

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет радіоелектроніки
Факультет Комп'ютерних наук
Кафедра Програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

Дослідження методів та моделей предиктивного обслуговування з використанням
машинного навчання

Виконав:

Студент 2 курсу, групи ІПЗм-20-1

Любченко Ю. М.

(ім'я, прізвище)

Спеціальність 121 Інженерія програмного
забезпечення

Тип програми освітньо – наукова

Керівник доц. Голян В. В.
(посада, прізвище, ініціали)

Допускається до захисту:

Зав. Кафедри

З.В. Дудар

2022 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
Кафедра Програмної інженерії
Рівень освіти другий (магістерський)
Спеціальність 121– Інженерія програмного забезпечення
Тип програми освітньо-наукова програма
Освітня програма Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. Кафедри _____
(підпис)

« _____ » _____ 202__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студента Любченку Юрію Михайловичу
(прізвище, ім'я, по батькові)

1. Тема роботи: дослідження методів та моделей предиктивного обслуговування з використанням машинного навчання.

затверджена наказом університету від « ____ » _____ 202__ р. № ____

2. Термін подання студентом роботи до екзаменаційної комісії: “ ____ ” _____ 202__ р.

3. Вихідні дані до роботи: предиктивне обслуговування обладнання, машинне навчання, мова програмування Python, пояснювальна записка.

4. Зміст пояснювальної записки: мета роботи, аналіз предметної галузі і постановка задачі, дослідження технологій предиктивного обслуговування обладнання зокрема технології машинного навчання, вивчення можливості її використання у інформаційній системі.

КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів роботи | Терміни виконання етапів роботи | Примітка |
|---|---|---------------------------------|----------|
| 1 | Аналітичний огляд | 20.01.2022 | виконано |
| 2 | Аналіз методів предиктивного обслуговування | 01.02.2022 | виконано |
| 3 | Аналіз методів машинного навчання | 08.02.2022 | виконано |
| 4 | Вибір методів з постановкою задач | 17.02.2022 | виконано |
| 5 | Написання пояснювальної записки | 20.03.2022 | виконано |
| 6 | Розробка практичної частини роботи | 25.04.2022 | виконано |
| 7 | Підготовка матеріалів для захисту | 08.04.2022 | виконано |

Дата видачі завдання _____ 202 р.

Студент _____

(підпис)

Керівