буде необхідним і своєчасним, оскільки обладнання підлягатиме обслуговуванню лише тоді, коли прогнозується несправність, що зменшує витрати та робочий час, витрачений на технічне обслуговування. Недоліком даного підходу є необхідність

інвестування в спеціальне обладнання для моніторингу, а також навчання персоналу, щоб правильно збирати та аналізувати отримані дані.

1.3 Прогнозне обслуговування та Industry 4.0

Індустрія 4.0 – це новий спосіб виробництва товарів та послуг. З розвитком технологій способи виробництва змінювались, і тепер виробники намагаються отримати максимальну віддачу від працівників та обладнання. Тепер комп'ютери постійно з'єднуються та спілкуються між собою. Передові системи та технології поєднані з доступом до даних роблять можливою Індустрію 4.0.

Індустрія 4.0 – це назва, дана сучасній тенденції автоматизації та обміну даними в промислових технологіях. Він включає промисловий інтернет речей (IIoT), бездротові датчики, хмарні обчислення, штучний інтелект (AI) і машинне навчання (ML).

Прогнозне обслуговування вважається найдосконалішим із видів промислового технічного обслуговування через його прогностичний характер. Використовуючи певні об'єктивні параметри, ми можемо виміряти зміни в роботі обладнання та виявити порушення перш ніж вони переростуть в серйозні проблеми та помилки. Завдяки стратегії прогностичного обслуговування поломка ніколи не застане виробництво зненацька.

Індустрія 4.0 заснована на тому, що обладнання вже підключено до інтернету. Іншими словами, кожна частина промислового виробництва взаємодіє з іншими частинами, щоб адаптуватись до постійних змін та досягти оптимальної ефективності в будь-який час. Це означає, що не потрібно втручатися в поточні виробничі процеси, щоб мати можливість вимірювати фізичні параметри, так як обладнання вже спілкується між собою. Це велика перевага промислового обслуговування 4.0 (Maintenance 4.0).

Maintenance 4.0 включає цілісне уявлення про джерела даних, способи підключення, способи збору, способи аналізу та рекомендовані дії, які необхідно

вжити, щоб забезпечити надійність і управління активами за допомогою цифрових технологій.

З використанням програмного забезпечення ми завжди маємо доступ до даних про стан нашого обладнання. Комп'ютерна система допоможе передбачити та вирішити, коли настав ідеальний час для ремонту або заміни, перш ніж виникне помилка/поломка. Крім того, завдяки особливостям Industry 4.0, решта обладнання зможе адаптуватись і досягти оптимальної ефективності, доки певне обладнання буде знаходитись в ремонті.

Ключовими компонентами Industry 4.0 є: кіберфізична система (Cyber Physical Systems, CPS), штучний інтелект (Artificial Intelligence, AI), машинне навчання (Machine Learning, ML), промисловий інтернет речей (Industrial Internet of Things, IIoT), інтернет послуг (Internet of Services, IoS).

Кіберфізична система – це система, в якій існує взаємодія між комп'ютерною системою та фізичними машинами. Комп'ютерна система керує машиною, а натомість отримує дані датчиків або інші статуси.

Штучний інтелект – це широка галузь науки, яка займається створенням розумних машин, здатних виконувати завдання, які зазвичай вимагають людського інтелекту. Використовуючи ці технології, комп'ютери можуть виконувати конкретні завдання, обробляючи великі обсяги даних і розпізнаючи закономірності в даних.

Машинне навчання навчає комп'ютер вчитися знаходити закономірності у великій кількості даних і робити висновки на основі цих закономірностей.

Промисловий інтернет речей об'єднує машини, дані та людей. По-перше, він використовує мережу промислових пристроїв, таких як датчики та програмне забезпечення для обслуговування, і дозволяє їх обмінюватись інформацією один з одним. Це забезпечує платформу для ефективнішого відстеження, збору, обміну, доступу та аналізу великої кількості даних. Висновки, отримані з цих даних, потім використовуються для покращення виробничих процесів.

Ідея «Інтернету послуг» полягає в тому, що система повинна мати можливість виходити в інтернет, щоб використовувати послуги, які є корисними в

її домені. Це вимагає, щоб служби розроблялися з акцентом на сумісність, тобто щоб служба була структурована таким чином, щоб її можна було легко використовувати іншими системами. Це також вимагає, щоб послуги були відкриті для всіх або, принаймні, була налаштована система доступу, яка дозволяє кваліфікованим користувачам використовувати їх.

Великі дані описують величезну кількість інформації, яку ми можемо збирати та аналізувати. Великі дані часто характеризуються способом використання даних, їх здатністю визначати причини та наслідки, а також їхнім наслідком для прийняття рішень. Коли виробники отримують і використовують статистику великих даних, вони можуть визначити причини неефективності, зменшити витрати та оптимізувати процеси.

Для більшості компаній завданням є поглянути на те, що вони вже мають, а потім вносити зміни, щоб скористатись перевагами нових технологій. Впровадження нових технологій можна охарактеризувати наступними етапами:

- Етап характеристики продукту. Метою цього етапу є оснащення машин і компонентів функціями Індустрії 4.0, щоб вони могли генерувати дані, що стосуються їх стану та роботи, обробляти ці дані та виконувати всі необхідні дії.
- Етап аналізу даних. Мета цього етапу полягає в тому, щоб отримати нову інформацію із накопичених даних шляхом застосування аналітики до потоку даних, який генерується всіма підключеними елементами.
- Етап нових послуг. Мета цього етапу – запровадити нові корисні послуги/сервіси. Знання отримані на попередньому етапі, можуть бути використані для нових сервісів. Це може бути наприклад, живий моніторинг різних показників ефективності, щоб дати працівникам та керівництву корисну інформацію.

Прогнозне обслуговування вже відіграє важливу роль в Industry 4.0. Оскільки машини стають «розумнішими», а технології та платформи штучного інтелекту вдосконалюються, рішення щодо технічного обслуговування все більше будуть

створюватись і реалізовуватись з використанням цих технологій, щоб зменшити ризик людської помилки та збільшити продуктивність виробництва.

1.4 Обґрунтування доцільності дослідження технічного обслуговування для підвищення ефективності промисловості

Існує широкий діапазон оцінок витрат на технічне обслуговування машин, обладнання, компонентів у виробництві. За деякими оцінками це становить від 15% до 70% вартості виробленої продукції. З точки зору впливу передових методів технічного обслуговування, існують різноманітні оцінки з використанням ряду показників. Щоб краще зрозуміти витрати на технічне обслуговування, Національний інститут стандартів і технологій (NIST) зібрав дані від виробників. Результати були опубліковані в публікації [7], в якому розглядалися витрати на технічне обслуговування устаткування. У даному звіті розглядалися щорічні витрати на технічне обслуговування разом із збитками через неадекватні стратегії технічного обслуговування. Крім того було розглянуто сприйняті та помічені переваги інвестування та вдосконалення стратегій обслуговування.

Витрати на технічне обслуговування. В 2016 році витрати на технічне обслуговування обладнання для NAICS 321-339 (за винятком 324 і 325) оцінювались в 57,3 мільярда доларів. Додаткові витрати через несправності та збої були оцінені в 16,3 мільярда доларів, а витрати на додаткове обладнання, щоб захиститись від проблем з технічним обслуговуванням, обійшлися в 0,9 мільярда доларів. Загалом на технічне обслуговування було витрачено 74,5 мільярда доларів.

Збитки, які можна запобігти. В 2016 році збитки через проблеми з технічним обслуговуванням, які можна запобігти, склали 119,1 мільярда доларів: 18,1 – через простої, 0,8 – через дефекти, 100,2 – через втрату продажів через затримки та дефекти. Крім того, приблизно 16,03 травм і 0,05 смертей на мільйон співробітників були пов'язані з цими проблемами технічного обслуговування.

Переваги розширених стратегій технічного обслуговування. Очікувана вигода від застосування прогнозного обслуговування в 2016 році становила 6,5

мільярда доларів від скорочення простоїв і 67,3 мільярда доларів від збільшення продажів (загалом 73,8 мільярда доларів). Інші передбачувані переваги, такі як зменшення кількості дефектів, також можуть мати місце.

В таблиці 1.1 відображено результати дослідження [7], що відображають відчутні та помічені переваги передових методів технічного обслуговування. Перші 25% установ, які покладаються на реактивне технічне обслуговування, мали в 3,3 рази більше простоїв, ніж ті, що знаходяться в нижчих 25%. Також в них в 16 разів більше дефектів та у 2,8 разів більше втрачених продажів через дефекти в результаті технічного обслуговування, в 2,4 рази більше втрачених продажів через затримки з технічним обслуговуванням і 4,9 разів більше збільшення запасів через проблеми з обслуговуванням. У середньому 45,7% технічного обслуговування було реактивним. Ті, хто менше покладався на реактивне технічне обслуговування, а більше на профілактичне та/або прогнозне технічне обслуговування, з більшою ймовірністю використовували стратегію вилучення запасів (тобто виготовляли на замовлення). Наслідком є те, що реактивне технічне обслуговування знижує якість продукції і збільшує невизначеність у часі виробництва.

Серед тих установ, які в основному покладаються на профілактичне та попереджувальне технічне обслуговування (тобто менше ніж на 50% реактивного технічного обслуговування), перші 50% у прогнозному технічному обслуговуванні мали на 15% менше простоїв, на 87% менший рівень дефектів та на 66% менше збільшення запасів. Ті установи, які більше покладалися на профілактичне обслуговування, ніж на прогнозне, з більшою ймовірністю мали стратегію виготовлення товарів на замовлення. Крім того планове обслуговування пов'язане з більш високою якістю продукції та коротшим часом виробництва за рахунок скорочення простоїв.

Таблиця 1.1 – Відчутні та помічені переваги передових методів технічного обслуговування

	Реактивне (найвищі 25% виробництв) проти інших (нижчі 25% виробництв)	Профілактичне обслуговування проти прогнозного (найвищі 50% виробництв)	Невеликі/незначні інвестиції в технічне обслуговування проти помірних/значних інвестицій у технічне обслуговування
Співвідношення витрат на технічне обслуговування до вартості виробництва	Інші в 1,61 разів більше	Прогнозний в 3,16 разів більше	Помірні/значні в 1,61 разів більше
Відсоток запланованого часу виробництва, який є простоем	Реактивне в 3,28 разів більше	Профілактичне в 1,18 разів більше	Невеликі/незначні в 1,78 разів більше
Відсоток простою через реактивне технічне обслуговування	Реактивне в 2,18 разів більше	Профілактичне в 1,21 разів більше	Невеликі/незначні в 1,06 разів більше
Відсоток запланованого часу виробництва, який є простоем через проблеми з технічним обслуговуванням	Реактивне в 7,90 разів більше	Профілактичне в 2,13 разів більше	Невеликі/незначні в 2,32 разів більше
Відсоток дефектів	Реактивне в 16 разів більше	Профілактичне в 7,8 разів більше	Невеликі/незначні в 2,16 разів більше
Відсоток втрачених продажів через дефекти, спричинені проблемами технічного обслуговування	Реактивне в 2,81 разів більше	Прогнозне в 1,34 разів більше	Невеликі/незначні в 1,53 разів більше
Відсоток втрачених продажів через затримки, спричинені проблемами технічного обслуговування	Реактивне в 2,37 разів більше	Профілактичне в 1,21 разів більше	Невеликі/незначні в 1,41 разів більше
Співвідношення додаткових витрат через невіправні несправності та збої виробництва	Реактивне в 2,25 разів більше	Прогнозне в 1,05 разів більше	Помірні/значні в 1,41 разів більше
Відсоткове збільшення запасів через технічне обслуговування	Реактивне в 4,89 разів більше	Профілактичне в 2,98 разів більше	Помірні/значні в 1,12 разів більше

В роботі [8] були досліджені витрати/вигоди, пов'язані з інвестуванням в передові методи технічного обслуговування. Використовуючи дані, зібрані від виробників, було проаналізовано та оцінено збитки через неналежне технічне обслуговування та проведено порівняння між тими, хто покладається на реактивне технічне обслуговування, профілактичне та прогнозне обслуговування. Загальні річні витрати/втрати, пов'язані з обслуговуванням, оцінюються в середньому в 222 мільярда доларів, за оцінками за допомогою аналізу Монте-Карло. Респондентів розділили на три групи та порівняли. Перша група – це 50% респондентів, які найбільше поклались на реактивне обслуговування. Решта респондентів припускали, що є значні переваги від реактивного технічного обслуговування до профілактичного та/або прогнозного технічного обслуговування. Останні 50%, які більшою мірою покладаються на профілактичне та/або прогнозне обслуговування, мали на 57,2% менше незапланованих простоїв та на 78,5% менше дефектів. Порівняння між двома меншими групами, які більшою мірою покладаються на профілактичне та прогнозне обслуговування, показує, що на 18,5% менше незапланованих простоїв і на 87,3% менше дефектів у тих, хто більше покладається на прогнозне технічне обслуговування ніж на профілактичне.

В роботі [9] розглядається література та наявні дані для оцінки витрат і втрат, пов'язаних з різними методами технічного обслуговування виробництва. Оцінки витрат на технічне обслуговування, зроблені в журналах і статтях, використовують широкий діапазон показників. Крім того, значення показників мають широкий діапазон. Наприклад, вартість технічного обслуговування оцінюється від 15% до 70% вартості проданих товарів. Потенційний вплив на витрати на технічне обслуговування від застосування методів прогнозного технічного обслуговування не задокументований на національних рівнях держав. Оцінки, зроблені на рівні підприємства, показують, що вплив прогнозного технічного обслуговування має широкий діапазон показників і в межах кожного показника, широкий діапазон значень. Ці дослідження походять з різних країн. Існують оцінки зменшення витрат на технічне обслуговування, дефектів, поломок, аварій та простоїв разом із оцінками збільшення продуктивності та вироблення продукції. Висновком

дослідження в роботі [9] є те, що при використанні методів прогнозного технічного обслуговування зменшення витрат на технічне обслуговування може становити від 15% до 98%, а окупність інвестицій, як правило, оцінюється як сприятлива.

В роботі [10] аналізується вплив від впровадження методів прогнозного технічного обслуговування. На рисунку 1.2 відображується вплив впровадження методів прогнозного обслуговування для технічного обслуговування на різні показники промислового підприємства. На рисунку 1.2 можна побачити, що впровадження методів прогнозного обслуговування позитивно впливає на наступні показники технічного обслуговування та ефективності підприємства в цілому: зменшення витрат на технічне обслуговування (1), зменшення кількості поломок (2), зменшення кількості дефектів та переробок (3), зменшення аварій (4), збільшення випуску продукції / виробництва (5), підвищення продуктивності праці (6), зменшення часу простою (7), скорочення запасів (8), зменшення кількості відмов клієнтів (9).

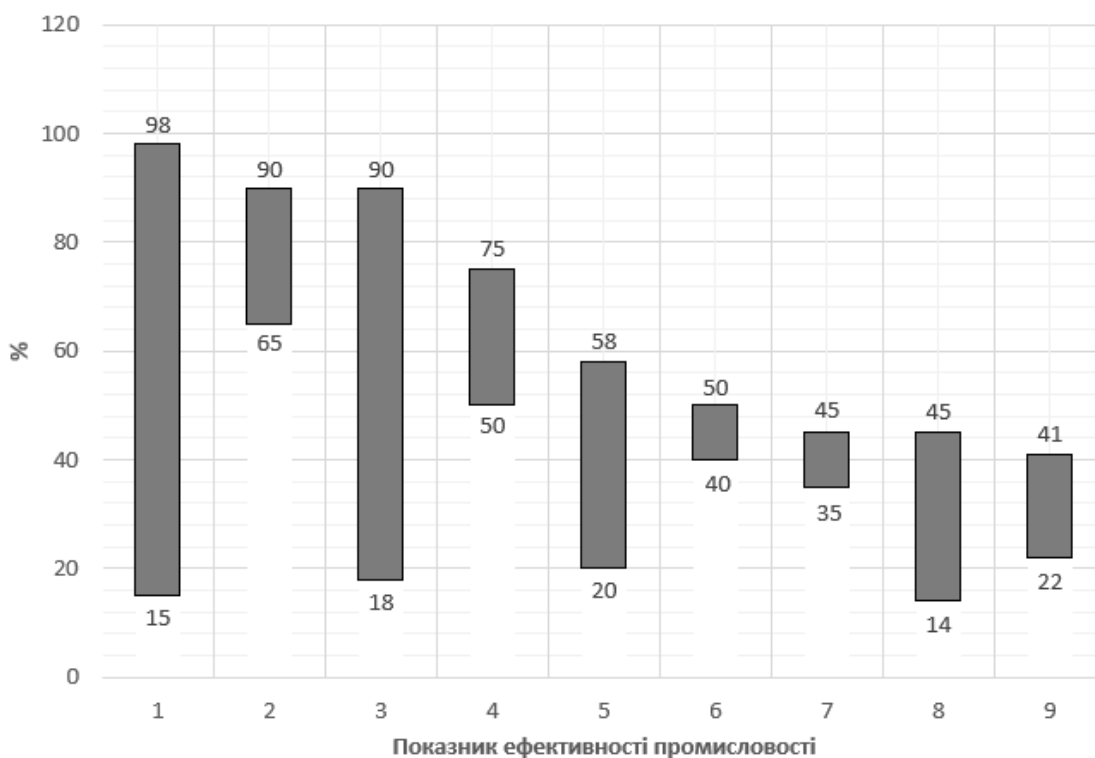


Рисунок 1.2 – Діапазон впливів на показники ефективності промисловості від впровадження методів прогнозного обслуговування

Отже в результаті огляду наукової літератури, можна зробити висновок, що впровадження методів прогнозного технічного обслуговування позитивно впливає на ефективність промислового підприємства, та допомагає покращити наступні показники:

- Значно скорочуються планові та позапланові простої, а отже, і витрати;
- Надання технікам цілеспрямованих вимог щодо перевірки, ремонту та інструментів;
- Продовження терміну служби устаткування (RUL);
- Скорочення чисельності технічної бригади, необхідної для ремонту устаткування.

1.5 Використання машинного навчання для прогнозного обслуговування

Своєчасне прогнозування несправностей і збоїв обладнання допомагає зменшити витрати на технічне обслуговування та ремонт, а також уникнути повної відмови та небажаних витрат на ремонт та заміну обладнання. Подальші фінансові втрати можуть бути не тільки прямими, а й непрямими – втрата довіри клієнтів і погіршення іміджу можуть викликати довгострокове зниження прибутку і відтік клієнтів. Використання прогнозової аналітики для прогнозування поломок дозволяє уникнути таких проблем.

Прогностична модель відповідає на два питання: що зламається і коли. Прогнозування несправності обладнання здійснюється як на основі накопичених даних, так і даних, отриманих в режимі реального часу.

З розвитком методів, пов'язаних з великими даними (наприклад, датчики, Інтернет речей (IoT), промисловий Інтернет речей (IIoT)) і постійного збільшення розміру великих даних, прогнозне обслуговування, що керується даними, стає все більш привабливим. Для отримання корисних знань і прийняття відповідних рішень із великих даних методи машинного навчання розглядаються як потужне рішення.

Протягом багатьох років деякі методи прогнозного обслуговування, засновані на машинному навчанні, виявились найбільш ефективними. Для прогнозного обслуговування потрібно підключати датчики до досліджуваного устаткування, і датчик буде відстежувати та збирати дані про роботу системи. Ці дані, отримані та зібрані датчиками для прогнозного обслуговування, відображаються з часовими рядами, що включають часові мітку та значення устаткування у визначений момент часу. Такі дані з мітками часу можуть допомогти програмі на базі машинного навчання точно передбачити збій із точним визначенням часу.

Прогнозне обслуговування на базі основі машинного навчання в основному створюється за допомогою будь-якого з двох методів згаданих нижче:

- Класифікація: цей підхід прогнозування передбачає можливість невдачі на будь-якому з наступних кроків;
- Регресія: цей підхід прогнозування часу, що залишився до виходу устаткування з ладу. Цей прогнозований час до збою системи також називається RUL.

Хоча перший підхід дає грубу відповідь, він може досягнути більшої точності, використовуючи менше даних. З іншого боку, останній використовує більше даних, але також надає більше деталей про майбутню поломку. Промислові підприємства широко використовують ці підходи прогнозування на основі машинного навчання для оцінки системних збоїв та їх запобігання заздалегідь.

Однією з основних проблем застосування алгоритмів машинного навчання для прогнозного обслуговування є вибір правильного робочого процесу, оскільки в літературі існує багато різних підходів до вирішення цієї проблеми, залежно від походження даних та цілей аналізу [11-13].

У цій роботі буде розглянуто послідовність дій для реалізації методів машинного навчання, представлену у роботі [14], представлену на рисунку 1.3. Проект машинного навчання завжди повинен починатись з ретельного і чіткого визначення цілей, оскільки система для прогнозного обслуговування виконує конкретне завдання, а визначення нечітких цілей може означати, що розроблена

модель не здатна точно передбачити, стан досліджуваної системи. Цілком можливо, що одним з найважливіших завдань для ефективної роботи з алгоритмами машинного навчання є здатність розуміти дані, і те, як вони пов'язані з завданням, яке ми хочемо вирішити.

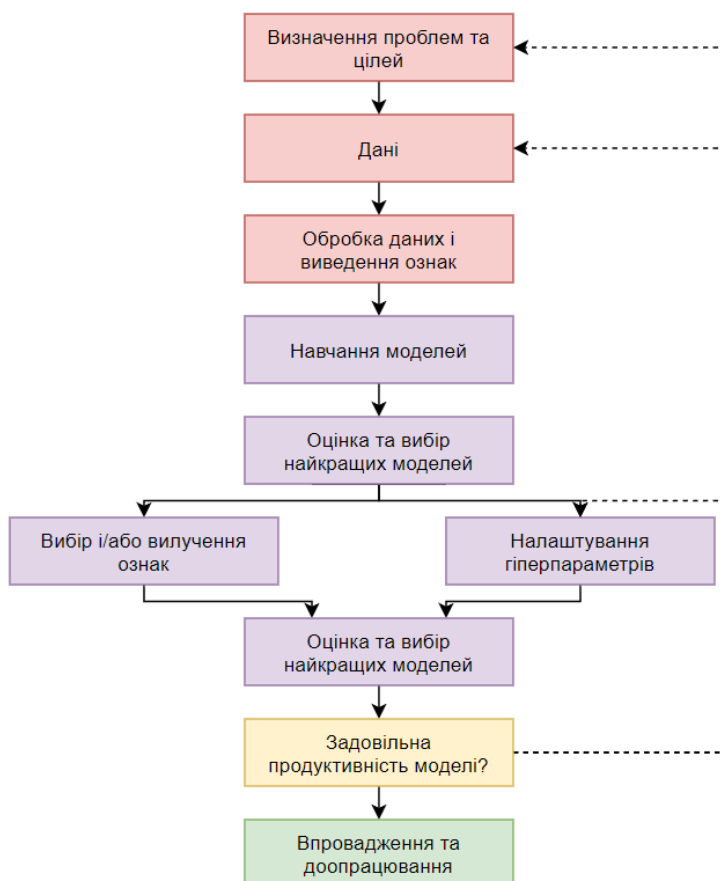


Рисунок 1.3 – Алгоритм реалізації методів машинного навчання

Типові алгоритми машинного навчання, такі як приховані моделі Маркова [15], напівмарківські моделі [16], самоорганізуюча нейронна мережа [17], метод опорних векторів (SVM) [18], мультимодальна глибока допоміжна векторна класифікація [19], глибокий випадковий ліс [20], генетичні алгоритми [21], сліпий поділ джерел (BSS) [22], нечітка логіка [23], алгоритми k-найближчого сусіда [24] та Байєсівські алгоритми [25], тощо, застосовувались для діагностики несправності обладнання.

У роботі [26] проводиться повний огляд останніх досягнень методів машинного навчання, що широко застосовуються для прогнозного технічного

обслуговування для розумних виробництв, класифікуючи дослідження відповідно до алгоритмів машинного навчання, категорій машинного навчання, машин/пристроїв, обладнання, що використовується, тощо. За результатами дослідження було встановлено, що найбільш використовуваними методами машинного навчання для прогнозного технічного обслуговування є: метод опорних векторів (SVM), випадковий ліс (RF) і штучні нейронні мережі (ANN). На основі літературного огляду було зазначено, що метод – випадковий ліс (RF) є найбільш широко застосовуваним методом машинного навчання для прогнозного обслуговування, оскільки він був застосований на різному промисловому обладнанні, компонентах або системах, включаючи обертові машини, турбовентиляторні двигуни, літаки, виробничі лінії, напівпровідники, промислові насоси, різальні установки, холодильні системи, вітрові турбіни, тощо.

У роботі [27] проведено систематичний огляд наукової літератури, з яких можна зробити декілька висновків. Стратегії прогнозного технічного обслуговування застосовуються до різноманітного обладнання в багатьох областях. Обладнання, де застосовуються методи машинного навчання, включає, але не обмежується ними, турбіни, двигуни, компресори, насоси, тощо. Близько 89% робіт використовуються реальні набори даних, а 11% використовують синтетичні набори даних. Що стосується алгоритмів машинного навчання в наукових публікаціях, то найбільш використовуваним є випадковий ліс (RF) – 33%, за ним йдуть методи, що засновані на нейронних мережах (NN), такі як штучна NN (ANN), згортова NN (CNN), довга короткочасна пам'ять (LSTM) і глибоке навчання – 27%, машина опорних векторів (SVM) – 25% і k-середніх – 13%.

Отже, в роботах [26] і [27], було встановлено, що найбільш використовуваними методами машинного навчання для прогнозного обслуговування є: випадковий ліс (RF), нейронні мережі (NN). Також можна відзначити, що методи машинного навчання для прогнозного обслуговування широко застосовуються до обладнання/машин/устаткування різних типів і в різних галузях.

2 МЕТОДИ МАШИННОГО НАВЧАННЯ ДЛЯ ПРОГНОЗУВАННЯ НЕСПРАВНОСТЕЙ ОБЛАДНАННЯ

2.1 Випадковий ліс (Random Forest)

Випадковий ліс (Random Forest) – це ансамблевий метод машинного навчання для класифікації, регресії, тощо, який працює за допомогою побудови численних дерев прийняття рішень (Decision Trees) під час тренування моделі й продукує моду для класів (класифікація) або усереднений прогноз (регресія) побудованих дерев. На рисунку 2.1 зображено приклад випадкового лісу.

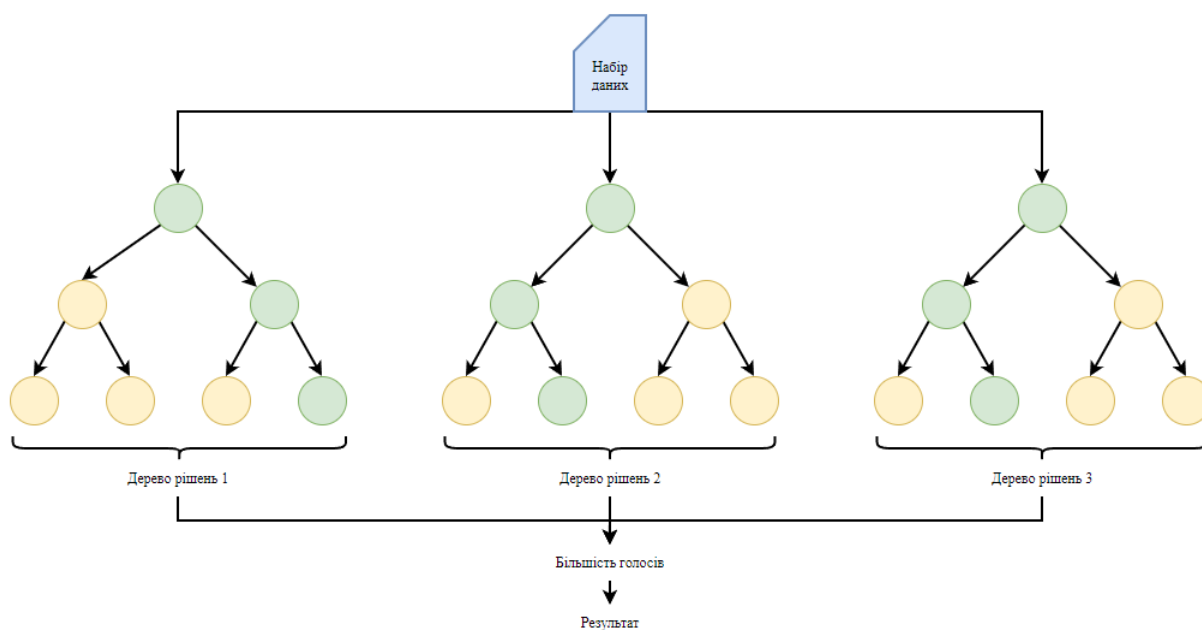


Рисунок 2.1 – Приклад випадкового лісу

У серії статей і технічних звітів Лео Брейман [28-31] продемонстрував, що значний вигрaш у класифікації та точності регресії можна отримати, використовуючи ансамблі дерев, де кожне дерево в ансамблі вирощується відповідно до випадкового параметра. Остаточні прогнози отримують шляхом агрегування ансамбля дерев. Оскільки базові складові ансамблю є предикторами з деревною структурою, і оскільки кожне з дерев ансамблю будується з

використанням випадкових параметрів, цей метод називається «випадковий ліс». Як підкреслюють різні емпіричні дослідження (наприклад [31]), випадкові ліси стали серйозними конкурентами найсучасніших методів, таких як підсилення (boosting) і метод опорних векторів (SVM). Вони швидкі та прості у реалізації, дають високоточні прогнози та можуть обробляти дуже велику кількість вхідних змінних без перенавчання.

Розглянемо приклад: $x \in R^N, x = [x_1, x_2, \dots, x_N]$, де x – вектор, що складається з N ознак. У класифікації метою прогнозуальної моделі є визначення класу, до якого належить конкретний екземпляр ознаки.

Нам необхідно призначити цей екземпляр одному з M класів C_1, C_2, \dots, C_M залежно від значень N ознак. Для навчання з наглядом модель виводиться на основі позначених спостережень, що подають у вигляді кортежів (x_i, y_i) , що містить ознаку x_i і правдиве цільове значення y_i . Ця колекція позначених даних відома як навчальний набір (2.1).

$$L = \{(x_1, y_1), \dots, (x_L, y_L)\} \quad (2.1)$$

де, x_1, x_L – ознаки для навчання;

y_1, y_L – правдиві цільові значення;

L – навчальний набір даних.

Оскільки, випадковий ліс – це ансамбль дерев, то передбачення випадкового лісу засноване на колективному прогнозі дерев, що утворюють ліс.

В умовах класифікації, передбаченням випадкового лісу, є найбільш домінуючий клас, серед прогнозів дерев.

Якщо є T дерев у лісі, то кількість отриманих голосів m , є (2.2).

$$v_m = \sum_{t=1}^T I(\hat{y}_t = m) \quad (2.2)$$

де, \hat{y}_t – є передбаченням для класу m ;

$I(\hat{y}_t == m)$ – функція індикатора;

v_m – голоси дерев для класу m .

Функція індикатора, повертає 1, якщо умова виконується, інакше вона дорівнює нулю.

Враховуючи ці голоси, остаточним прогнозом лісу є клас з найбільшою кількістю голосів, (2.3).

$$\hat{y}_t = \operatorname{argmax}_{m \in \{1, \dots, m\}} v_m, \quad (2.3)$$

де, v_m – голоси дерев.

Навчання випадкового лісу засноване на ідеї бутстрап-агрегування (Bootstrap aggregation), також відомого, як бегінг (Bagging) [32,33] (рисунок 2.2). Ансамблеве навчання на основі бегінгу працює наступним чином:

1. Випадкова вибірка (із заміною) L навчальних прикладів із навчального набору розміру L .
2. Тренування моделі використовуючи алгоритм дерева прийняття рішень, використовуючи дані отримані в результаті кроку 1.

Вищезазначені кроки повторюються для кожного з T дерев, які утворюють випадковий ліс. Кількість дерев у лісі, T , є гіперпараметром моделі, який зазвичай обчислюється сотнями або тисячами, в залежності від навчального набору даних. Його можна налаштувати за допомогою перехресної перевірки або помилки «out-of-bag». Помилка «out-of-bag» це помилка дерева на спостереженнях, які не були частиною даних отриманих кроком 1, яка використовується для навчання цього конкретного дерева.

Наведені вище кроки 1 і 2 працюють для створення будь-якого ансамблю моделей на основі бегінгу. У випадку випадкових лісів, подальша рандомізація вводить у вигляді бегінгу ознак (feature bagging).

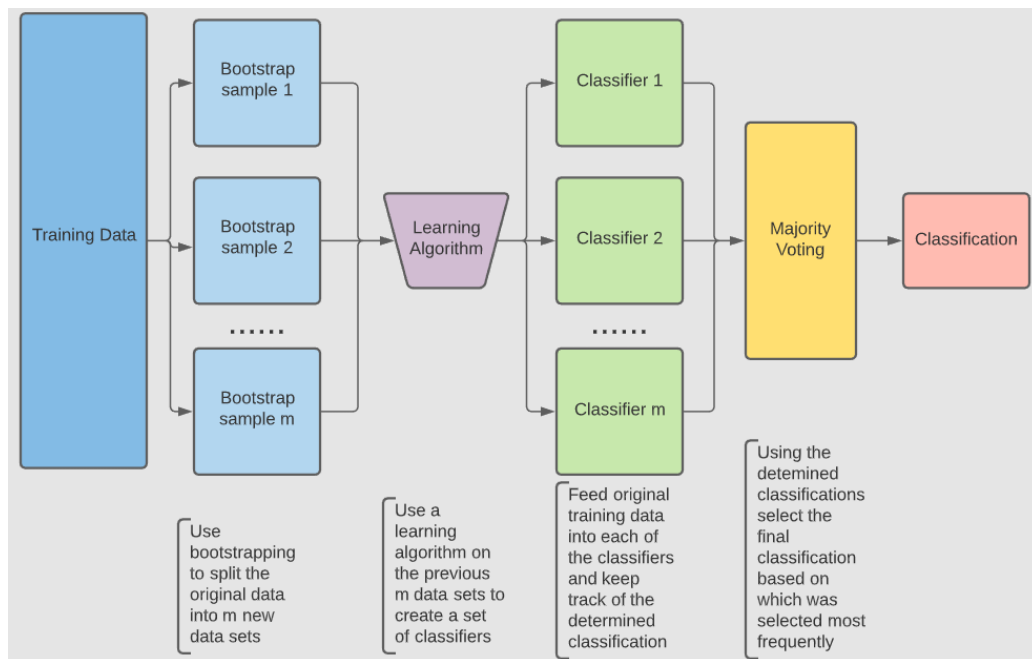


Рисунок 2.2 – Алгоритм бегінгу (bagging)

Однією з найбільших переваг випадкового лісу є його універсальність. Його можна використовувати як для завдань регресії, так і для завдань класифікації, а також легко побачити відносну важливість, яку він надає вхідним ознакам.

Випадковий ліс є також зручним для алгоритмом, оскільки гіперпараметри за замовчуванням, які він використовує, часто дають хороший результат прогнозу. Зрозуміти гіперпараметри досить просто, і їх також не так багато.

Однією з найбільших проблем машинного навчання є *overfitting*, але в більшості випадків цього не станеться завдяки класифікатору на основі методу випадковий ліс. Якщо в лісі достатньо дерев, то цю проблему можна уникнути.

Основне обмеження випадкового лісу полягає в тому, що велика кількість дерев може зробити алгоритм занадто повільним і неефективним для передбачення в реальному часі. Загалом, ці алгоритми швидко навчаються, але досить повільно створюють передбачення після навчання. Для більш точного прогнозу потрібно більше дерев, що призводить до повільнішої моделі. У більшості реальних додатків алгоритм випадкового лісу досить швидкий, але, безсумнівно, коли продуктивність під час виконання є важливим фактором, то можливо краще використати іншу модель.

Випадковий ліс є інструментом прогнозного моделювання, а не інструментом опису, тобто, якщо ви шукаєте опис взаємозв'язків ваших даних, інші підходи були б кращими.

Отже, випадковий ліс – чудовий алгоритм для навчання на початку процесу розробки моделі, щоб побачити, як він працює. Алгоритм також є чудовим вибором для тих, кому потрібно швидко розробити модель. Крім того, він є досить хорошим індикатором важливості, яку він надає вхідним ознакам. Випадкові ліси також дуже важко перевершити за продуктивністю. Звичайно, ми можемо знайти модель, яка може працювати краще, наприклад, нейронну мережу, але для її розробки необхідно більше часу, хоча вони можуть обробляти багато різних типів функцій, таких як двійкові, категоричні, числові.

Загалом, випадковий ліс – це (переважно) швидкий, простий і гнучкий інструмент, який не позбавлений обмежень.

2.2 Підсилення градієнта (Gradient Boosting Tree)

Посилення градієнта дерева або дерева рішень з посиленням градієнта (GBDT) є узагальненням посилення до довільних диференційованих функцій втрат [34-35]. Підсилення градієнта є ітераційним алгоритмом, що дозволяє реалізувати сильний класифікатор на основі слабких класифікаторів (наприклад, Decision Tree).

В загальному вигляді алгоритм навчання можна описати наступним виглядом ($L(F(x), y)$ – є диференційованою функцією втрат, $\{(x_i, y_i), i = 1, \dots, n\}$ – є навчальним набором даних, де n – кількість навчальних прикладів, та T – кількість ітерацій):

1. Впровадження моделі зі сталими значеннями (2.4):

$$F_0(x) = \operatorname{argmin}_a \sum_{i=1}^n L(a, y_i), \quad (2.4)$$

де i – кількість елементів навчального набору.

2. Для кожного кроку навчання t ($t = 1, \dots, T$):

а) Обраховуємо псевдо-залишкову модель (2.5):

$$r_{it} = -\left[\frac{\partial L(F(x_i), y_i)}{\partial F(x_i)}\right]_{F(x)=F_{t-1}(x)}, \quad (2.5)$$

де i – кількість елементів навчального набору.

б) Навчаємо базовий алгоритм «слабкого» учня $b_t(x)$ по вибірці $\{(x_i, r_{it}), i = 1, \dots, n\}$.

в) Вираховуємо коефіцієнт a_t перед базовим алгоритмом $b_t(x)$, як вирішення наступної задачі оптимізації (2.6):

$$a_t = \operatorname{argmin}_a \sum_{i=1}^n L(F_{t-1}(x_i) + ab_t(x_i), y_i). \quad (2.6)$$

г) Розширюємо модель (2.7):

$$F_t(x) = F_{t-1}(x) + a_t b_t(x). \quad (2.7)$$

3. Отримана функція: $F_T(x)$.

Підвищення градієнта зазвичай використовується з деревами рішень фіксованого розміру в якості базових учнів. Загальне підвищення градієнта на t -му кроці відповідало б дереву рішень $b_t(x)$ до псевдо-залишків. Нехай J_t – кількість листків дерева. Дерево розбиває вхідний простір на J_t областей $R_{1t} \dots R_{J_t}$ і прогнозує для кожного регіону значення. Використовуючи індикаторну нотацію, вихід $b_t(x)$ для входу x можна записати формулою 2.8.

$$b_t(x) = \sum_{j=1}^{J_t} y_{jt} I_{R_{jt}}(x), \quad (2.8)$$

де $b_t(x)$ – вихід дерева;

y_{jt} – прогнозоване значення в області R_{jt} .

$I_{R_{jt}}(x)$ – функція індикатора.

Тоді коефіцієнти множаться на деяке значення a_t вибране за допомогою лінійного пошуку, щоб мінімізувати функцію втрат, і модель оновлюється формулами 2.9 та 2.10.

$$F_t(x) = F_{t-1}(x) + \sum_{j=1}^{J_t} a_t I_{R_{jt}}(x). \quad (2.9)$$

$$a_t = \operatorname{argmin}_a \sum_{x_i \in R_{jt}} L(F_{t-1}(x_i) + a, y_i). \quad (2.10)$$

В результаті обчислень ми отримуємо модель градієнтного бустингу з використання дерев підтримки рішень як базових учнів.

На рисунку 2.3 зображено простий приклад використання методу підсилення градієнта.

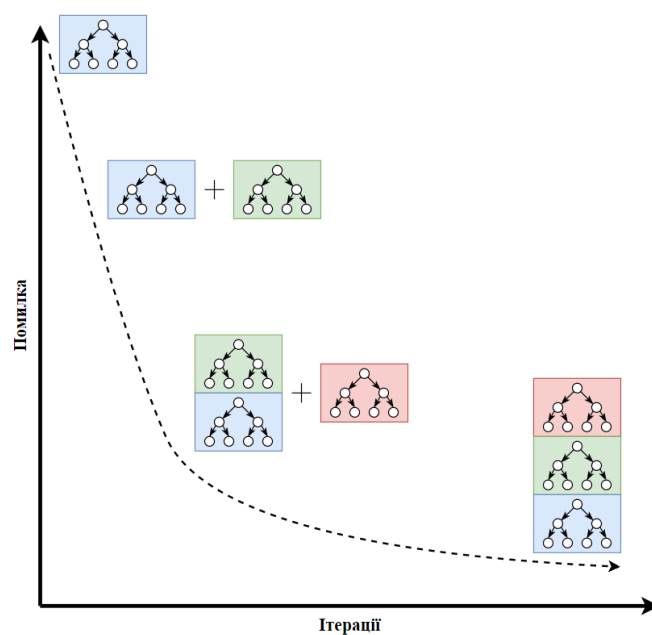


Рисунок 2.3 – Метод підсилення градієнта

Переваги даного методу:

- Часто забезпечує точність прогнозування, яку складно перевершити;
- Можна оптимізувати різні функції втрат і надає кілька опцій налаштування гіперпараметрів, що робить алгоритм доволі гнучким;
- Попередня обробка даних непотрібна – часто чудово працює з категоріальними та числовими значеннями як є.

Недоліки методу:

- Метод продовжуватиме вдосконалюватись, щоб звести помилки до мінімуму. Це може збільшити відхилення. Для нейтралізації негативних ефектів зведення помилок до мінімуму необхідно використовувати перехресну перевірку;
- Дорогі обчислення – метод часто вимагає багато дерев (> 1000), які можуть вичерпувати час і пам'ять;
- Висока гнучкість призводить до багатьох параметрів, які взаємодіють і сильно впливають на поведінку підходу (кількість ітерацій, глибина дерева, тощо).

Отже, моделі посилення градієнта – це потужні алгоритми, які можна використовувати для завдань класифікації. Моделі посилення градієнта можуть неймовірно добре працювати на дуже складних наборах даних, але як і інші методи, вони мають свої недоліки.

2.3 Нейронні мережі (Neural Networks)

Нейронні мережі (Neural network, NN) або Штучні нейронні мережі (Artificial neural networks, ANN) – це один з видів машинного навчання. ANN є системою з'єднаних і взаємодіючих між собою простих процесорів (штучних нейронів) [36]. Найпростіша нейронна мережа – це лише один нейрон, як показано на рисунку 2.4. Більші нейронні мережі формуються об'єднанням окремих нейронів. Процес переходу від входу через вузол до виходу аналогічний для всіх нейронів мережі.

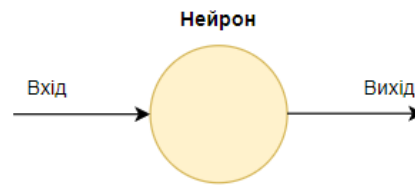


Рисунок 2.4 – Зображення найпростішої нейронної мережі

Зазвичай нейронні мережі мають вхідний шар, приховані шари та вихідний шар. Кожен шар має певну кількість нейронів. Приховані шари – це всі шари, що не є вхідними або вихідними. Вихідний шар відповідає прогнозним значенням. Вхідні дані в основному позначаються як x або X , а вихідні дані – \hat{y} або Y .

На рисунку 2.5 показано приклад 3-шарової нейронної мережі. Ця нейронна мережа має 6 нейронів у вхідному шарі, чотири нейрони у першому прихованому шарі, три нейрони у другому прихованому шарі і один нейрон у вихідному шарі.

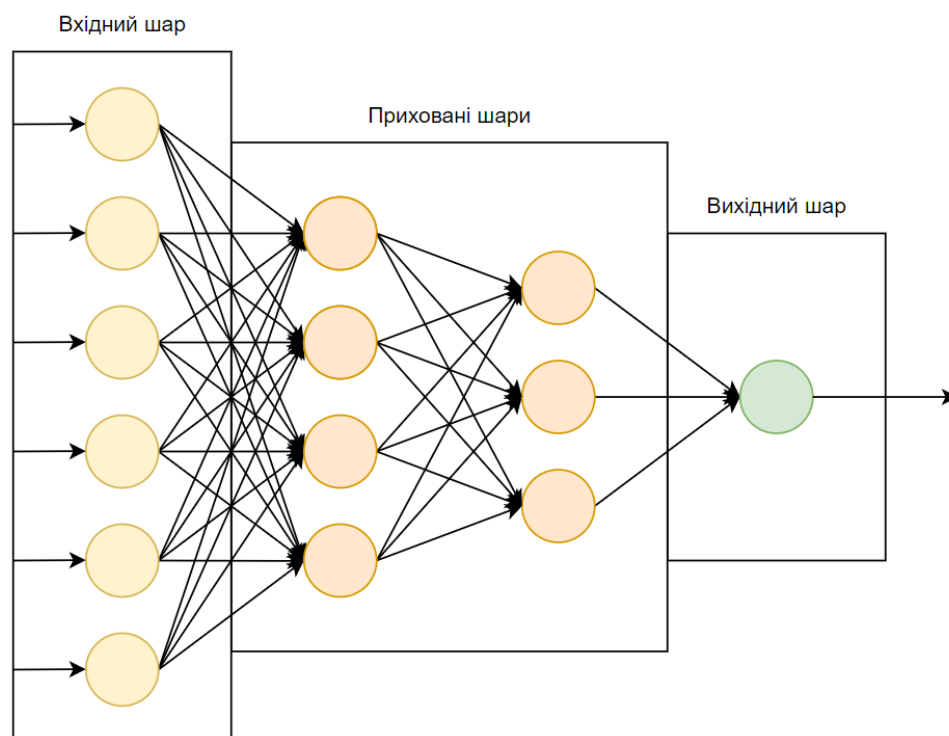


Рисунок 2.5 – Багатошарова нейронна мережа

Під час пошуку прогнозного значення деякі ознаки можуть бути більш доречними, ніж інші. Для цього вводиться значення ваги (weight – w або W). Це

дійсні числа, що виражають важливість змінних відповідних входів до виходів. Інший сценарій: коли певна функція важлива, незалежно від значення ознаки. Для цього сценарію додається зміщення (bias – b).

Для кожного нейрона в мережі вихід залежить від комбінації вхідних значень, ваг і зміщень. Наприклад, для нейронної мережі з одним входом для кожного нейрона буде виглядати, як показано у формулі (2.11).

$$y = wx + b \quad (2.11)$$

де, x – вхід нейрона,

y – вихід нейрона,

w – вага нейрона,

b – зміщення.

Під час налаштування нейронної мережі всі ваги та зміщення отримують початкове значення. Для початкового тестування зміщення зазвичай можна встановити на нуль, а коефіцієнти ваги на випадкове число.

Пряме поширення (Forward propagation) – це процес переходу вхідних значень x через нейронну мережу до прогнозного значення \hat{y} . На рисунку 2.6 показано приклад з трьома входами x_1 , x_2 , x_3 , що входять в нейрон, виходом нейрона є прогнозоване значення \hat{y} .

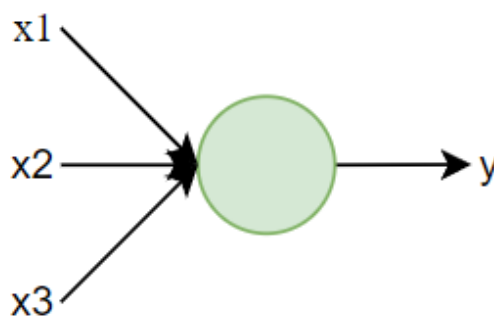


Рисунок 2.6 – Нейрон з трьома входами

Для $x = [x_1, x_2, x_3]$, пряме поширення може бути обчислене за формулою 2.12.

$$\hat{y} = wx + b \quad (2.12)$$

де, w – вага для вхідних значень $w = [w_1, w_2, w_3]$ (кожне w відповідає значенню x),

x – вхідне значення нейрона $x = [x_1, x_2, x_3]$.

Зазвичай, значення, знайдене в кожному нейроні позначається як z , де значення z вихідного шару дорівнює передбачуваному значенню \hat{y} .

Важливою реалізацією в нейронних мережах є функції активації (Activation function) [37]. Загальна ідея полягає в тому, що значення, знайдене в даному нейроні – z , має бути вищим за заданий поріг для активації нейрона. Якщо це значення досягнуто, то нейрон «запуститься» і на виході дасть одиницю, якщо менше, нейрон залишиться бездіяльним і на виході дасть нуль. Для багатьох нейронних мереж бажано, щоб нейрон виводив інші значення, а не просто 0 або 1. Снує багато різних функцій активації. Дві найчастіше використовуваних функцій – це сигмоїд (перетворює значення z на значення від 0 до 1) та ReLu (Rectified Linear Unit), що виводить найбільше значення з нуля або z . Дані функції активації обчислюються за формулами: сигмоїд – 2.13, ReLu – 2.14. Виходи даних функцій зображені рисунку 2.7.

$$sigmoid = \frac{1}{1 + e^{-z}} \quad (2.13)$$

$$ReLU = \max(0, wx + b) = \max(0, z) \quad (2.14)$$

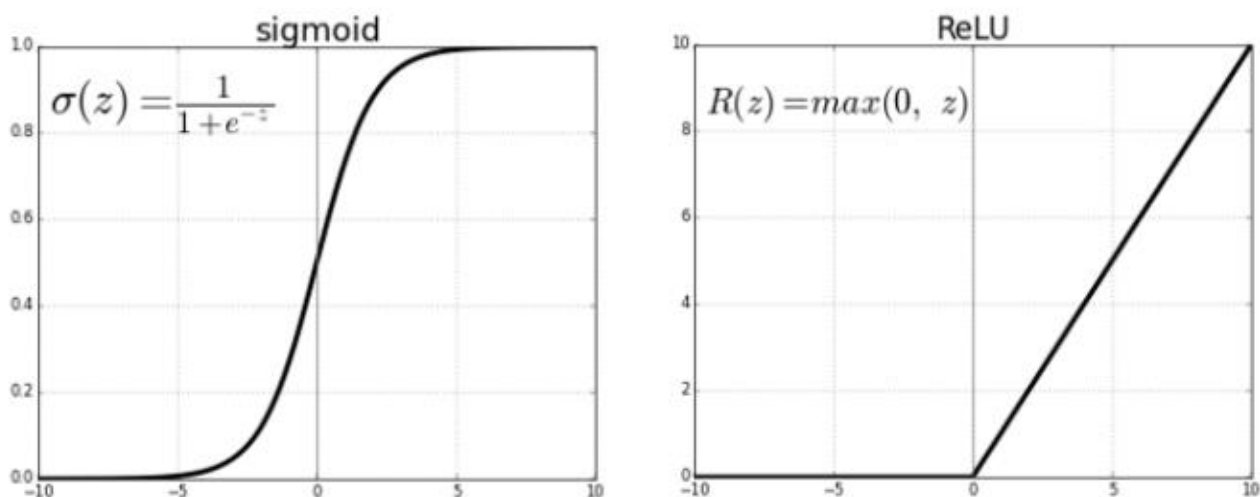


Рисунок 2.7 – Відображення функцій сигмоїд та ReLU

Функція активації softmax (або нормована експоненційна функція) застосовується в проміжному процесі навчання між повністю підключеним і вихідними шарами. Ця функція активації перетворює отримані числові значення з повністю підключеного шару в ймовірні значення, які знаходяться в інтервалі $[0, 1]$. Значення функції активації обчислюється за формулою 2.15. На рисунку 2.8 зображено графік функції softmax.

$$\text{softmax} = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (2.15)$$

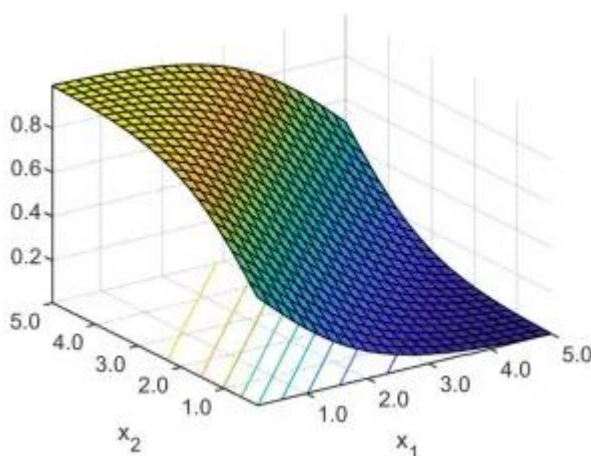


Рисунок 2.8 – Графік функції softmax

Дана функція активації використовується як функція активації у вихідному шарі моделей нейронної мережі, які передбачають мультиноміальний розподіл ймовірностей. Тобто функція активації softmax використовується для проблем класифікації з кількома класами, де членство в класі потрібно для більш ніж двох класів.

Продовжуючи з прикладом нейронної мережі з одним нейроном, що зображений на рисунку 2.6, загальне проходження по нейрону (з функцією активації) можна уявити, як показано на рисунку 2.9 і математично записати формулами 2.16 та 2.17.

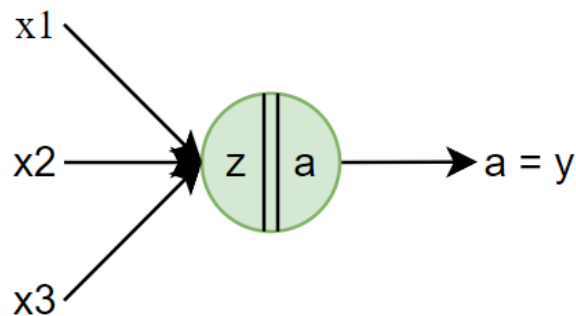


Рисунок 2.9 – Нейрон з функцією активації

$$z = w^T x + b \quad (2.16)$$

де, x – вхід нейрона, $x = [x_1, x_2, x_3]$,

z – вихід нейрона,

w – вага нейрона, $w = [w_1, w_2, w_3]$,

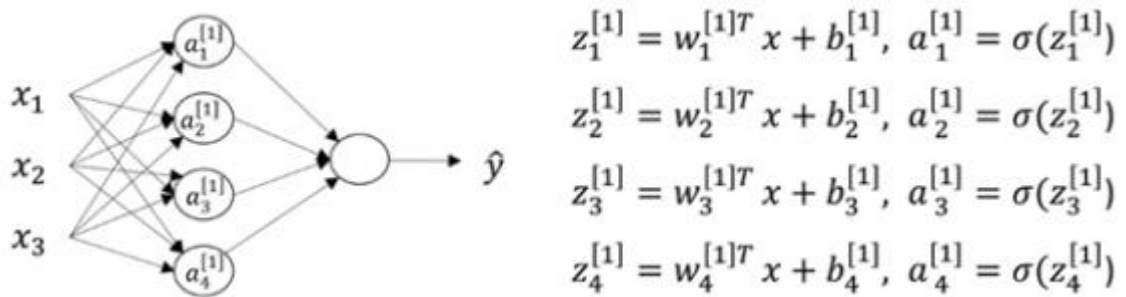
a – вихід функції активації (прогнозоване значення),

b – зміщення.

$$a = \sigma(z) \quad (2.17)$$

де, $\sigma(z)$ – функція активації.

Щоб додатково проілюструвати процес прямого поширення, до нейронної мережі додано прихований шар (рисунок 2.10). Прихований шар містить чотири нейрони. Тепер для позначення функцій у нейронній мережі використаємо наступне: для кожного шару використаємо великий регістр та для кожного нейрона в шарі будемо використовувати малий регістр.



$$z_1^{[1]} = w_1^{[1]T} x + b_1^{[1]}, a_1^{[1]} = \sigma(z_1^{[1]})$$

$$z_2^{[1]} = w_2^{[1]T} x + b_2^{[1]}, a_2^{[1]} = \sigma(z_2^{[1]})$$

$$z_3^{[1]} = w_3^{[1]T} x + b_3^{[1]}, a_3^{[1]} = \sigma(z_3^{[1]})$$

$$z_4^{[1]} = w_4^{[1]T} x + b_4^{[1]}, a_4^{[1]} = \sigma(z_4^{[1]})$$

Рисунок 2.10 – Зліва: Нейронна мережа, що містить чотири нейрони у прихованому шарі; Справа: рівняння z і a для прихованого шару

Замість запису функцій a і z для кожного з нейронів, їх можна записати як вектор для кожного шару. Відповідно для нейронної мережі на рисунку 2.10, обчислення шарів ми можемо записати формулами 2.18 і 2.19 для першого шару та 2.20 і 2.21 для другого.

$$z^{[1]} = W^{[1]}x + b^{[1]} \quad (2.18)$$

$$a^{[1]} = \sigma(z^{[1]}) \quad (2.19)$$

$$z^{[2]} = W^{[2]}a^{[1]} + b^{[2]} \quad (2.20)$$

$$a^{[2]} = \sigma(z^{[2]}) \quad (2.21)$$

Звідси узагальнені рівняння для кожного шару можна описати наступними формулами 2.22 та 2.23. Де a – вихід одного шару і вхід для наступного, а $a^{[0]} = x$, тобто входу першого шару.

$$z^{[l]} = W^{[l]}a^{[l-1]} + b^{[l]} \quad (2.22)$$

$$a^{[l]} = \sigma(z^{[l]}) \quad (2.23)$$

Щоб навчити нейронну мережу, важливо визначити, наскільки добре вона прогнозує правильне значення. Втрата, L , або іноді її називають функцією помилки – це функція, яка вимірює, наскільки наш результат \hat{y} відповідає істинній мітці y . Одним із прикладів функції помилки може бути функція квадратичної помилки (формула 2.24) або функція логістичної помилки (формула 2.25).

$$L = (y - \hat{y})^2 \quad (2.24)$$

$$L = -(y \log \hat{y} + (1 - y) \log(1 - \hat{y})) \quad (2.25)$$

де, L – значення помилки,
 y – істинне значення,
 \hat{y} – прогнозоване значення.

Багатошаровий перцептрон (MLP) – це нейронна мережа з прямим зв'язком, яка є методом глибокого навчання. Багатошаровий перцептрон характеризується кількома шарами вхідних вузлів, з'єднаних у вигляді орієнтованого графа між вхідними і вихідними шарами. Даний метод використовує зворотне поширення для навчання моделі. Багатошаровий перцептрон можна зобразити як показано на рисунках 2.6 та 2.11.

Розглянемо математичне формулювання багатошарового перцептрону. Наведено набір навчальних прикладів $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, де $x_i \in R^n$ і $y \in$

$\{0, 1\}$ і один прихований шар з одним нейроном. Дана нейронна мережа обчислюється формулою 2.26, де $W_1 \in R^m$ і $W_2, b_1, b_2 \in R$ – є параметрами моделі. W_1, W_2 – представляють ваги вхідного і прихованого шару відповідно; b_1, b_2 – представляють зміщення прихованого і вихідного шару відповідно; g – функції активації.

$$f(x) = W_2 g(W_1^T x + b_1) + b_2 \quad (2.26)$$

Для бінарної класифікації вихід нейрона проходить через функцію активації – сигмоїд (формула 2.25), щоб отримати вихідні значення від нуля до одиниці. Поріг встановлений на 0,5, призначав би вибірки результатів, більших або рівних 0,5 до «позитивного» класу, а решту до «негативного» класу. Якщо є більше двох класів, то як функція активації буде використовуватись функція softmax (формула 2.27). Результатом функції є вектор, що містить ймовірності того, що вибірка належить кожному класу. Вихідним значенням буде клас з найбільшою ймовірністю.

Багатошаровий перцептрон використовує різні функції втрат залежно від типу проблеми. Функція втрат для класифікації це – крос-ентропія (Cross-Entropy) і для двійкової класифікації обчислюється за формулою 2.27, де $a||W||_2^2$ – є терміном L2 -регуляризації (так званим «штрафом»), і a – є невід’ємним гіперпараметром, який контролює величину штрафу

$$Loss(\hat{y}, y, W) = -y \ln \hat{y} - (1 - y) \ln(1 - \hat{y}) + a ||W||_2^2 \quad (2.28)$$

Починаючи з початкових випадкових ваг, багатошаровий перцептрон мінімізує функцію втрат шляхом багаторазового оновлення цих ваг. Після обчислення втрат зворотній прохід поширює його з вихідного шару на попередні шари, надаючи кожному параметру ваги з оновленим значенням, призначеним для зменшення втрат. Алгоритм зупиняється, коли досягає заданої максимальної кількості ітерацій; або коли поліпшення втрат нижче певного заданого числа.

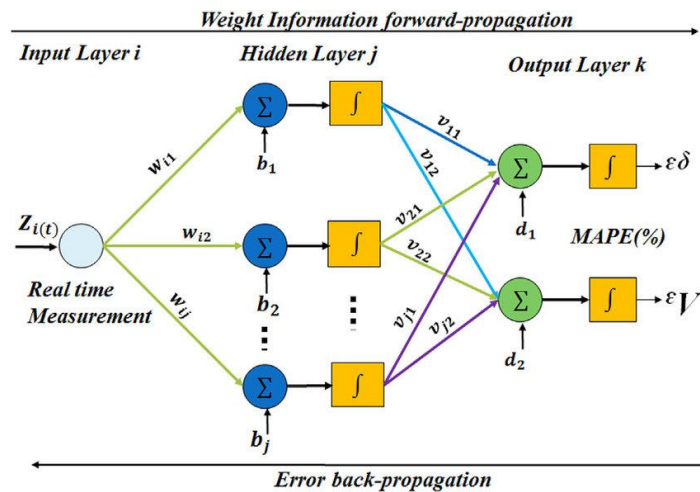


Рисунок 2.11 – Модель багатошарового перцептрона

Основна мета будь-якого алгоритму машинного навчання або глибокого навчання – передбачити результати, ближчі до реальних. Алгоритми оптимізації – це процедура, яка виконується ітеративно, порівнюючи різні рішення, поки не буде знайдено оптимальне рішення, яке також називають глобальним мінімумом [38, 39].

Гرادієнтний спуск (Gradient Descent). Градієнтний спуск – це найпростіший і найпопулярніший алгоритм оптимізації. Це алгоритм оптимізації першого порядку, який залежить від похідної першого порядку функції помилки. Він розраховує, яким чином потрібно змінити ваги, щоб досягти мінімуму. За допомогою зворотного поширення втрати переносяться з одного шару на інший, а параметри моделі (ваги та зміщення) змінюються залежно від втрат, щоб їх можна було мінімізувати. Градієнтний спуск зображено на рисунку 2.12.

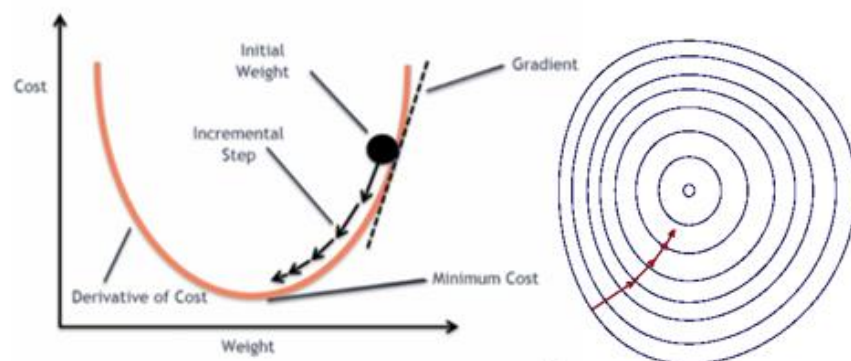


Рисунок 2.12 – Градієнтний спуск (Gradient Descent)

Градiєнт вказує напрямок збільшення. Оскільки нам потрібно мінімізувати втрати, ми рухаємось в протилежному напрямку. Потім ми оновлюємо наші параметри за формулами 2.28 та 2.29 в напрямку, протилежному градієнтам, при цьому швидкість навчання η визначає розмір кроків, які ми робимо для досягнення мінімумів.

$$W_{\text{оН}} = W - \eta \frac{\partial C}{\partial W} \quad (2.28)$$

$$b_{\text{оН}} = b - \eta \frac{\partial C}{\partial b} \quad (2.29)$$

Перевагами даного алгоритму є: легко інтерпретується; менш складний за інші оптимізатори, оскільки є похідною першого порядку. Недоліками даного алгоритму є: з дуже великими наборами даних може знадобитись занадто багато часу для обчислень; велика потреба в пам'яті для обчислення градієнта для всього набору даних.

Стохастичний градієнтний спуск (Stochastic Gradient Descent). Це наступна версія алгоритму градієнтного спуску. Алгоритм стохастичного градієнтного спуску (рисунок 2.13) оновлює параметри після обробки кожного навчального прикладу, що призводить до частішого оновлення. Стохастичний градієнтний спуск долає один з основних недоліків градієнтного спуску, йому потрібно менше обчислень. Градієнтний спуск використовує зайві обчислення оскільки перераховує градієнти для подібних прикладів перед кожним оновленням параметра, тоді як стохастичний градієнтний спуск виконує одне оновлення за раз, і зазвичай це набагато швидше.

Перевагами даного алгоритму є: часте оновлення параметрів, таким чином оптимізація відбувається швидше; Вимагає менше пам'яті ніж метод градієнтного спуску. Недоліками даного методу є: висока дисперсія параметрів моделі; швидкість навчання; швидкість навчання має бути змінена, щоб модель працювала

подібно до градієнтного спуску або щоб дані з постійною швидкістю навчання оброблялись пакетами, щоб зменшити дисперсію параметрів.

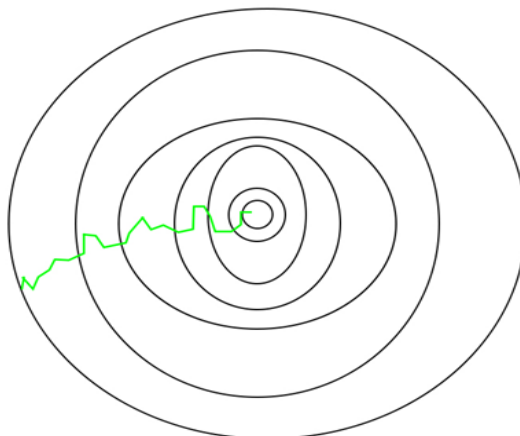


Рисунок 2.13 – Стохастичний градієнтний спуск

Міні-пакетний градієнтний спуск – це покращена версія градієнтного спуску, яка регулює дисперсію стохастичного градієнтного спуску (рисунок 2.14). Це досягається шляхом розгляду невеликих фрагментів або міні-пакетів даних для кожної ітерації для оновлення параметрів.

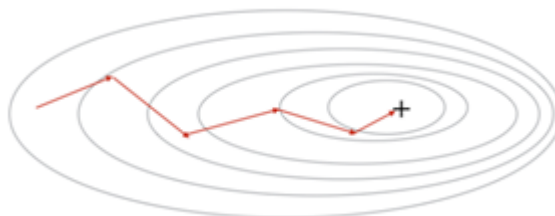


Рисунок 2.14 – Міні-пакетний градієнтний спуск

Перевагами даного методу є: чітко зменшує дисперсію параметрів порівняно з стохастичним градієнтним спуском та часто оновлює параметри порівняно з градієнтним спуском; вимагає менше пам'яті в порівнянні з градієнтним спуском. Недоліком даного методу є те, що оскільки швидкість навчання постійна, то конвергенція не є гладкою.

Адаптивний градієнтний спуск (Adagrad, Adaptive Gradient Descent). В даному методі оптимізації, ми маємо швидкість навчання η , яка змінюється

динамічно. Очікується, що ця функція допоможе досягти оптимальних значень швидше ніж раніше. Чим ближче параметр наближається до мінімуму, тим більше зменшується швидкість навчання для більш плавної конвергенції (рисунок 2.15). Відповідно для цього методу параметри обчислюються формулами 2.30 та 2.31, де η – початкова швидкість навчання, η'_t - динамічна швидкість навчання (обчислюється формулою 2.32), α_t – сума квадратів градієнтів за часом t (обчислюється формулами: 2.33 – для ваги та 2.34 для зміщень), ε – це невелике число, додане, щоб уникнути помилки ділення на нуль.

$$W_{t+1} = W_t - \eta'_t \frac{\partial C}{\partial W_t} \quad (2.30)$$

$$b_{t+1} = b_t - \eta'_t \frac{\partial C}{\partial b_t} \quad (2.31)$$

$$\eta'_t = \frac{\eta}{\sqrt{\alpha_t + \varepsilon}} \quad (2.32)$$

$$\alpha_t = \sum_{i=1}^t \left(\frac{\partial C}{\partial W_i} \right)^2 \quad (2.33)$$

$$\alpha_t = \sum_{i=1}^t \left(\frac{\partial C}{\partial b_i} \right)^2 \quad (2.34)$$

Градієнти за часом обчислюються, підводяться до квадрату і підсумовуються разом. Очікується, що з часом це значення буде зростати. Таким чином, для кожного кроку швидкість навчання ділиться на квадратний корінь $\alpha_t + \varepsilon$ (який збільшується з кроками), тим самим зменшуючи швидкість навчання.

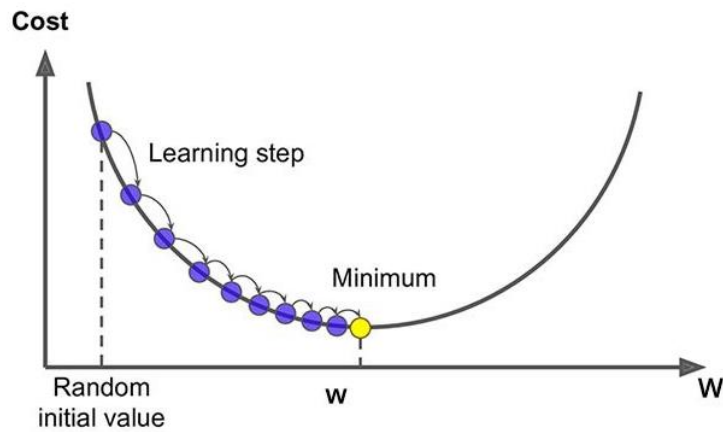


Рисунок 2.15 – Конвергенція. Метод «Adagrad»

Перевагами даного методу є: не потрібно вручну налаштовувати швидкість навчання. Плавно наближаться до оптимальних параметрів у порівнянні з іншими методиками. Недоліками даного методу є: багато обчислень; для дуже глибоких нейронних мереж параметри можуть радикально зменшитись, що призведе до проблеми зникнення градієнту.

Середньоквадратичне поширення (RMS Prop, Root Mean Square Propagation) є широко використовуваним методом оптимізації для міні-пакетного навчання нейронних мереж. Цей процес ефективно усереднює градієнти для послідовних міні-пакетів, щоб ваги можна було точно налаштовувати, регулюючи проблему зменшення градієнта за допомогою Adagrad. RMS Prop досягає проблеми зменшення градієнта шляхом експоненційного згладжування швидкості навчання. Параметри обчислюються формулами 2.30 та 2.31, η'_t - динамічна швидкість навчання (обчислюється формулами 2.35 – для ваги, та 2.36 – для зміщень), S_{dw_t} , S_{db_t} – експоненційні ковзні середні градієнтних квадратів обчислюються формулами 2.37 та 2.38 відповідно.

$$\eta'_t = \frac{\eta}{\sqrt{S_{dw_t} + \epsilon}} \quad (2.35)$$

$$\eta'_t = \frac{\eta}{\sqrt{S_{db_t} + \varepsilon}} \quad (2.36)$$

$$S_{dW_t} = \beta S_{dW_{t-1}} + (1 - \beta) \left(\frac{\partial C}{\partial W_{t-1}} \right)^2 \quad (2.37)$$

$$S_{db_t} = \beta S_{db_{t-1}} + (1 - \beta) \left(\frac{\partial C}{\partial b_{t-1}} \right)^2 \quad (2.38)$$

Експоненціальні ковзні середні для квадратного градієнта допомагають нам регулювати проблему адаптивного градієнта. Він використовує величину останніх градієнтних спусків для нормалізації градієнта. У цьому випадку швидкість навчання ніколи не стає занадто малою.

Перевагами даного методу є: швидкість навчання коригується на основі останніх градієнтів, а не всіх; кожен параметр може мати різну швидкість навчання.

Адаптивний момент (Adam, Adaptive Moment). Цей оптимізатор оновлює параметри на основі швидкості навчання від RMS Prop і за допомогою згладжування градієнтів з оптимізаторів: імпульс та стохастичний градієнтний спуск. Параметри обчислюються за формулами 2.39 – ваги та 2.40 – зміщення, де V_{dW_t} , V_{db_t} – експоненційні середні ковзні квадратів градієнтів для згладжування швидкості навчання (RMS Prop), S_{dW_t} , S_{db_t} – експоненційні середні ковзні градієнтів для градієнтного спуску (імпульс і стохастичний градієнтний спуск).

$$W_{t+1} = W_t - \eta'_t V_{dW_t} \quad (2.39)$$

$$\eta'_t = \frac{\eta}{\sqrt{S_{dW_t} + \varepsilon}} \quad (2.40)$$

$$V_{dW_t} = \beta_1 V_{dW_{t-1}} + (1 - \beta_1) \left(\frac{\partial C}{\partial W_{t-1}} \right) \quad (2.41)$$

$$S_{dw_t} = \beta_2 S_{dw_{t-1}} + (1 - \beta_2) \left(\frac{\partial C}{\partial W_{t-1}} \right)^2 \quad (2.42)$$

$$b_{t+1} = b_t - \eta'_t V_{db_t} \quad (2.43)$$

$$\eta'_t = \frac{\eta}{\sqrt{S_{db_t} + \varepsilon}} \quad (2.44)$$

$$V_{db_t} = \beta_1 V_{db_{t-1}} + (1 - \beta_1) \left(\frac{\partial C}{\partial b_{t-1}} \right) \quad (2.45)$$

$$S_{db_t} = \beta_2 S_{db_{t-1}} + (1 - \beta_2) \left(\frac{\partial C}{\partial b_{t-1}} \right)^2 \quad (2.46)$$

Зазначені у формулах (2.43, 2.44, 2.45, 2.46) гіперпараметри β_1 і β_2 різні, але зазвичай знаходяться в діапазоні (0.95 – 0.99).

Перевагами даного методу є: може обробляти «розріджені» градієнти в наборах даних; значення за замовчуванням працюють добре в більшості випадків; вимагає менше обсягів пам'яті та обчислювальної потужності; добре працює з великими наборами даних. Недоліком даного методу є те, що в деяких випадках можуть не знайтись оптимальні параметри.

На рисунках 2.16 та 2.17 порівняно декілька найбільш відомих оптимізаторів. Ми бачимо, що точність тренування найкраща у оптимізаторів: Adam, Adamax, SGC. Для точності тренування, після 50-ти поколінь, продуктивності алгоритмів подібні для Adam, Adamax, SGC. Помилка під час перевірки для алгоритмів Adamax і SGC збільшуються до кінця, що свідчить про необхідність механізму ранньої зупинки. Найкращі результати (найменше значення помилки, точність) має оптимізатор Adam.

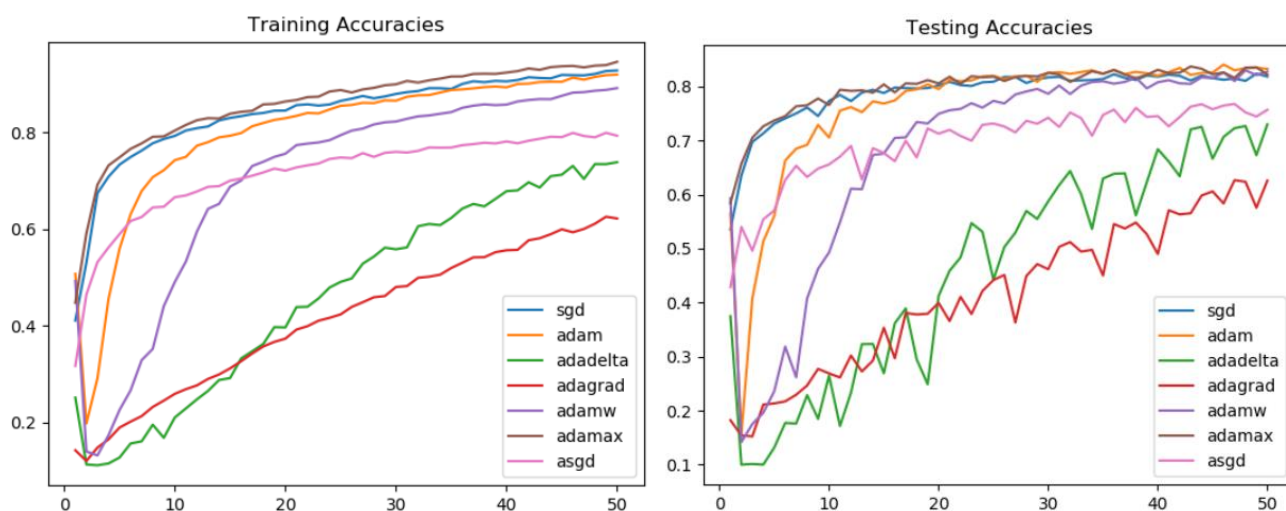


Рисунок 2.16 – Порівняння оптимізаторів для нейронних мереж (Точність)

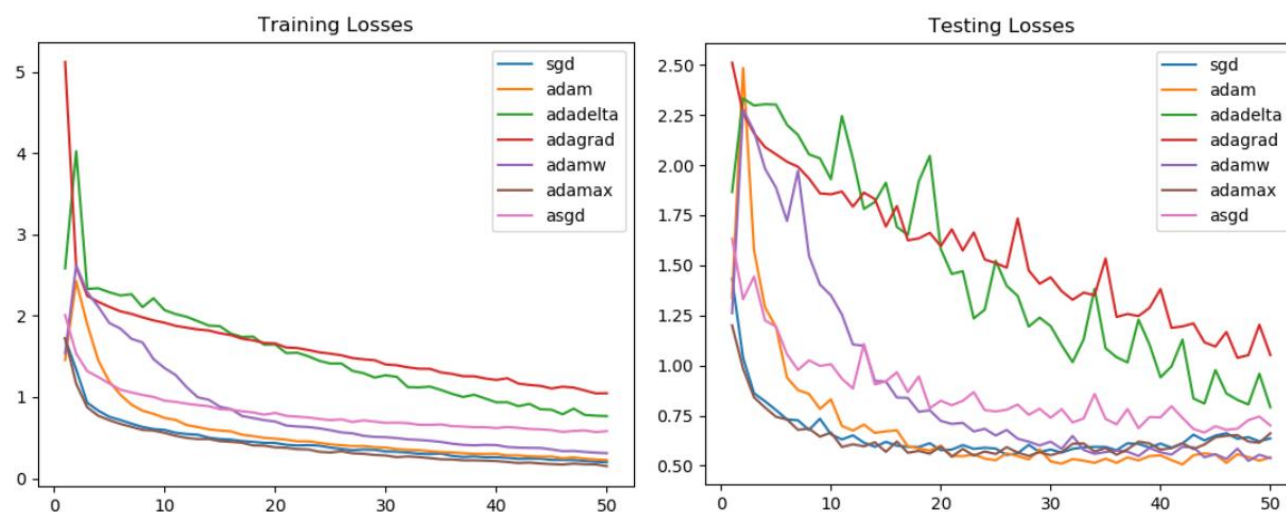


Рисунок 2.17 – Порівняння оптимізаторів для нейронних мереж (Помилка)

Переваги нейронних мереж:

- Зберігання інформації у всій мережі: інформація, зберігається у всій мережі, а не в базі даних. Зникнення кількох частин інформації в одному місці не заважає мережі функціонувати;
- Здатність працювати з неповними знаннями: після навчання нейронні мережі можуть працювати навіть маючи неповну інформацію. Втрата продуктивності мережі залежить від важливості ненаданої інформації;
- Стійкість до відмов: пошкодження однієї або кількох шарів нейронної мережі не заважає їй генерувати вихідні дані;

- Наявність розподіленої пам'яті: для того, щоб нейронна мережа могла навчитися, необхідно мати інформації з прикладами та навчити мережу відповідно до бажаного результату. Успіх мережі прямо пропорційний навчальним даним, і якщо навчальні дані не відображають всі можливі ситуації, що можуть відбутися з об'єктом, мережа може давати помилковий вихід;
- Можливість паралельної обробки.

Недоліки нейронних мереж:

- Апаратна залежність: нейронні мережі вимагають процесорів з паралельною потужністю обробки відповідно до їх структури;
- Визначення належної структури мережі: немає конкретного правила для визначення структури нейронних мереж. Відповідна структура мережі досягається шляхом досвіду та методом проб та помилок;

2.4 Вибір методів машинного навчання та дослідження метрик якості методів машинного навчання

За результатами дослідження методів машинного навчання для вирішення завдання прогнозування несправностей промислового обладнання було обрано наступні методи машинного навчання: випадковий ліс, градієнтний бустинг, нейронна мережа. На основі огляду літературних джерел було встановлено, що дані методи машинного навчання часто використовуються для прогнозування несправностей в стратегіях прогнозного технічного обслуговування.

Також дані методи машинного навчання використовуються для обладнання різних типів в різних галузях промисловості, тому дослідження та використання цих методів є доцільним.

Для оцінки точності моделей буде використовуватись матриця невідповідностей та показники точності, які можна за допомогою неї обрахувати.

Матриця невідповідностей також відома як матриця помилок (Confusion matrix) – це таблиця (рисунок 2.18), яка часто використовується для опису ефективності моделі класифікації на наборі тестових даних, для яких відомі

справжні значення [40]. Справжні позитивні та справжні негативні – це спостереження, які правильно передбачені (на рисунку 2.18 – позначені зеленим кольором).

		Predicted class	
		Class = Yes	Class = No
Actual Class	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

Рисунок 2.18 – Матриця помилок

Справжні позитивні показники (True Positives, TP) – це правильно передбачені істинні значення, що означає, що значення фактичного класу «+» і значення прогнозованого класу – «+» є однаковими.

Справжні негативні показники (True Negatives, TN) – це правильно передбачені негативні значення, які означають, що значення фактичного класу «-» і значення прогнозованого класу «-» є однаковими.

Помилкові позитивні та помилково негативні, це ті значення, що виникають, коли наш фактичний клас не збігається з прогнозованим.

Помилкові позитивні (False Positives, FP) – коли фактичний клас «-», а прогнозований «+».

Помилкові негативні (False Negatives, FN) – коли фактичний клас «+», а прогнозований «-».

Параметри TP, TN, FP, FN використовуються для розрахунку метрик: Accuracy, Precision, Recall.

Точність (Accuracy) – є найбільш інтуїтивно зрозумілим показником ефективності моделі, і це просто співвідношення правильно передбачених прикладів до їх загальної кількості, обчислюється за формулою 2.47. Хоч точність і є чудовим показником ефективності моделі, але лише тоді, коли для навчання моделі використовуються симетричні набори даних, де кількість класів майже однакова, у інших випадках, ефективність моделі необхідно оцінювати також з використанням інших показників ефективності.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (2.47)$$

Точність (Precision) – це відношення правильно передбачених позитивних спостережень до загальної кількості прогнозованих позитивних прикладів, обчислюється за формулою 2.48.

$$Precision = \frac{TP}{TP + FP} \quad (2.48)$$

Повнота (Recall) – це відношення правильно передбачених позитивних прикладів до всіх спостережень у фактичному класі, обчислюється за формулою 2.49.

$$Recall = \frac{TP}{TP + FN} \quad (2.49)$$

Оцінка F1 – це середнє зважене значення точності та повноти, обчислюється формулою 2.50. Таким чином ця оцінка враховує як помилкові позитивні, так і помилкові негативні.

$$F1 = \frac{2(Precision \cdot Recall)}{Precision + Recall} \quad (2.50)$$

Інтуїтивно це не так легко зрозуміти, як точність, але F1 зазвичай корисніша, ніж точність, особливо, якщо модель навчається на наборі даних з нерівномірним розподілом класів.

Отже, розраховуючи дані оцінки буде визначатись точність результатів прогнозування моделей машинного навчання.

3 АЛГОРИТМ ПРОГНОЗНОГО ТЕХНІЧНОГО ОБСЛУГОВУВАННЯ

3.1 Отримання даних (Data Extraction)

З запровадженням Industry 4.0 кількість та різноманітність інформації різко зростає. Це змушує промислові виробництва сумніватись, які ознаки а дані є обов'язковими для того, щоб реалізувати прогнозне обслуговування. Щоб полегшити процес впровадження прогнозного технічного обслуговування розглянемо модель впровадження прогнозного технічного обслуговування, яка допоможе крок за кроком успішно реалізувати модель прогнозного технічного обслуговування на промисловому підприємстві. На рисунку 3.1 зображено алгоритм для впровадження прогнозного технічного обслуговування, яка поділена на дві фази: отримання даних (Data Extraction) та прогнозування даних (Data Prediction).

Етап отримання даних дозволяє користувачам отримати доступ і створити основні структури, необхідні для впровадження прогнозного обслуговування. Ця модель, поділена на три основні кроки, починається з оцінки ситуації (рисунок 3.1). Обов'язковою є оцінка основних питань і структур, що вимагають додаткової уваги на основі традиційних методологій технічного обслуговування. Цей крок дозволяє зрозуміти, які є можливості, як з точки зору отримання та накопичення даних (чи доступні необхідні дані?), так і з точки зору оцінки ризиків та прибутків (якщо ресурси обмежені).

У можливому сценарії, коли не всі необхідні дані доступні для проведення аналізу, пропонується встановлення додаткових датчиків для збирання необхідної інформації.

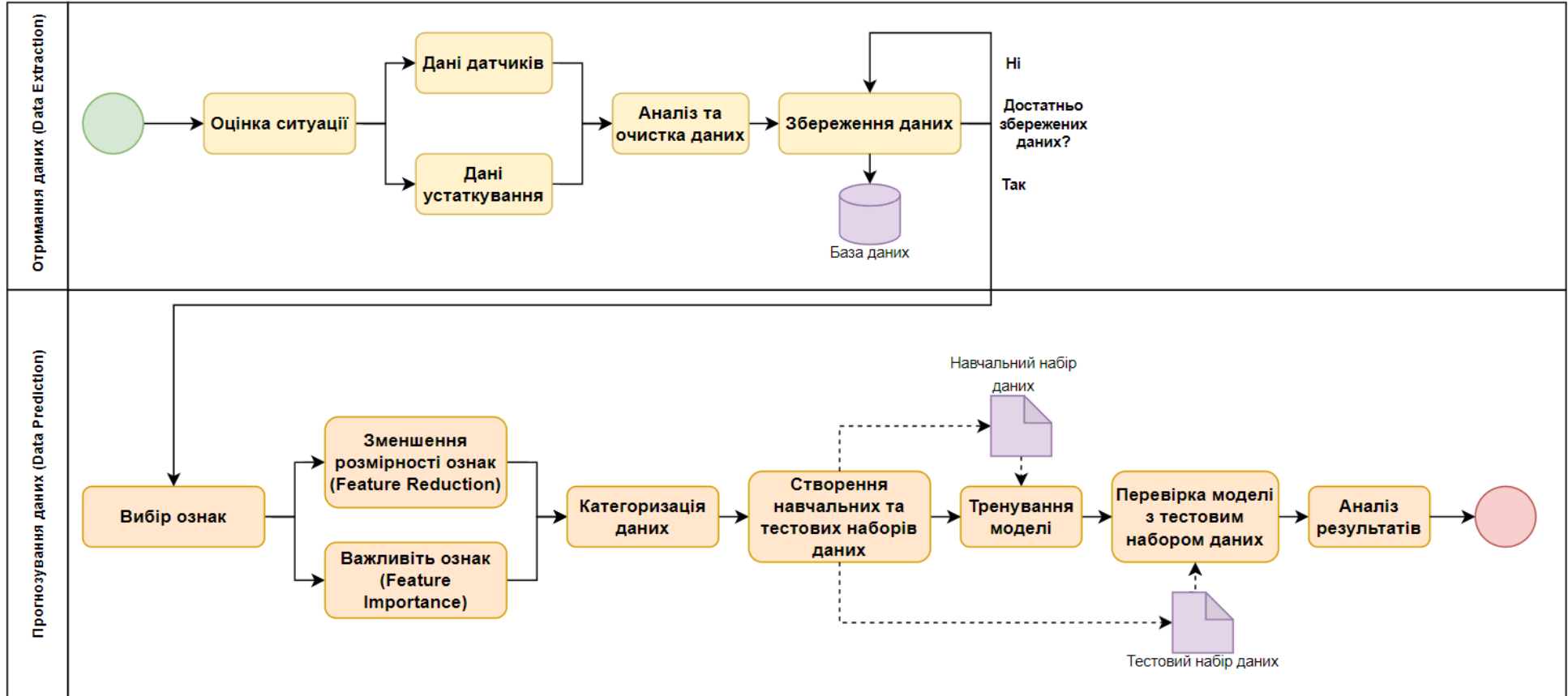


Рисунок 3.1 – Алгоритм прогнозного технічного обслуговування

Здійснюючи вилучення та аналіз даних, можна отримати надійний набір даних, який буде використовуватись для методологій прогнозного обслуговування. Однак одного вилучення в більшості випадків недостатньо для виконання цих методологій, оскільки набір можливих поведінок устаткування невеликий, а моделі класифікації безпосередньо опираються на історичні дані. Тому навчання моделей з великим об'ємом вхідних даних дозволяє отримати більш точні результати. Цю передумову можна виконати за допомогою регулярного вилучення, аналізу та збереження даних (рисунок 3.1) в базі даних або озері даних [41], в залежності від об'ємів даних. Розуміння обсягу даних, необхідних для аналізу даних є складним питанням, оскільки більшість випадків прогнозного технічного обслуговування потребує великого проміжку часу. На цьому закінчується перший етап моделі прогнозного технічного обслуговування.

3.2 Прогнозування даних (Data Prediction)

Другим етапом моделі прогнозного технічного обслуговування є прогнозування даних, що стосується аналітичного та практичного використання даних отриманих на першому етапі. Цей етап надає можливість безпосереднього доступу на аналізу даних для отримання корисних знань, які можуть бути використані в процесі прийняття рішень.

Першим рекомендованим кроком є виконання вибору ознак (рисунок 3.1), дуже важливого кроку в процесі реалізації прогнозного обслуговування, оскільки вибрані ознаки відіграють важливу роль у кінцевих результатах моделей.

Оскільки функції відіграють таку важливу роль у роботі моделей класифікації, наведемо приклад загальних і специфічних ознак: загальні ознаки (вібрація, напруга, тиск, кількість обертів, температура, поломки обладнання); специфічні ознаки (час використання обладнання, загальні відомості про обладнання, положення обладнання). Слід брати до уваги, що перелік ознак не може охопити широкий спектр усіх секторів галузей, наведений вище набір демонструє приклад загальних і специфічних ознак для кращого розуміння. Список

специфічних ознак стосується тих, які не використовуються більше ніж в одному варіанті використання. Їх слід розглядати за суворими сценаріями, якщо вони доступні і актуальні для аналізу, наприклад, тип обладнання або його положення в процесі. Список загальних ознак, як впливає з назви, стосується ознак, які використовуються в технологічному процесі та пов'язані з часом, наприклад, температура, вібрації, показники двигуна та інші.

Коли вибір ознак буде завершено, наступним запропонованим кроком, є категоризація даних (рисунок 3.1). Цей крок загалом важливий з точки зору класифікації вихідних даних, а точніше з точки зору нечислових даних. Можливість перетворення нечислових даних в текстові дає користувачеві доступ до більш широкого набору можливостей. Однією з основних переваг категоризації даних є можливість зменшити різноманітність потоків даних, які вводяться в модель.

Після завершення категоризації наступним кроком є підготовка наборів даних для моделі (рисунок 3.1 – б). Підготовка набору даних виконується шляхом створення двох наборів даних, для тренування моделі (70-80% від набору даних) та для тестування моделі (20-30% від набору даних). Основна відмінність між цими двома наборами даних стосується лише вихідної змінної, оскільки тестовий набір даних не містить вихідних даних, які слід передбачити. У звичайних сценаріях розділення вихідного набору даних, як правило, сприяє більш високому відсотку даних у навчальному наборі.

Підготовка моделі виконується зі згенерованими наборами даних. Спочатку модель необхідно навчити, використання навчального набору даних дозволяє побудувати модель і підготувати її до умов, для яких бажаний результат визначається вихідними змінними навчального набору даних. Використання тестового набору даних дозволяє оцінити продуктивність моделі на невидимих даних, оскільки існує можливість того, що модель добре навчається але має погані результати ефективності на невидимих даних. Дорожня карта немає методології для вибору самого алгоритму машинного навчання, оскільки вибір алгоритму може залежати від великої кількості факторів та бажаних результатів.

Наступним кроком в реалізації прогнозного обслуговування є аналіз результатів (рисунок 3.1). Аналіз результатів моделі виконується шляхом аналізу параметрів, отриманих при виконанні моделі з тестовим набором даних. Зазвичай основним параметром оцінювання моделі є точність, яка стосується співвідношення правильних випадків у загальній кількості випадків, яка є легкою для розуміння та широко використовуваною метрикою оцінки моделі машинного навчання. Для перевірки точності моделей рекомендується використовувати наступні показники: precision, recall, f1-score, які розраховуються за допомогою матриці невідповідностей.

Модель, що наведена на рисунку 3.1 також може розширюватись додатковими кроками, якщо цього потребує задача, для досягнення більш ефективного впровадження технічного обслуговування, що базується на прогнозуванні несправностей обладнання з використанням методів машинного навчання.

Після виконання всіх кроків, ми отримуємо модель, яка в подальшому може використовуватись для прогнозного обслуговування. Якщо отримана модель не має бажаних показників якості, слід повторити етап прогнозування даних з використанням інших підходів до аналізу даних або з використанням інших методів машинного навчання, поки не буде отримана модель з бажаними показниками точності.

Дана модель може використовуватись для різних галузей промисловості та обладнання різних типів, тому що модель зосереджена не на конкретному обладнанні певної галузі, а на загальних кроках для отримання моделі на основі певного методу машинного навчання.

4 ЗАСТОСУВАННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ПРОГНОЗУВАННЯ НЕСПРАВНОСТЕЙ ОБЛАДНАННЯ

4.1 Вибір технічних засобів

Для моделювання методів машинного навчання буде використано наступні технічні засоби:

1. Python – мова програмування, що відмінно підходить для роботи з даними та методами машинного навчання;
2. Jupyter Notebook – інтерактивна платформа, для роботи з даними та обчисленнями (підтримує мову Python);
3. Pandas – бібліотека мови Python для роботи з наборами даних;
4. Matplotlib – бібліотека для візуалізації даних;
5. Scikit – бібліотека для роботи з методами машинного навчання;

4.2 Характеристика набору даних

4.2.1 Вихідні дані обладнання

Першим кроком, для реалізації прогнозного обслуговування є отримання даних, як описано в розділі 3. Це включає аналіз та очищення даних. Джерела даних для прогнозного обслуговування зазвичай включають наступні: інформація про обладнання (наприклад, розмір двигуна, марка, модель); вихідні дані обладнання (наприклад, дані датчиків, такі як температура, тиск, вібрація, швидкість); історія технічного обслуговування (історія ремонту обладнання та журнали з зазначеним часом технічного обслуговування); історія відмов (історія відмов обладнання, компонентів).

Щоб передбачити невдачі, дані повинні містити як приклади невдачі так і приклади нормальної роботи обладнання. Велика кількість прикладів призведе до більш кращих та точних результатів моделі. Також важливо мати дані з пристроїв, що вийшли з ладу через конкретну проблему яка нас цікавить, і з обладнання, що

вийшло з ладу з інших причин. В обох випадках чим більше даних у нас є, тим краще рішення [42].

З кожним роком у промисловому середовищі з'являється все більше відкритих наборів даних. Для реалізації рішення прогнозованого технічного обслуговування будемо використовувати набір даних, що був опублікований компанією Microsoft, який включає промислові дані за один рік, з першого січня 2015 року до першого січня 2016 року [43]. Набір даних містить інформацію з п'яти джерел: вихідні дані обладнання (показники датчиків), журнал помилок, історія технічного обслуговування, історія несправностей та відомості про обладнання.

Вихідні дані обладнання (показники датчиків виробництва) містять 876 100 записів для 100 одиниць обладнання включаючи наступні дані: ідентифікатор обладнання (`machineID`), дата (`datetime`), напруга (`volt`), кількість обертів (`rotate`), тиск (`pressure`), вібрація (`vibration`). Показники датчиків містять дані з першого січня 2015 року до першого січня 2016 року. Для відображення поведінки даних на рисунку 4.1 зображено статистичну характеристику даних, де відображено кількість записів, середнє та стандартне відхилення, а також мінімальне та максимальне значення параметрів. На рисунку 4.2 відображено дані з показами датчиків з набору даних.

	<code>machineID</code>	<code>volt</code>	<code>rotate</code>	<code>pressure</code>	<code>vibration</code>
count	876100.000000	876100.000000	876100.000000	876100.000000	876100.000000
mean	50.500000	170.777736	446.605119	100.858668	40.385007
std	28.866087	15.509114	52.673886	11.048679	5.370361
min	1.000000	97.333604	138.432075	51.237106	14.877054
25%	25.750000	160.304927	412.305714	93.498181	36.777299
50%	50.500000	170.607338	447.558150	100.425559	40.237247
75%	75.250000	181.004493	482.176600	107.555231	43.784938
max	100.000000	255.124717	695.020984	185.951998	76.791072

Рисунок 4.1 – Статистичний аналіз показників датчиків

	datetime	machineID	volt	rotate	pressure	vibration
0	2015-01-01 06:00:00	1	176.217853	418.504078	113.077935	45.087686
1	2015-01-01 07:00:00	1	162.879223	402.747490	95.460525	43.413973
2	2015-01-01 08:00:00	1	170.989902	527.349825	75.237905	34.178847
3	2015-01-01 09:00:00	1	162.462833	346.149335	109.248561	41.122144
4	2015-01-01 10:00:00	1	157.610021	435.376873	111.886648	25.990511

Рисунок 4.2 – Покази датчиків з набору даних

На рисунку 4.3 зображено ядрову оцінку густини розподілу даних, а саме тиску, кількості обертів, вібрації та напруги.

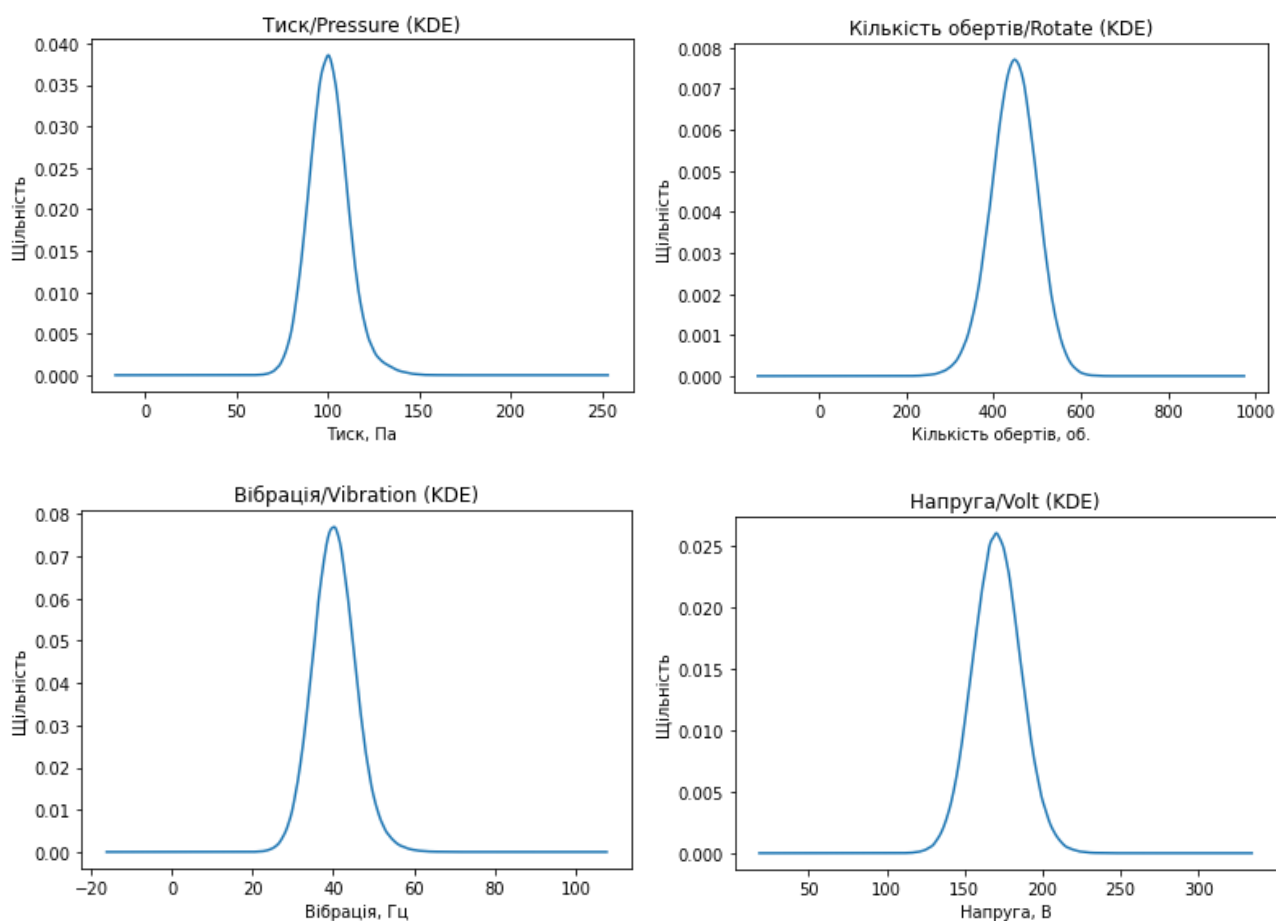


Рисунок 4.3 – Ядрова оцінка густини розподілу даних датчиків

На рисунках 4.4, 4.5, 4.6, 4.7 відображено дані з датчиків для обладнання з ідентифікатором 1 ($machineID = 1$), тиску, кількості обертів, вібрації та напруги відповідно.

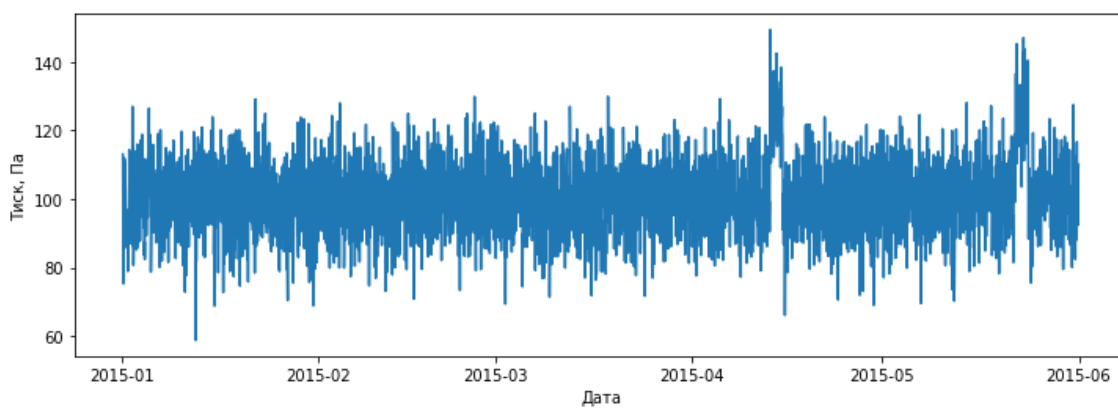


Рисунок 4.4 – Дані датчиків тиску для обладнання з ідентифікатором 1

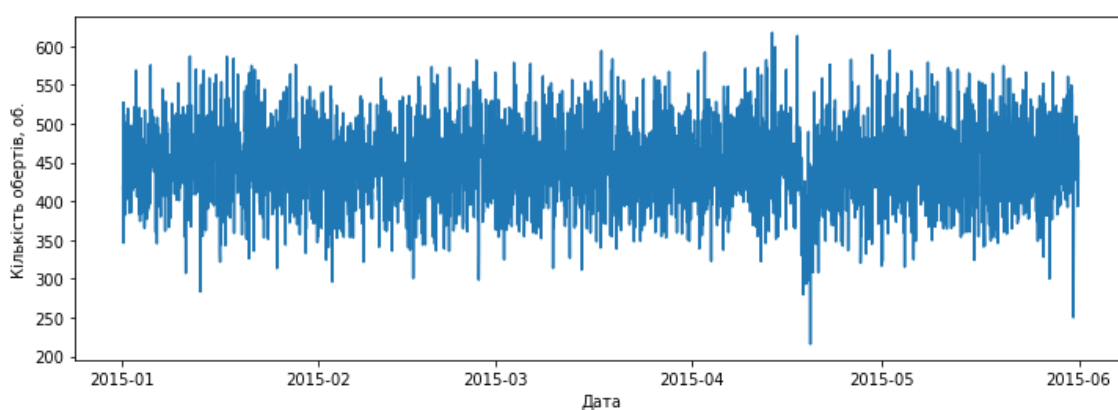


Рисунок 4.5 – Дані датчиків кількості обертів для обладнання з ідентифікатором 1

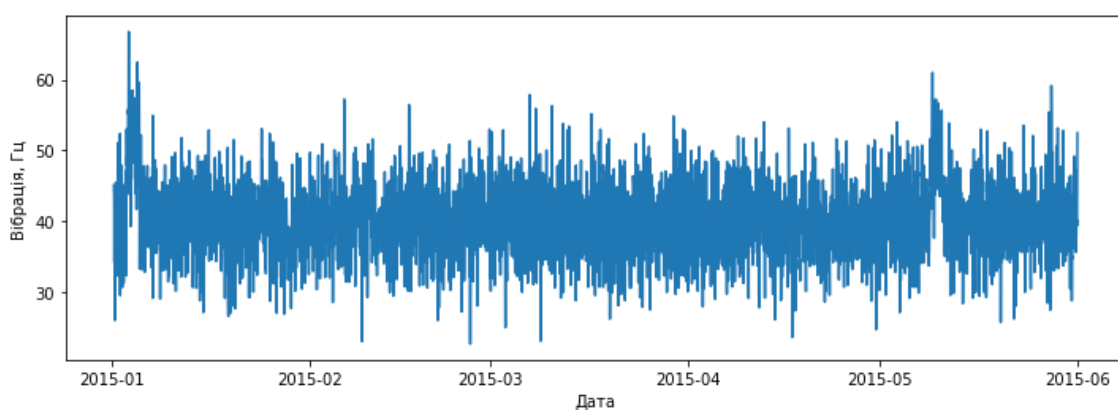


Рисунок 4.6 – Дані датчиків вібрації для обладнання з ідентифікатором 1

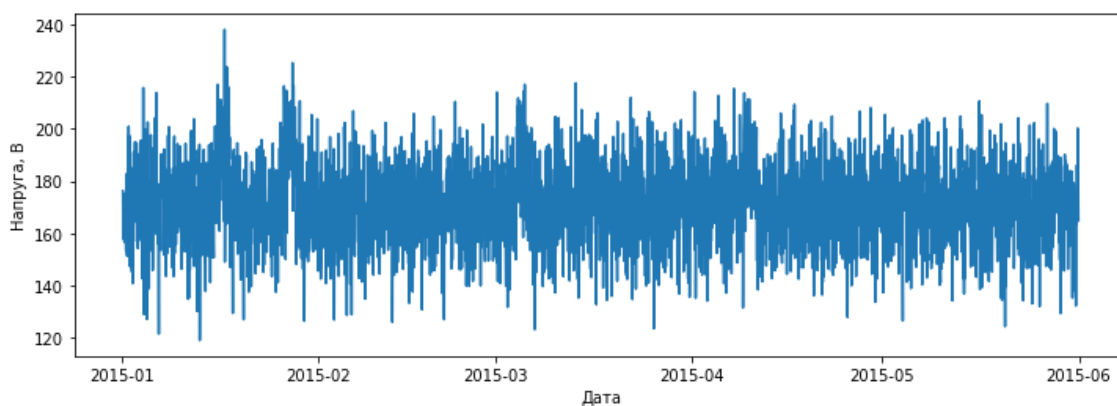


Рисунок 4.7 – Дані датчиків напруги для обладнання з ідентифікатором 1

4.2.2 Журнал помилок обладнання

Журнал помилок обладнання містить 3919 записів і включає наступну інформацію: дата (datetime), ідентифікатор обладнання (machineID), ідентифікатор поломки (errorID). В журналі помилок існує п'ять типів помилок: error1, error2, error3, error4, error5. На рисунку 4.8 показано приклад записів в журналі помилок для обладнання з ідентифікатором 1.

	datetime	machineID	errorID
0	2015-01-03 07:00:00	1	error1
1	2015-01-03 20:00:00	1	error3
2	2015-01-04 06:00:00	1	error5
3	2015-01-10 15:00:00	1	error4
4	2015-01-22 10:00:00	1	error4

Рисунок 4.8 – Дані про помилки для обладнання з ідентифікатором 1

На рисунку 4.9 зображено кількість помилок за типом помилки, на даному рисунку можна побачити, що найбільш частою помилкою для обладнання є помилка з ідентифікатором error1.

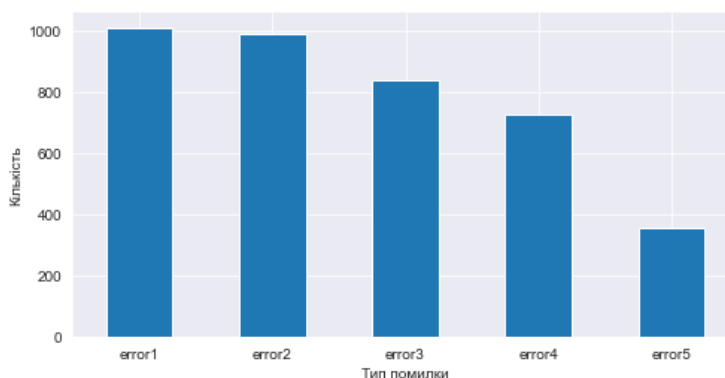


Рисунок 4.9 – Зображення кількості помилок за типом помилки

4.2.3 Історія технічного обслуговування

Історія технічного обслуговування містить 3286 записів та включає наступну інформацію: дата (datetime), ідентифікатор обладнання (machineID), назва компонента (comp) для якого відбувалось технічне обслуговування. В історії технічного обслуговування наведено дані про обслуговування чотирьох компонентів обладнання: comp1, comp2, comp3, comp4. На рисунку 4.10 наведено приклад записів для технічного обслуговування обладнання з ідентифікатором 1.

	datetime	machineID	comp
0	2014-06-01 06:00:00	1	comp2
1	2014-07-16 06:00:00	1	comp4
2	2014-07-31 06:00:00	1	comp3
3	2014-12-13 06:00:00	1	comp1
4	2015-01-05 06:00:00	1	comp4

Рисунок 4.10 – Дані про технічне обслуговування для обладнання з ідентифікатором 1

На рисунку 4.11 зображено кількість технічного обслуговування за типом компонента. На даному рисунку можна побачити, що найчастіше технічне обслуговування здійснювалось для компонента з ідентифікатором comp2.

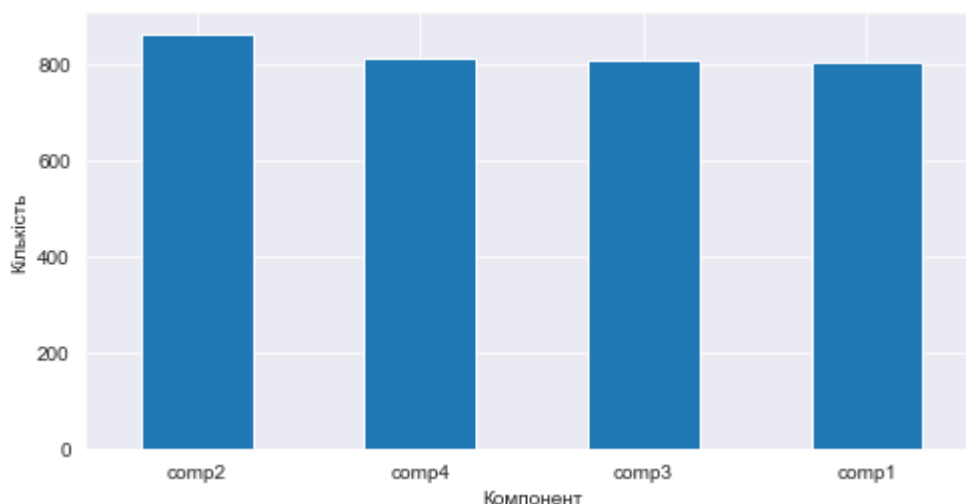


Рисунок 4.11 – Зображення кількості технічного обслуговування за типом компонента

4.2.4 Історія поломок обладнання

Історія поломок обладнання містить 761 запис і включає наступні дані: дата (datetime), ідентифікатор обладнання (machineID), назва компонента, причина поломки (failure). В причині поломки містяться дані про компонент, який зазнав поломки. На рисунку 4.12 зображено дані про поломки для обладнання з ідентифікатором 1.

	datetime	machineID	failure
0	2015-01-05 06:00:00	1	comp4
1	2015-03-06 06:00:00	1	comp1
2	2015-04-20 06:00:00	1	comp2
3	2015-06-19 06:00:00	1	comp4
4	2015-09-02 06:00:00	1	comp4

Рисунок 4.12 – Дані про поломки для обладнання з ідентифікатором 1

На рисунку 4.13 зображено кількість поломок за причиною поломки. На даному рисунку можна побачити, що найчастіше зазнає поломок обладнання через причину з ідентифікатором компонента: comp2.

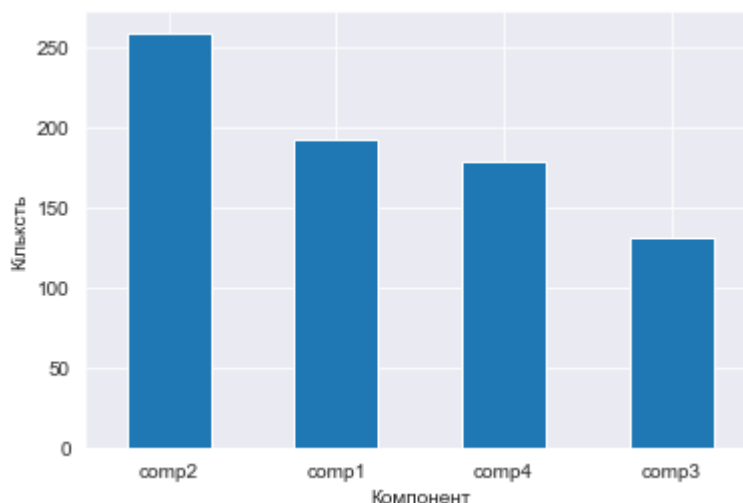


Рисунок 4.13 – Зображення кількості поломок обладнання за причиною поломки

4.2.5 Відомості про обладнання

Відомості про обладнання містять 100 записів, тобто 100 одиниць обладнання, та включають наступні дані: ідентифікатор обладнання (`machineID`), модель (`model`), термін експлуатації обладнання (`age`). На рисунку 4.14 зображено гістограму, що відображає кількість обладнання та термін експлуатації за його моделлю.

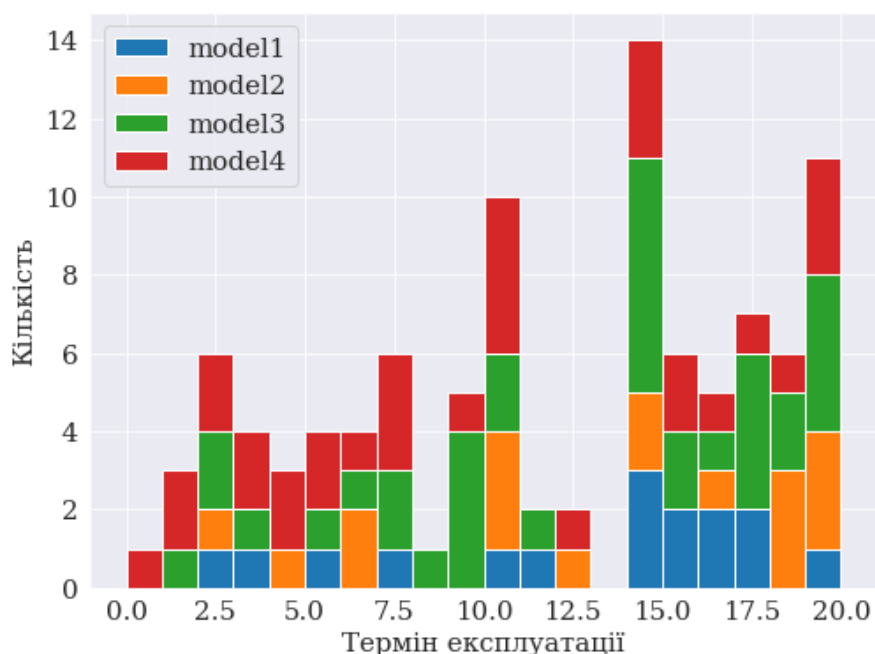


Рисунок 4.14 – Гістограма, що відображає кількість обладнання і термін експлуатації за його моделлю

4.3 Розробка ознак

Розробка ознак (Feature Engineering) – це техніка машинного навчання, яка використовує дані для створення нових змінних, яких немає в навчальному наборі. За допомогою цієї техніки можуть створюватись нові функції як для навчання з наглядом, так і для навчання без нагляду, з метою спрощення та прискорення перетворення даних, а також підвищення точності моделі. Під час роботи з моделями машинного навчання необхідна розробка ознак. Незалежно від даних чи архітектури моделі, погана ознака матиме прямий вплив на точність моделі.

Метою розробки ознак є прагнення збільшити потужність передбачення алгоритмів машинного навчання, створюючи нові ознаки з доступних даних. Як правило, спочатку здійснюється розробка ознак, а потім відбувається відбір ознак, усуваючи нерелевантні, зайві або сильно корельовані ознаки.

Різні джерела інформації, що були розглянуті в розділі 4.1 будуть об'єднані в єдиний набір даних в результаті процесу розробки ознак, та будуть використані для навчання та тестування моделей машинного навчання.

Прогнозне обслуговування передбачає поломки обладнання у майбутньому періоді часу на основі ознак, які представляють поведінку обладнання протягом історичного моменту часу. Зокрема, для показників датчиків можливі збурення, такі як шум, що ускладнюють прогнозування. Таким чином, дані можуть бути агреговані у часові вікна, що дозволяє «згладити» значення, мінімізуючи вплив шуму на ознаки, які використовуються моделями. Маючи на увазі наскільки далеко в майбутнє модель повинна бути здатна передбачити, важливо визначити, як далеко модель повинна «дивитись», щоб зробити ці передбачення. Цей проміжок часу, який проходить до моменту, коли модель повинна «оглянутися назад», називається відставанням (затримка, лаг, lag). Функції, розроблені протягом цього періоду затримки, називаються функціями відставання (lag features) [45]. Набір даних, який використовується для генерування функцій відставання, зазвичай пов'язаний з датою/часом.

Для кожного запису в наборі даних створюється часове вікно розмірності W , і ознаки відставання обчислюються для періоду W до дати/часу цього запису. На рисунку 4.15 показано приклад функцій відставання для вимірювання t_i з $W = 3$. Значення W зазвичай виражається в хвилинах або в годинах, в залежності від природи даних.

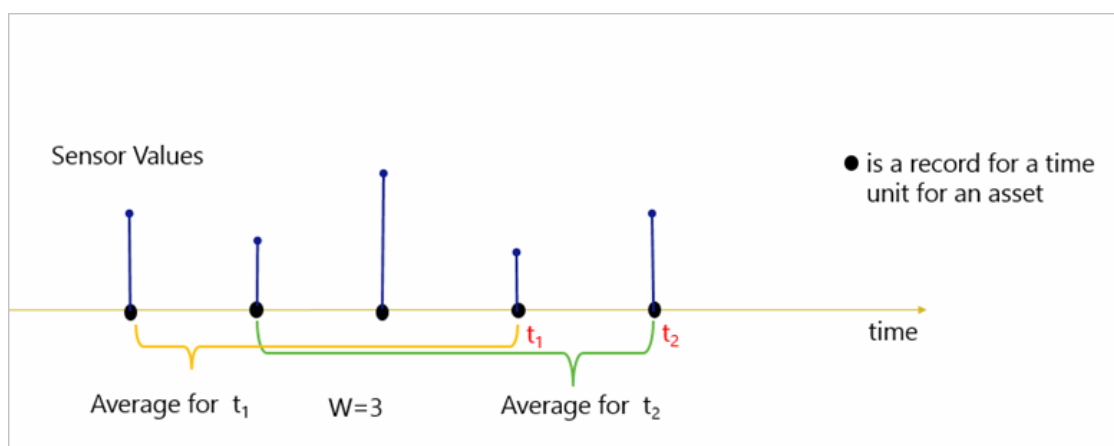


Рисунок 4.15 – Ознаки/Функції відставання

Таким чином було створено два тимчасових вікна. Перше, тривалістю 3 години, щоб відобразити поведінку даних в короткостроковій перспективі (рисунок 4.16, 4.17), друге, тривалістю 24 години, щоб представити довгострокову перспективу даних (рисунок 4.18, 4.19). У кожному з цих проміжків часу обчислюється два нових параметри: ковзне середнє (mean) і стандартне відхилення (std). Слід звернути увагу, що для проміжку $W = 24$, два нових параметри недоступні для початкових моментів (перші 24 години).

	machineID	datetime	voltmean_3h	rotatemean_3h	pressuremean_3h	vibrationmean_3h
0	1	2015-01-01 09:00:00	170.028993	449.533798	94.592122	40.893502
1	1	2015-01-01 12:00:00	164.192565	403.949857	105.687417	34.255891
2	1	2015-01-01 15:00:00	168.134445	435.781707	107.793709	41.239405
3	1	2015-01-01 18:00:00	165.514453	430.472823	101.703289	40.373739
4	1	2015-01-01 21:00:00	168.809347	437.111120	90.911060	41.738542

Рисунок 4.16 – Приклад ознак відставання для даних датчиків з $W = 3$ (mean)

machineID	datetime	voltstd_3h	rotatedstd_3h	pressuresd_3h	vibrationsd_3h
0	1 2015-01-01 09:00:00	6.721032	67.849599	18.934956	5.874970
1	1 2015-01-01 12:00:00	7.596570	50.120452	8.555032	7.662229
2	1 2015-01-01 15:00:00	10.124584	55.084734	5.909721	5.169304
3	1 2015-01-01 18:00:00	4.673269	42.047278	4.554047	2.106108
4	1 2015-01-01 21:00:00	14.752132	47.048609	4.244158	2.207884

Рисунок 4.17 – Приклад ознак відставання для даних датчиків з $W = 3$ (std)

machineID	datetime	voltmean_24h	rotatemean_24h	pressuremean_24h	vibrationmean_24h
7	1 2015-01-02 06:00:00	169.733809	445.179865	96.797113	40.385160
8	1 2015-01-02 09:00:00	170.614862	446.364859	96.849785	39.736826
9	1 2015-01-02 12:00:00	169.893965	447.009407	97.715600	39.498374
10	1 2015-01-02 15:00:00	171.243444	444.233563	96.666060	40.229370
11	1 2015-01-02 18:00:00	170.792486	448.440437	95.766838	40.055214

Рисунок 4.18 – Приклад ознак відставання для даних датчиків з $W = 24$ (mean)

machineID	datetime	voltstd_24h	rotatedstd_24h	pressuresd_24h	vibrationsd_24h
7	1 2015-01-02 06:00:00	11.233120	48.717395	10.079880	5.853209
8	1 2015-01-02 09:00:00	12.519402	48.385076	10.171540	6.163231
9	1 2015-01-02 12:00:00	13.370357	42.432317	9.471669	6.195076
10	1 2015-01-02 15:00:00	13.299281	41.346121	8.731229	5.687944
11	1 2015-01-02 18:00:00	13.954518	43.490234	8.061653	5.898069
12	1 2015-01-02 21:00:00	14.402740	42.626186	10.408012	5.941890
13	1 2015-01-03 00:00:00	15.513819	40.395881	10.833294	5.737671
14	1 2015-01-03 03:00:00	15.726970	39.648116	11.904700	5.601191
15	1 2015-01-03 06:00:00	15.635083	41.828592	11.326412	5.583521
16	1 2015-01-03 09:00:00	13.995465	40.843882	11.036546	5.561553

Рисунок 4.19 – Приклад ознак відставання для даних датчиків з $W = 24$ (std)

Як і з даними датчиків, журнал помилок також має дату/час, пов'язаний з ними. Однак ці дані категоричні, а не числові. У цьому випадку кількість помилок кожного типу додається кожні три години для вікна $W = 24$ (рисунок 4.20). Кожен рядок в наборі даних представляє суму кількості помилок кожного типу за 24 години до вказаної дати та часу.

	machineID	datetime	error1count	error2count	error3count	error4count	error5count
7	1	2015-01-02 06:00:00	0.0	0.0	0.0	0.0	0.0
8	1	2015-01-02 09:00:00	0.0	0.0	0.0	0.0	0.0
9	1	2015-01-02 12:00:00	0.0	0.0	0.0	0.0	0.0
10	1	2015-01-02 15:00:00	0.0	0.0	0.0	0.0	0.0
11	1	2015-01-02 18:00:00	0.0	0.0	0.0	0.0	0.0

Рисунок 4.20 – Приклад ознак відставання для журналу помилок обладнання

Журнал технічного обслуговування, який містить інформацію, пов'язану із заміною компонентів, дозволяє генерувати дані, які містять інформацію про термін з останньої заміни обладнання (рисунок 4.21). Очікується, що ця функція добре пов'язана з можливими поломками компонентів, оскільки, чим довший час використання компонента, тим більш очікувана його поломка.

	datetime	machineID	comp1	comp2	comp3	comp4
0	2015-01-01 06:00:00	1	16436.250000	16436.250000	16436.250000	16436.250000
1	2015-01-01 07:00:00	1	16436.291667	16436.291667	16436.291667	16436.291667
2	2015-01-01 08:00:00	1	16436.333333	16436.333333	16436.333333	16436.333333
3	2015-01-01 09:00:00	1	16436.375000	16436.375000	16436.375000	16436.375000
4	2015-01-01 10:00:00	1	16436.416667	16436.416667	16436.416667	16436.416667

Рисунок 4.21 – Приклад ознак відставання для журналу технічного обслуговування

На рисунку 4.22 відображено результати лінійного аналізу кореляції між ознаками. Коефіцієнт кореляції коливається від -1 до 1. Цей коефіцієнт дає змогу побачити, чи одна змінна виправдовує лінійну змінну іншої. Коли він близький до 1, це означає, що існує сильна позитивна кореляція, тобто якщо ознака А збільшується, то ознака Б також збільшується, а якщо ознака А зменшується, то ознака Б також зменшується. У нашому випадку виявляється, що кореляція між ознаками переважно низька (коефіцієнт кореляції близький до нуля).

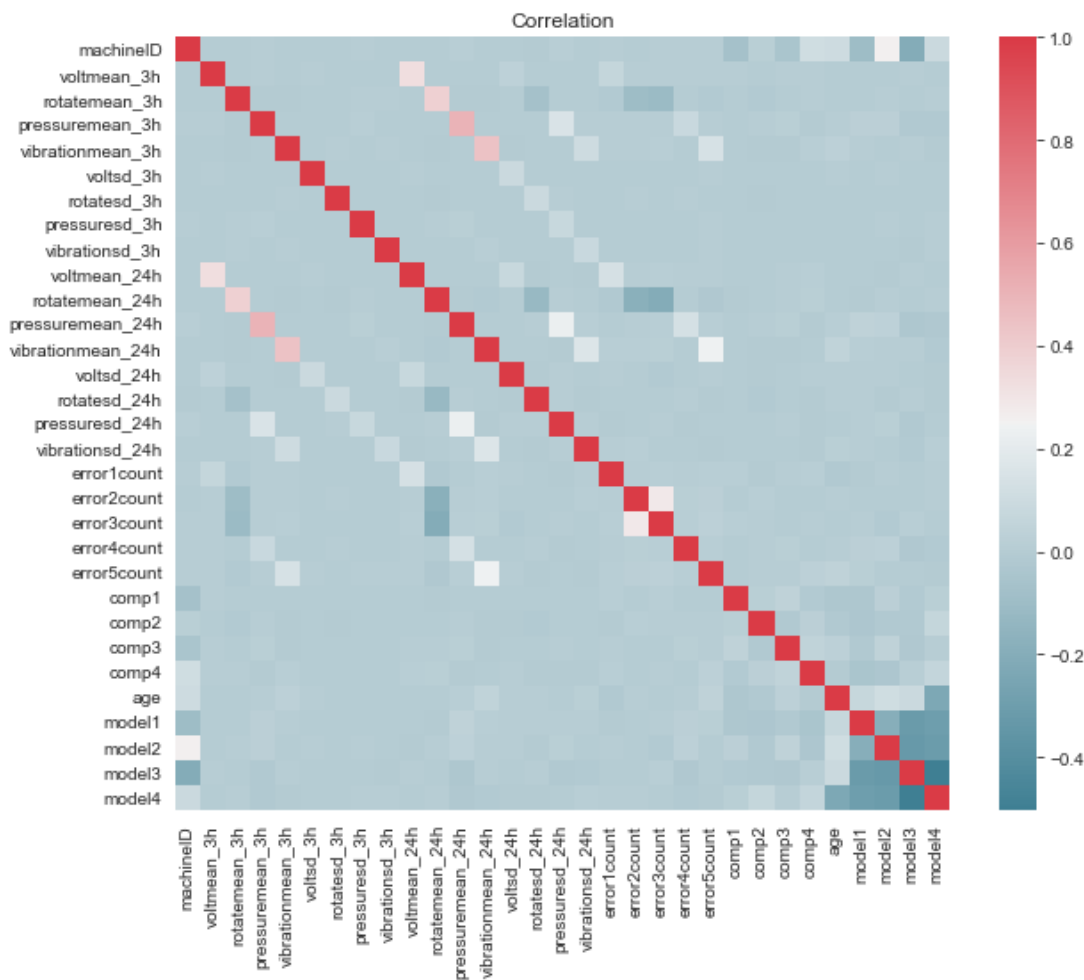


Рисунок 4.22 –Аналіз кореляції ознак

Як згадувалось раніше, проблема прогнозного обслуговування є випадком навчання з наглядом. Щоб навчити модель передбачати збої, потрібні не лише приклади невдач, а й дані, які призвели до цієї невдачі. Крім того, моделі потрібні дані «нормальної» роботи обладнання, щоб вона могла «побачити» різницю між станами обладнання.

Основною метою використовуваних моделей машинного навчання буде передбачити ймовірність збою, що станеться протягом наступних 24-х годин. Таким чином буде додана нова категоріальна ознака «відмова» («failure»), де всі дані до виникнення збою компонента з ідентифікатором 1, будуть мати запис comp1, і так само для компонентів 2, 3, 4. Дані, що не мають жодних збоїв, будуть мати позначення «none».

В результаті розробки ознак було отримано наступні ознаки: machineID, datetime, voltmean_3h, rotatemean_3h, pressuremean_3h, vibrationmean_3h, voltstd_3h, rotatestd_3h, pressurestd_3h, vibrationstd_3h, voltmean_24h, rotatemean_24h, pressuremean_24h, vibrationmean_24h, voltstd_24h, rotatestd_24h, pressurestd_24h, vibrationstd_24h, error1count, error2count, error3count, error4count, error5count, comp1, comp2, comp3, comp4, age, model1, model2, model3, model4, failure.

4.4 Застосування методів машинного навчання

Оскільки для реалізації прогнозного обслуговування необхідно вирішити завдання багатокласової класифікації, для вирішення цієї задачі були обрані наступні методи машинного навчання: випадковий ліс, нейронна мережа, градієнтний бустинг, наївний класифікатор Байєса, knn.

Для навчання та тестування моделі машинного навчання, набір даних було розділено на навчальний – з 1 січня 2015 року по 30 вересня 2015 року, 217132 записів, та тестовий з 1 жовтня 2015 року до 1 січня 2016 року, 73909 записів.

Найкращі результати мають наступні моделі машинного навчання: нейронна мережа, випадковий ліс, градієнтний бустинг. На рисунку 4.23 наведено матрицю невідповідностей для нейронної мережі, та на рисунку 4.24 – результати класифікації.

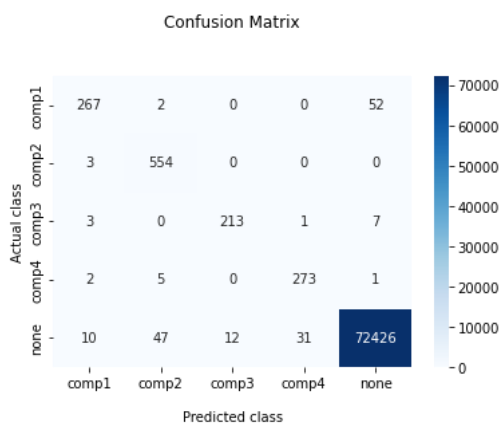


Рисунок 4.23 – Матриця невідповідностей для нейронної мережі

	precision	recall	f1-score	support
comp1	0.94	0.83	0.88	321
comp2	0.91	0.99	0.95	557
comp3	0.95	0.95	0.95	224
comp4	0.90	0.97	0.93	281
none	1.00	1.00	1.00	72526
accuracy			1.00	73909
macro avg	0.94	0.95	0.94	73909
weighted avg	1.00	1.00	1.00	73909

Рисунок 4.24 – Результати класифікації нейронної мережі

На рисунку 4.25 наведено матрицю невідповідностей для випадкового лісу, на рисунку 4.26 – важливість ознак, на рисунку 4.27 – результати класифікації.

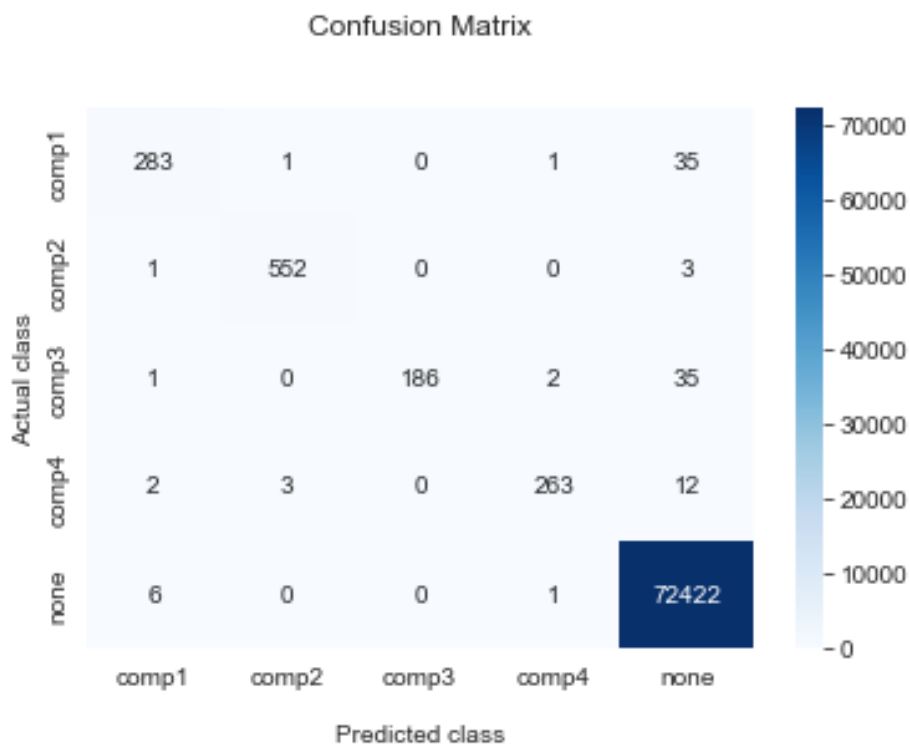


Рисунок 4.25 – Матриця невідповідностей для випадкового лісу

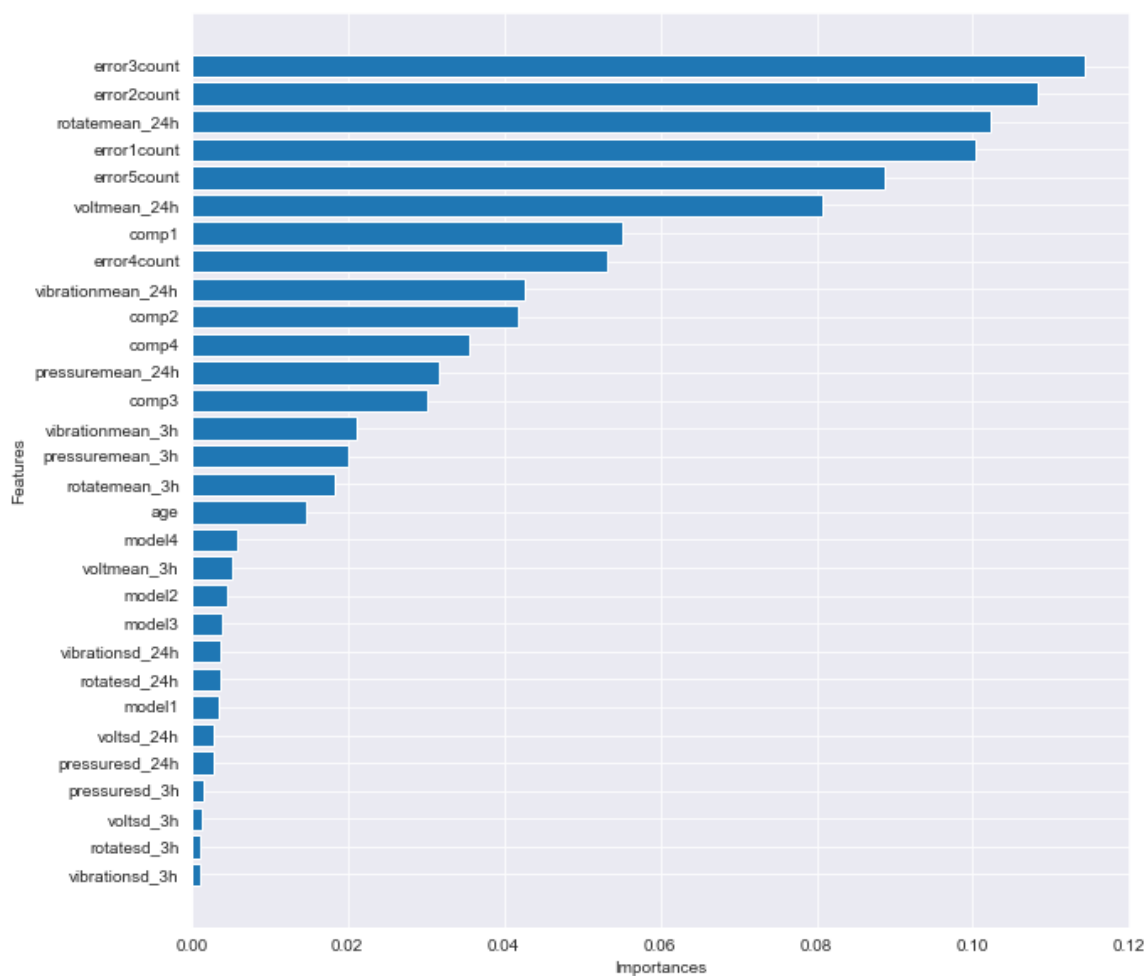


Рисунок 4.26 – Важливість ознак (Feature Importance)

	precision	recall	f1-score	support
comp1	0.97	0.88	0.92	320
comp2	0.99	0.99	0.99	556
comp3	1.00	0.83	0.91	224
comp4	0.99	0.94	0.96	280
none	1.00	1.00	1.00	72429
accuracy			1.00	73809
macro avg	0.99	0.93	0.96	73809
weighted avg	1.00	1.00	1.00	73809

Рисунок 4.27 – Результати класифікації випадкового лісу

На рисунку 4.28 – наведено матрицю невідповідностей для градієнтного бустингу, на рисунку 4.29 – результати класифікації.

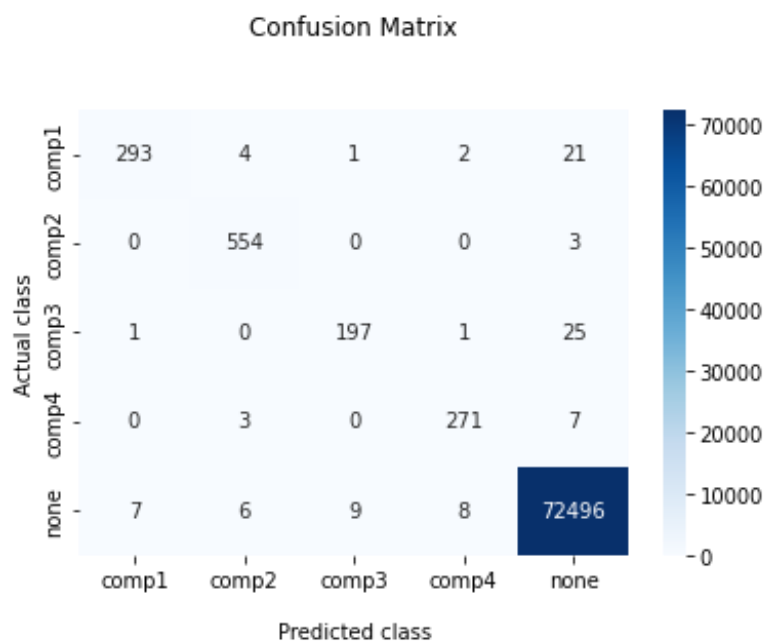


Рисунок 4.28 – Матриця невідповідностей градієнтного бустингу

	precision	recall	f1-score	support
comp1	0.97	0.91	0.94	321
comp2	0.98	0.99	0.99	557
comp3	0.95	0.88	0.91	224
comp4	0.96	0.96	0.96	281
none	1.00	1.00	1.00	72526
accuracy			1.00	73909
macro avg	0.97	0.95	0.96	73909
weighted avg	1.00	1.00	1.00	73909

Рисунок 4.29 – Результати класифікації градієнтного бустингу

На основі отриманих результатів класифікації методів машинного навчання можна зробити висновок, що методи, що мали найкращі результати класифікації, а саме, випадковий ліс, нейронна мережа, градієнтний бустинг, доцільно використовувати для багатокласової класифікації для прогнозного технічного обслуговування. Враховуючи, що при вирішенні завдання прогнозного технічного обслуговування ми стикаємось з проблемою нерівномірного розподілу класів, оскільки обладнання на промислових підприємствах рідко зазнає поломок і більше

даних в навчальних та тестових наборах даних відповідають за нормальну роботу обладнання, ми отримали хороші результати точності класифікації. На рисунках 4.24, 4.27, 4.29 можна побачити, що значення precision, recall, f1 досягають значень більших 0.9-1, що є чудовими показниками точності моделей.

4.5 Архітектура програмного забезпечення для використання моделі машинного навчання для прогнозного обслуговування

Постачальник хмарних послуг (наприклад AWS, Azure, GCP) надає сервери, де клієнти хмарних послуг можуть орендувати обчислювальні можливості та розміщувати власне програмне забезпечення. Провайдери пропонують три види хмарних послуг:

- Інфраструктура як послуга (IaaS): віртуальна обчислювальна інфраструктура (віртуальні центральні процесори або мережеві компоненти);
- Платформа як послуга (PaaS): віртуальна інфраструктура з програмною платформою (віртуальні машини з операційними системами);
- Програмне забезпечення як послуга (SaaS): готове до використання програмне забезпечення (наприклад, бази даних або хмарні системи планування ресурсів підприємства ERP).

Кожна з цих послуг може бути використана клієнтом хмари, тоді як провайдер керує всіма основними аспектами та відповідає за надійність систем.

У Industry 4.0 промислові виробництва отримують користь від використання хмарних обчислень. Видимість централізованої інформації в режимі реального часу дозволяє застосовувати більш ефективні підходи для роботи з даними та інфраструктурою виробництва. Серед всіх технологій Industry 4.0, які застосовують в промисловості, хмарні обчислення мають 85% поширення. Intel і Oracle провели дослідження серед 1200 менеджерів компаній і виявили, що 60% вважають, що хмарна інфраструктура є необхідною для розкриття повного функціоналу Industry 4.0.

Перевагами використання хмарних технологій для прогнозного обслуговування є: знижені витрати, легкий доступ до даних, віддалений доступ, уніфікація інформації, автоматичні оновлення.

Однією з головних переваг використання хмарних технологій є позитивний економічний ефект. Це важливо, оскільки система не має фізичної інфраструктури. Витрати, пов'язані з обладнанням та звичайним обслуговуванням цих рішень відсутні.

Доступність є ще однією важливою перевагою системи. Це допомагає персоналу з певними правами доступу отримувати доступ до систем та даних. Персонал стане більш відповідальним, розуміючи фактичні та очікувані результати поточного операційного стану.

Можливість віддаленого доступу реалізується за рахунок незалежності від фіксованої інфраструктури для вилучення інформації, тому можна переглядати та контролювати дані в режимі реального часу, з будь якого місця та будь якого пристрою, що гарантує високий рівень безпеки даних та їх історії.

Завдяки уніфікації даних можна обробляти їх швидше та ефективніше. Це дозволить зрозуміти та інтерпретувати їх у простий спосіб, покращуючи прийняття рішень виробництвом.

Оновлення системи відбувається автоматично, що означає низькі витрати на обслуговування та безпеку.

Враховуючи можливості та розвиток хмарних технологій, їх використання є доцільним для розробки програмного забезпечення для прогнозного обслуговування. На рисунку 4.30 запропоновано архітектуру системи для прогнозного обслуговування на основі хмарних технологій, а саме провайдера хмарних технологій Amazon Web Services (AWS) [46].

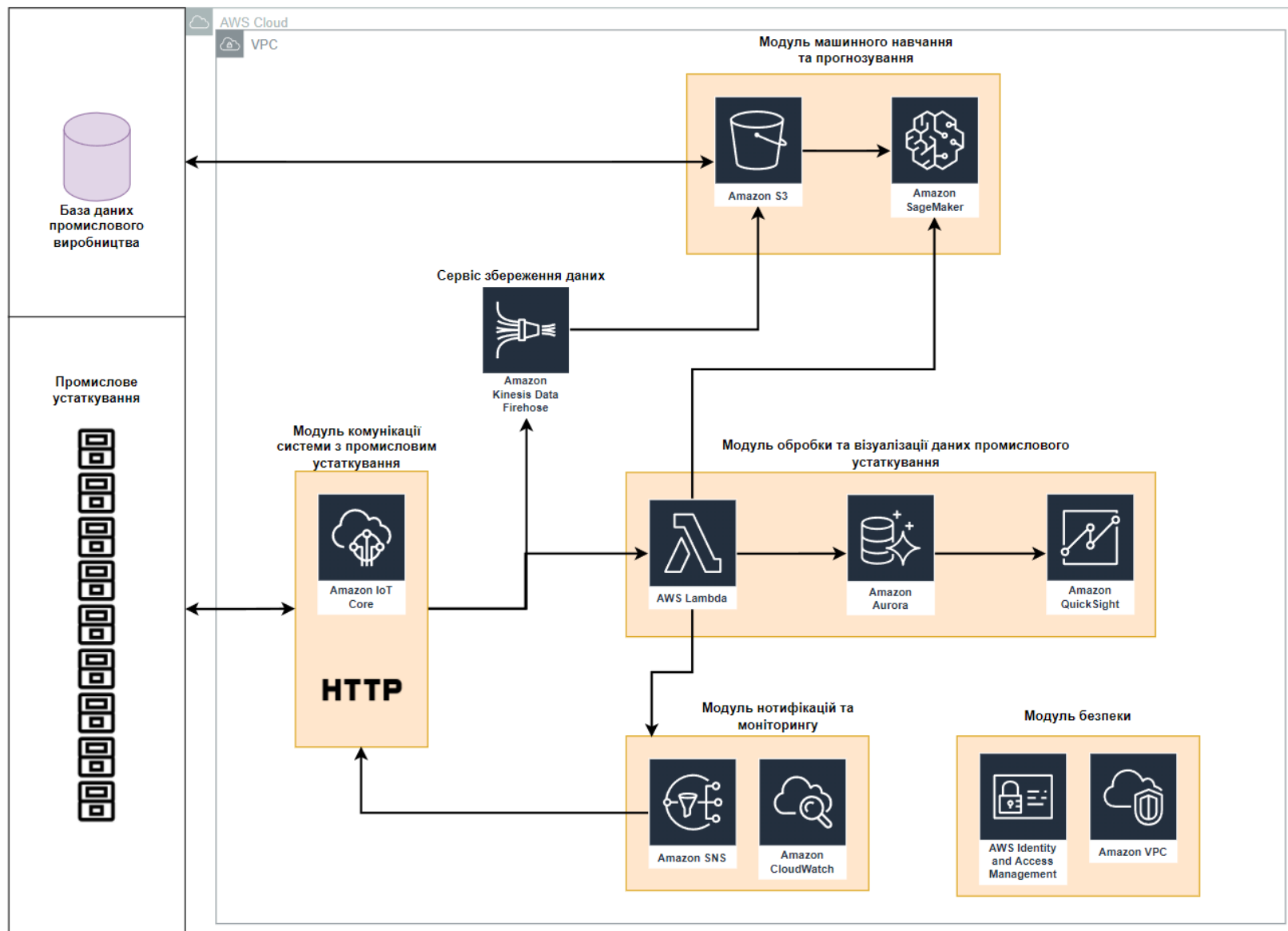


Рисунок 4.30 – Архітектура системи програмного забезпечення для прогнозного обслуговування на основі хмарних технологій

Розглянемо детальніше запропоновану архітектуру на рисунку 4.30. Оскільки в промислового виробництва на момент початку реалізації прогнозного обслуговування з використанням хмарних технологій вже можуть бути накопичені історичні дані про показники обладнання, історію технічного обслуговування, історію поломок та інше, то ці сховища даних необхідно з'єднати з хмарою. Amazon S3 [47] буде використовуватись для збереження наборів даних промислового підприємства.

Промислове обладнання взаємодіє з хмарою за допомогою HTTP запитів або сервісу Amazon IoT Core [48]. Далі дані посилаються до сховища промислових даних, використовуючи сервіс Amazon Kinesis Data Firehose [49], що дозволяє збирати, аналізувати та обробляти дані в режимі реального часу, та посилаються до сервісу AWS Lambda [50].

AWS Lambda виконує застосунок, який використовуючи модель машинного навчання, що зберігається і виконується сервісом Amazon SageMaker [51], прогнозує несправність обладнання на основі отриманих даних з промислового обладнання. Результати прогнозів зберігаються в базу даних – AWS Aurora DB [52].

Агреговані результати прогнозів відображаються за допомогою сервісу Amazon QuickSight Dashboard [53]. Результати прогнозів також надходять до сервісу сповіщень Amazon SNS [54], який в свою чергу посилає сповіщення про результати прогнозування несправностей всім своїм підписникам. Також для результатів прогнозування та для системи налаштовуються сигналізація за допомогою сервісу Amazon Cloudwatch [55], для сповіщень про результати прогнозування та роботи системи в цілому. За допомогою даних сповіщень можна отримувати інформацію про поточний стан системи, працездатність застосунків, використання ресурсів, тощо.

За допомогою сервісу Amazon VPC [56] налаштовується мережева безпека інфраструктури, де налаштовуються доступи між застосунками та мережевий доступ до хмарної інфраструктури в цілому.

Одним з найважливіших для безпеки сервісів є – Amazon Identity and Access Management [57], за допомогою нього налаштовуються ролі користувачі та застосунків в системі, надаються відповідні доступи до систем користувачам.

Для простори створення та розгортання програмного забезпечення на основі хмарних сервісів AWS пропонується використовувати AWS CloudFormation [58] або Terraform [59].

AWS CloudFormation дозволяє моделювати та розподіляти ресурси AWS та сторонні ресурси, а також керувати ними завдяки підходу «Infrastructure as a Code». Даний сервіс дозволяє описувати інфраструктуру AWS за допомогою файлів формату JSON, YAML. У файлах описуються ресурси та залежності між ними. Файли, що описують інфраструктуру можна параметризувати та використовувати в подальшому змінюючи налаштування, що забезпечує більш гнучке використання ресурсів.

Terraform є інструментом для створення та налаштування інфраструктури, що створений на основі підходу «Infrastructure as a Code» та дозволяє описувати інфраструктуру за допомогою файлів налаштувань. Terraform кодифікує хмарні API у декларативні файли конфігурацій. Terraform дозволяє використання різних провайдерів налаштувань, в той час як AWS CloudFormation підтримує налаштування тільки AWS сервісів.

Дана архітектура дозволяє реалізувати прогнозне технічне обслуговування на основі хмарних технологій використовуючи всі переваги хмарних технологій. Дану архітектуру можна масштабувати та покращувати в залежності від конкретних завдань на промисловому виробництві. Для більш ефективного використання ресурсів рекомендується використовувати підхід «Infrastructure as a Code» для створення та розгортання інфраструктури за допомогою сервісів AWS CloudFormation або Terraform.

ВИСНОВКИ

В магістерській дисертації було досліджено технічне обслуговування на основі технологій Індустрії 4.0, а саме, методів машинного навчання, хмарних обчислень, великих даних та аналітиці.

Оскільки технології Індустрії 4.0 дуже стрімко розвиваються, а промислові підприємства все більше масштабуються, вони потребують кращих рішень для покращення ефективності промислових процесів для підвищення прибутків.

Одним з ключових напрямів для покращення ефективності промислових підприємств є технічне обслуговування промислового устаткування.

В першому розділі – «Наукова проблематика. Завдання прогностичного технічного обслуговування». Було проаналізовано сучасні підходи до технічного обслуговування обладнання. Існуючі методи технічного обслуговування, такі як, реактивне обслуговування (обслуговування здійснюється при поломці обладнання), профілактичне обслуговування (обслуговування здійснюється за чітко спланованим графіком) дуже активно використовуються як стратегії технічного обслуговування та поєднуються між собою для того щоб понизити ризики виходу з ладу обладнання та підвищити час експлуатації. Більш сучасним методом технічного обслуговування є прогностичне обслуговування при якому обслуговування планується за результатами прогнозів. Сучасні технології Індустрії 4.0 дозволяють нам робити промислові виробництва «розумними», що дозволяє отримувати дані від обладнання, аналізувати та використовувати їх для підвищення ефективності промисловості. Зібрані дані з обладнання та устаткування, що використовується в різних виробничих процесах дозволяє застосовувати методи машинного навчання. Прогностичне обслуговування є одним з основних напрямків застосування технологій машинного навчання. В результаті огляду наукової літератури, можна зробити висновок, що впровадження методів прогностичного технічного обслуговування позитивно впливає на ефективність промислового підприємства, та допомагає покращити наступні показники: значно скорочуються планові та позапланові простой виробництва, а отже, і витрати; бригадам технічного

обслуговування надається цілеспрямовані вимоги щодо перевірки та ремонту обладнання; продовження терміну служби; скорочення чисельності технічних бригад, необхідних для ремонту.

В другому розділі «Методи машинного навчання для прогнозування несправностей обладнання». Було розглянуто та досліджено декілька методів машинного навчання, що часто використовуються для вирішення проблеми прогнозування несправностей промислового обладнання. Оскільки завдання прогнозування несправностей обладнання є задачею багатокласової класифікації було розглянуто наступні методи машинного навчання: випадковий ліс, градієнтний бустинг, нейронні мережі. Дані методи були обрані на основі огляду літератури та отриманих показників точності прогнозування в результаті експерименту. Для оцінки точності моделей було обрано матрицю невідповідностей, що широко застосовується при вирішенні задач класифікації. На основі матриці невідповідності для моделей було розраховано такі показники точності як, precision, recall, f1-score.

В третьому розділі «Модель прогнозного технічного обслуговування» було запропоновано модель (дорожню карту) впровадження прогнозного технічного обслуговування для промислових підприємств. Дана модель розділена на два етапи: отримання даних та прогнозування даних. Дана модель може розширюватись та доповнюватись додатковими етапами для більш ефективного впровадження прогнозного обслуговування, але вона містить в собі основні кроки, що необхідно виконати підприємству для досягнення мети.

В четвертому розділі «Застосування методів машинного навчання для прогнозування несправностей обладнання» було застосовано методи машинного навчання для вирішення завдання прогнозування несправностей, що були розглянуті в другому розділі. Для реалізації прогнозування несправностей було використано набір даних, що був опублікований компанією Microsoft. Даний набір даних, що містить показники обладнання за один рік (1.1.2015-1.1.2016) було проаналізовано наведено характеристику даних в підрозділі 4.1. З даного набору даних було вилучено ознаки необхідні для ефективного навчання методів

машинного навчання, оскільки ми не могли використовувати необроблені дані для навчання моделей. Вилучення ознак є обов'язковим кроком під час підготовки даних до навчання, оскільки це сильно впливає на точність навченої моделі. Отриманий набір було розділено на два набори для навчання та тестування моделей. В результаті навчання моделей було отримано моделі з хорошими показниками точності класифікації.

Для використання методів машинного навчання для прогностного обслуговування було запропоновано архітектуру програмного забезпечення на базі сервісів хмарних обчислень, а саме – Amazon Web Services. Дана архітектура дозволяє безпечно взаємодіяти з сховищами даних та обладнанням підприємства. На даній архітектурі було відображено сервіси для прогнозування несправностей, також враховуючи здатність до масштабування сервісів хмарних обчислень, дана архітектура може розширюватись та удосконалюватись для додавання нової та покращення існуючої функціональності системи.

ПЕРЕЛІК ПОСИЛАНЬ

1. E. Standard, "Maintenance terminology," in European Committee for Standardization 13306, ed. Brussels, 2001.
2. K.-S. Wang, Z. Li, J. Braaten, and Q. Yu, "Interpretation and compensation of backlash error data in machine centers for intelligent predictive maintenance using ANNs," *Advances in Manufacturing*, vol. 3, pp. 97-104, 2015
3. M. C. Garcia, M. A. Sanz-Bobi, and J. del Pico, "SIMAP: Intelligent System for Predictive Maintenance: Application to the health condition monitoring of a wind turbine gearbox," *Computers in Industry*, vol. 57, pp. 552-568, 2006
4. M. Aljumaili, K. Wandt, R. Karim, and P. Tretten, "eMaintenance ontologies for data quality support," *Journal of Quality in Maintenance Engineering*, vol. 21, pp. 358-374, 2015.
5. The Rise of Industrial Big Data. Available. [Електронний ресурс]. – Режим доступу: http://leadwise.mediadroit.com/files/19174the_rise_of_industrial_big_data_wp_gft834.pdf
6. J. Liu, D. Djurdjanovic, J. Ni, N. Casotto, and J. Lee, "Similarity based method for manufacturing process performance prediction and diagnosis," *Computers in Industry*, vol. 58, pp. 558-566, 8// 2007.
7. Economics of Manufacturing Machinery Maintenance. A Survey and Analysis of U. S. Costs and Benefits. [Електронний ресурс]. – Режим доступу: <https://nvlpubs.nist.gov/nistpubs/ams/NIST.AMS.100-34.pdf>
8. Maintenance costs and advanced maintenance techniques: Survey and Analysis. [Електронний ресурс]. – Режим доступу: <https://papers.phmsociety.org/index.php/ijphm/article/view/2883>
9. The Costs and Benefits of Advanced Maintenance in Manufacturing. [Електронний ресурс]. – Режим доступу: <https://nvlpubs.nist.gov/nistpubs/ams/NIST.AMS.100-18.pdf>
10. Advanced maintenance in manufacturing. Costs and Benefits. [Електронний ресурс]. – Режим доступу: <https://papers.phmsociety.org/index.php/phmconf/article/view/536>

11. Cheng, J.C.; Chen, W.; Chen, K.; Wang, Q. Data-driven predictive maintenance planning framework for MEP components based on BIM and IoT using machine learning algorithms. *Autom. Constr.* 2020, 112, 103087.
12. Ran, Y.; Zhou, X.; Lin, P.; Wen, Y.; Deng, R. A Survey of Predictive Maintenance: Systems, Purposes and Approaches. *IEEE Commun. Surv. Tutor* 2019
13. Florian, E.; Sgarbossa, F.; Zennaro, I. Machine learning for predictive maintenance: A methodological framework. In Proceedings of the XXIV Summer School “Francesco Turco”—Industrial Systems Engineering, Bergamo, Italy, 9–11 September 2020.
14. Filipe Gomes Pereira, L. Previsão de Falhas em Empanques Mecânicos da Refinaria de Matosinhos Usando Modelos de Machine Learning. Master’s Thesis, FEUP, University of Porto, Porto, Portugal, 2018.
15. Mba, C.U.; Makis, V.; Marchesiello, S.; Fasana, A.; Garibaldi, L. Condition monitoring and state classification of gearboxes using stochastic resonance and hidden markov models. *Measurement* 2018, 126, 76–95
16. Li, X.; Makis, V.; Zuo, H.; Cai, J. Optimal bayesian control policy for gear shaft fault detection using hidden semi-markov model. *Comput. Ind. Eng.* 2018, 119, 21–35.
17. Zuber, N.; Bajrić, R. Gearbox faults feature selection and severity classification using machine learning. *Eksploat. Niezawodn. Maint. Reliab.* 2020, 22, 748–756
18. Zaeri, R.; Ghanbarzadeh, A.; Attaran, B.; Moradi, S. Artificial neural network based fault diagnostics of rolling element bearings using continuous wavelet transform. In Proceedings of the 2nd International Conference on Control, Instrumentation and Automation, Shiraz, Iran, 27–29 December 2011; pp. 753–758.
19. Li, C.; Zurita, G.; Cerrada, M.; Cabrera, D. Multimodal deep support vector classification with homologous features and its application to gearbox fault diagnosis. *Neurocomputing* 2015, 168, 119–127.
20. Saufi, S.R.; Ahmad, Z.A.B.; Leong, M.S.; Lim, M.H. An intelligent bearing fault diagnosis system: A review. *MATEC Web Conf.* 2019, 255, 06005.
21. Chen, F.; Tang, B.; Chen, R. A novel fault diagnosis model for gearbox based on wavelet support vector machine with immune genetic algorithm. *Measurement* 2013, 46, 220–232.

22. Li, Z.; Peng, Z. A new nonlinear blind source separation method with chaos indicators for decoupling diagnosis of hybrid failures: A marine propulsion gearbox case with a large speed variation. *Chaos Solitons Fractals* 2016, 89, 27–39.
23. Cerrada, M.; Li, C.; Sánchez, R.V.; Pacheco, F.; Cabrera, D.; Oliveira, J.V.D. A fuzzy transition based approach for fault severity prediction in helical gearboxes. *Fuzzy Sets Syst.* 2018, 337, 52–73.
24. Lei, Y.; Zuo, M.J. Gear crack level identification based on weighted k-nearest neighbor classification algorithm. *Mech. Syst. Sig. Process.* 2009, 23, 1535–1547.
25. Jiang, R.; Yu, J.; Makis, V. Optimal bayesian estimation and control scheme for gear shaft fault detection. *Comput. Ind. Eng.* 2012, 63, 754–762.
26. Thyago, P.; Carvalho, A.; Fabrizzio, M.N.; Soares, R.V.; Roberto Francisco, P.; Basto, S.A. A systematic literature review of machine learning methods applied to predictive maintenance. *Comput. Ind. Eng.*
27. Çınar, Z.M.; Abdussalam Nuhu, A.; Zeeshan, Q.; Korhan, O.; Asmael, M.; Safaei, B. Machine Learning in Predictive Maintenance towards Sustainable Smart Manufacturing in Industry 4.0. *Sustainability* 2020, 12, 8211
28. L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.
29. L. Breiman. Some infinity theory for predictor ensembles. Technical Report 577, UC Berkeley, 2000.
30. L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001
31. L. Breiman. Consistency for a simple model of random forests. Technical Report 670, UC Berkeley, 2004.
32. Essence of Bootstrap Aggregation Ensembles. [Электронный ресурс]. – Режим доступа: <https://machinelearningmastery.com/essence-of-bootstrap-aggregation-ensembles/>
33. Bagging (Bootstrap Aggregation). [Электронный ресурс]. – Режим доступа: <https://corporatefinanceinstitute.com/resources/knowledge/other/bagging-bootstrap-aggregation/>

34. Gradient Boosting Decision Tree Algorithm Explained [Електронний ресурс]. – Режим доступу: <https://towardsdatascience.com/machine-learning-part-18-boosting-algorithms-gradient-boosting-in-python-ef5ae6965be4>
35. An Introduction to Gradient Boosting Decision Trees [Електронний ресурс]. – Режим доступу: <https://www.machinelearningplus.com/machine-learning/an-introduction-to-gradient-boosting-decision-trees/>
36. Artificial neural networks для вирішення бізнес задач [Електронний ресурс]. – Режим доступу: <https://evergreens.com.ua/ua/development-services/neural-network.html>
37. Siddharth Sharma, Simone Sharma, Anidhya Athaiya ACTIVATION FUNCTIONS IN NEURAL NETWORKS International Journal of Engineering Applied Sciences and Technology, 2020 Vol. 4, Issue 12, ISSN No. 2455-2143, Pages 310-316
38. Various Optimization Algorithms For Training Neural Network [Електронний ресурс]. – Режим доступу: <https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6>
39. Optimization Algorithms in Neural Networks [Електронний ресурс]. – Режим доступу: <https://www.kdnuggets.com/2020/12/optimization-algorithms-neural-networks.html>
40. Understanding Confusion Matrix [Електронний ресурс]. – Режим доступу: <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>
41. Що таке озеро даних? [Електронний ресурс]. – Режим доступу: <https://uk.education-wiki.com/8044363-what-is-a-data-lake>
42. Azure Predictive Maintenance. [Електронний ресурс]. – Режим доступу: <https://azure.microsoft.com/en-us/use-cases/predictive-maintenance>
43. Microsoft Azure Predictive Maintenance [Електронний ресурс]. – Режим доступу: <https://www.kaggle.com/arnabbiswas1/microsoft-azure-predictive-maintenance>
44. Fundamental Techniques of Feature Engineering for Machine Learning [Електронний ресурс]. – Режим доступу: <https://towardsdatascience.com/feature-engineering-for-machine-learning-3a5e293a5114>
45. Azure AI for predictive maintenance solutions [Електронний ресурс]. – Режим доступу: <https://docs.microsoft.com/en-us/azure/architecture/data-science-process/predictive-maintenance-playbook>

46. Cloud Services. Amazon Web Services [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/>
47. Amazon Simple Service Storage [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/ru/s3/>
48. Amazon IoT Core [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/ru/iot-core/>
49. Amazon Kinesis [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/ru/kinesis/>
50. Amazon Lambda [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/ru/lambda/>
51. Amazon SageMaker [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/ru/sagemaker/>
52. Amazon Relational Database [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/ru/rds/>
53. Amazon QuickSight [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/ru/quicksight/>
54. Amazon Simple Notification Service [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/ru/sns/>
55. Amazon CloudWatch [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/ru/cloudwatch/>
56. Amazon Virtual Private Cloud [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/ru/vpc/>
57. Amazon Identity and Access Management [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/ru/iam/>
58. AWS CloudFormation. [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/ru/cloudformation/>
59. Terraform [Электронный ресурс]. – Режим доступа: <https://www.terraform.io/>

ДОДАТОК А

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ



Кафедра інженерії програмного забезпечення

Магістерська робота

«Підвищення ефективності промислових підприємств на основі
нейронних мереж»

Виконав: студент групи ПДМ-61 Кривов'яз Юрій Олександрович

Керівник: к.т.н., доц., кафедри ІТЗ Щербина Ірина Сергіївна

Київ - 2021

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: підвищення ефективності промислових виробництв за рахунок впровадження прогнозного технічного обслуговування на основі методів машинного навчання.

Об'єкт дослідження: устаткування промислових виробництв.

Предмет дослідження: прогнозування несправностей промислового устаткування на основі методів машинного навчання.

КЛАСИФІКАЦІЯ СТРАТЕГІЙ ТЕХНІЧНОГО ОБСЛУГОВУВАННЯ

Стратегії технічного обслуговування	Коригувальне обслуговування	Відкладене
		Негайне
	Профілактичне обслуговування	На основі надійності
		На основі часу (заплановане)
		По можливості
	При поломці обладнання	При поломці обладнання
		На основі статистики
		На основі умов
		На основі умов

Коригувальне обслуговування

Переваги:

- Потрібне мінімальне планування;
- Процес дуже простий, тому його легко зрозуміти;
- Потрібно менше персоналу, оскільки щодня виконується менше роботи.

Недоліки:

- Невдача дуже непередбачувана;
- Ремонт обладнання може бути надзвичайно дорогим;
- Цей процес створює загрозу безпеці працівників та обладнанню.

Профілактичне обслуговування

Переваги:

- Безпека покращується завдяки зменшенню ймовірності катастрофічної відмови;
- Планування технічного обслуговування;
- Дозволяє збільшити термін експлуатації обладнання та скоротити час незапланованих простоїв.

Недоліки:

- Плани обслуговування іноді можуть бути неефективними та призвести до непотрібного технічного обслуговування (включаючи заміну деталей), що вимагає часу та грошей;
- Технічне обслуговування обладнання низького пріоритету.

Прогнозне обслуговування

Переваги:

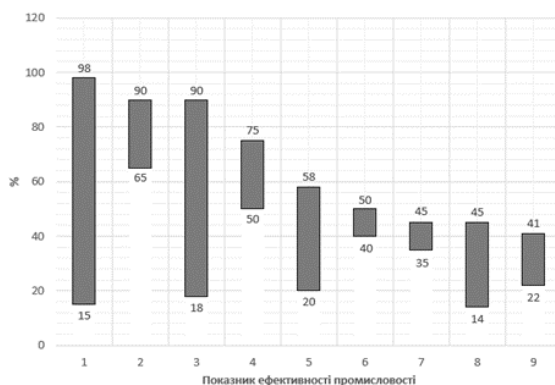
- Значно скорочуються планові та позапланові простої, а отже, і витрати;
- Надання технікам цілеспрямованих вимог щодо перевірки, ремонту та інструментів;
- Продовження терміну служби устаткування;
- Скорочення чисельності технічної бригади, необхідної для ремонту устаткування.

Недоліки:

- Додаткові витрати на навчання персоналу та встановлення технічних засобів для моніторингу стану обладнання.

3

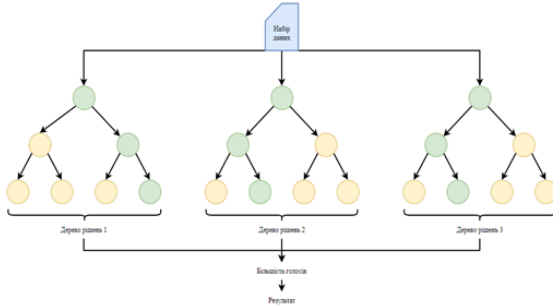
ВПЛИВ ВПРОВАДЖЕННЯ СТРАТЕГІЇ ПРОГНОЗНОГО ТЕХНІЧНОГО ОБСЛУГОВУВАННЯ



1. Зменшення витрат на технічне обслуговування
2. Зменшення кількості поломок обладнання
3. Зменшення кількості дефектів та переробок
4. Зменшення кількості аварій
5. Збільшення випуску продукції/виробництва
6. Підвищення продуктивності праці
7. Зменшення часу простою
8. Скорочення запасів
9. Зменшення кількості відмов клієнтів

4

МЕТОД МАШИННОГО НАВЧАННЯ: ВИПАДКОВИЙ ЛІС (RANDOM FOREST)



Розрахунок голосів дерев обчислюється за формулою:

$$v_m = \sum_{t=1}^T I(\hat{y}_t = m), \quad (1)$$

де, v_m – голоси дерев для класу m ,
 I – функція індикатора для класу m ,
 T – кількість дерев в ансамблі,
 m – клас,
 \hat{y}_t – результат дерева t .

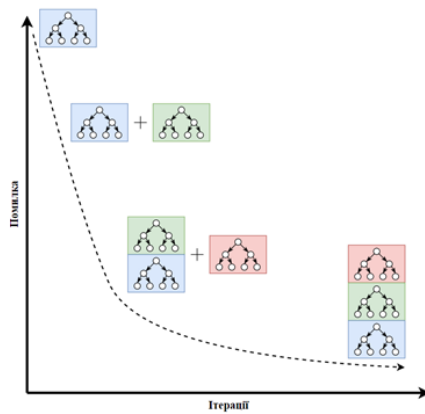
Результат випадкового лісу обчислюється формулою:

$$\hat{y} = \operatorname{argmax}_{m \in \{1, \dots, M\}} v_m, \quad (2)$$

де, \hat{y} – результат випадкового лісу,
 M – кількість класів.

5

МЕТОД МАШИННОГО НАВЧАННЯ: ГРАДІЄНТНИЙ БУСТИНГ (GRADIENT BOOSTED TREE)



Алгоритм градієнтного бустингу

Диференційована функція втрат: $L(F(x), y)$. (3)

Навчальна вибірка: $\{(x_i, r_i), i = 1, \dots, n\}$. (4)

Кількість ітерацій: T .

1. Впровадження моделі зі сталими значеннями:

$$F_0(x) = \operatorname{argmin}_a \sum_{i=1}^n L(a, y_i). \quad (5)$$

2. Для кожного кроку навчання t ($t = 1, \dots, T$):

а) Обраховуємо псевдо-залишок

$$\text{модель: } r_{it} = - \left[\frac{\partial L(F(x_i), y_i)}{\partial F(x_i)} \right]_{F(x)=F_{t-1}(x)}, \quad i = 1, \dots, n. \quad (6)$$

б) Навчаємо базовий алгоритм «слабкого» учня $b_t(x)$ по вибірці $\{(x_i, r_{it}), i = 1, \dots, n\}$.

в) Вираховуємо коефіцієнт a_t перед базовим алгоритмом $b_t(x)$, як вирішення наступної задачі оптимізації:

$$a_t = \operatorname{argmin}_a \sum_{i=1}^n L(F_{t-1}(x_i) + a b_t(x_i), y_i). \quad (7)$$

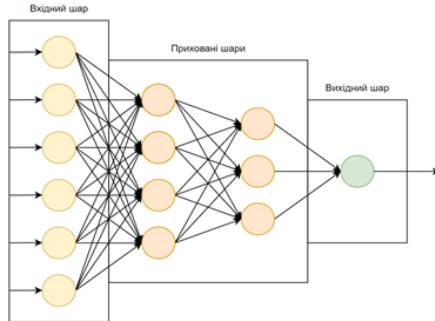
д) Розширюємо модель:

$$F_t(x) = F_{t-1}(x) + a_t b_t(x). \quad (8)$$

3. Шукана функція: $F_T(x)$.

6

НЕЙРОННА МЕРЕЖА (NEURAL NETWORK)



Значення нейрону обчислюється формулою:

$$z = wa + b, \quad (9)$$

де, z – вихід нейрону,
 a – вихід функції активації попереднього нейрону,
 b – зміщення,
 w – вага нейрону.

Значення нейрону з функцією активації обчислюється формулою:

$$a = \sigma(z), \quad (10)$$

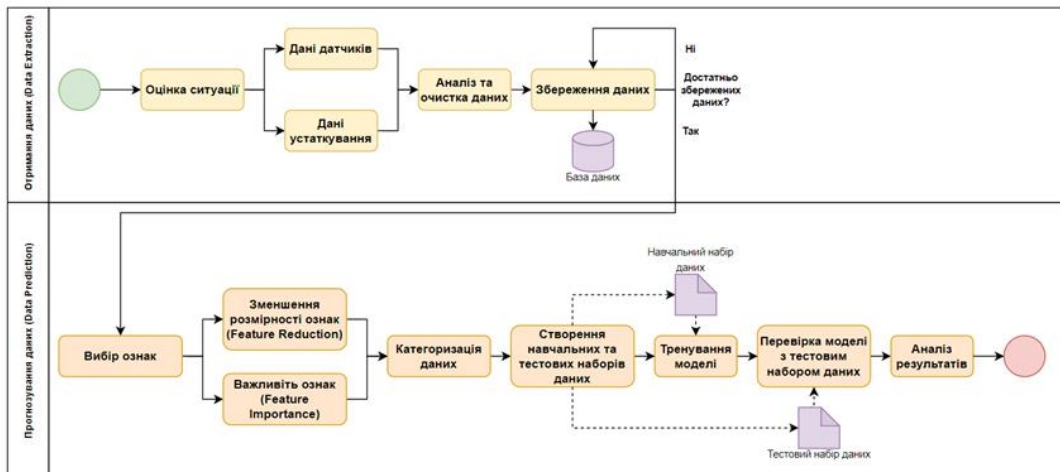
де, σ – функція активації нейрону.

Функція активації ReLu, обчислюється за формулою:

$$\sigma(z) = \max(0, z) \quad (11)$$

7

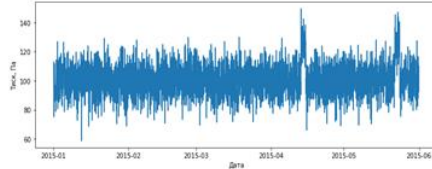
АЛГОРИТМ ПРОГНОЗНОГО ТЕХНІЧНОГО ОБСЛУГОВУВАННЯ



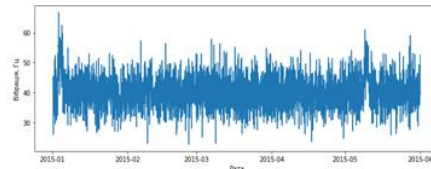
8

ХАРАКТЕРИСТИКА НАБОРУ ДАНИХ

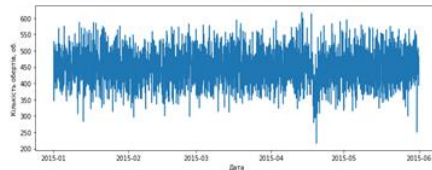
Вихідні дані обладнання (показники датчиків виробництва) містять 876 100 записів для 100 одиниць обладнання включаючи наступні дані: ідентифікатор обладнання (machineID), дата (datetime), напруга (volt), кількість обертів (rotate), тиск (pressure), вібрація (vibration).



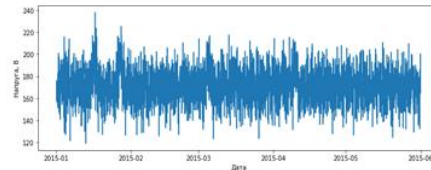
Залежність тиску від часу (machineID = 1)



Залежність вібрації від часу (machineID = 1)



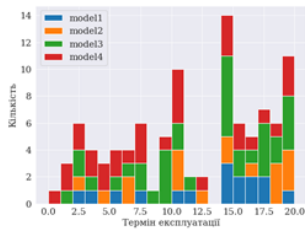
Залежність кількості обертів від часу (machineID = 1)



Залежність напруги від часу (machineID = 1)

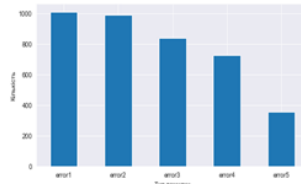
9

ХАРАКТЕРИСТИКА НАБОРУ ДАНИХ



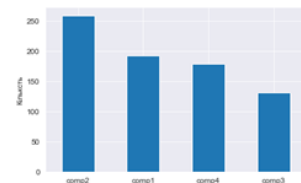
Залежність кількості обладнання за терміном експлуатації та моделлю

Відомості про обладнання містять дані про 100 одиниць обладнання, та включають наступні дані: ідентифікатор обладнання (machineID), модель (model), термін експлуатації обладнання (age).



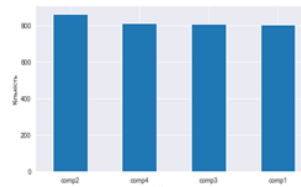
Залежність кількості помилок обладнання за типом помилки

Журнал помилок обладнання містить 3919 записів і включає наступну інформацію: дата (datetime), ідентифікатор обладнання (machineID), ідентифікатор поломки (errorID). В журналі помилок існує п'ять типів помилок: error1, error2, error3, error4, error5.



Залежність кількості поломок обладнання за компонентом

Історія поломок обладнання містить 761 запис і включає наступні дані: дата (datetime), ідентифікатор обладнання (machineID), назва компонента, причина поломки (failure). В причині поломки містяться дані про компонент, який зазнав поломки.

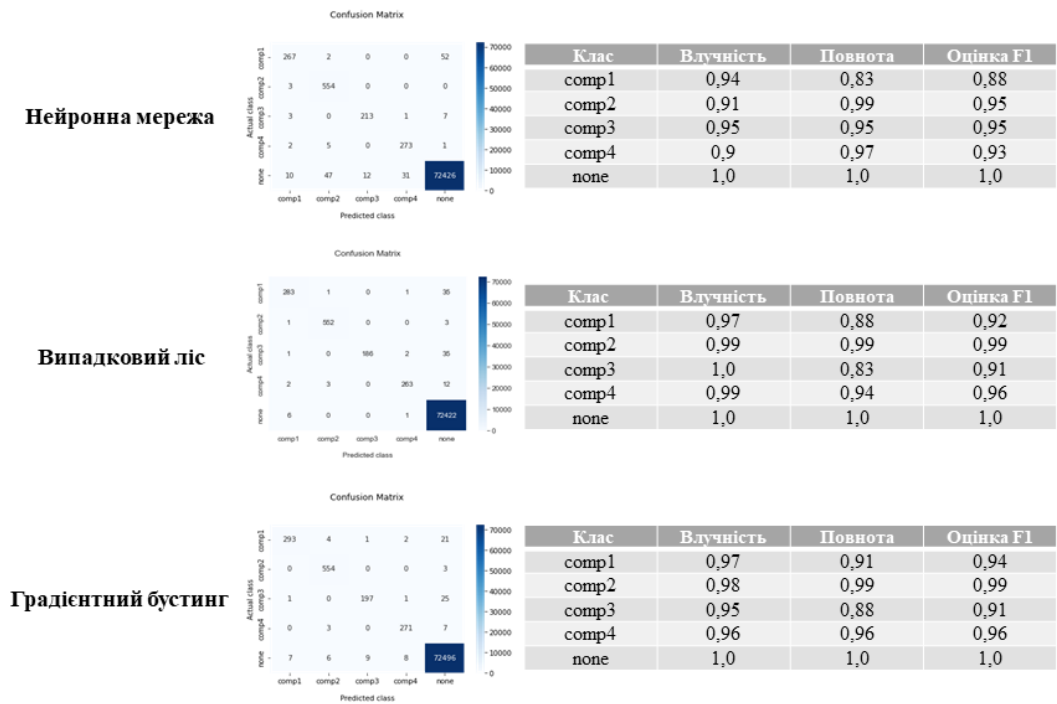


Залежність кількості технічного обслуговування за типом компонентом для якого виконувалось обслуговування

Історія технічного обслуговування містить 3286 записів та включає наступну інформацію: дата (datetime), ідентифікатор обладнання (machineID), назва компонента (comp) для якого відбувалось технічне обслуговування.

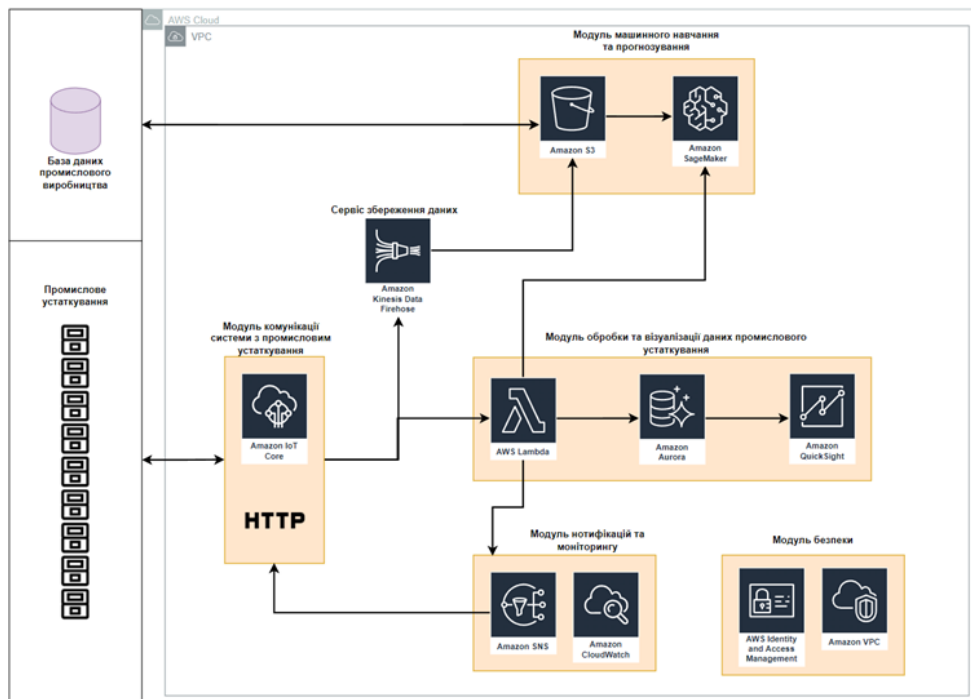
10

РЕЗУЛЬТАТИ МЕТОДІВ МАШИННОГО НАВЧАННЯ



11

АРХІТЕКТУРА СИСТЕМИ ПРОГНОЗУВАННЯ НЕСПРАВНОСТЕЙ ДЛЯ ПРОГНОЗНОГО ОБСЛУГОВУВАННЯ



12

ВИСНОВКИ

1. Було розглянуто та досліджено підходи технічного обслуговування: коригувальне обслуговування, профілактичне обслуговування та прогнозне обслуговування.
2. Для прогнозування несправностей було досліджено та обрано наступні методи машинного навчання: випадковий ліс, градієнтний бустинг, нейронні мережі. Для оцінки точності моделей було обрано матрицю невідповідностей, що широко застосовується при вирішенні задач класифікації. На основі матриці невідповідності для моделей було розраховано такі показники точності як, precision, recall, f1-score.
3. Було запропоновано алгоритм впровадження прогнозного технічного обслуговування для промислових підприємств.
4. Було застосовано методи машинного навчання, такі як, випадковий ліс, градієнтний бустинг, нейронні мережі для вирішення завдання прогнозування несправностей. В результаті навчання моделей було отримано моделі з хорошими показниками точності класифікації.
5. Для використання методів машинного навчання для прогнозного обслуговування було розроблено архітектуру програмного забезпечення на базі сервісів хмарних обчислень, а саме – Amazon Web Services.

13

АПРОБАЦІЯ РОБОТИ

Статті:

1. Кривов'яз Ю. О. Застосування концепцій прогнозного технічного обслуговування з використанням методів машинного навчання. // Актуальні наукові дослідження в сучасному світі.

Тези доповідей:

1. Кривов'яз Ю. О., Щербина І. С. Використання нейронних мереж в задачах і підвищення ефективності промисловості. // Науково-технічна конференція «Сучасний стан та перспективи розвитку ІоТ» - Київ: ДУТ, 2021. – С.64-66.

14

ДЯКУЮ ЗА УВАГУ!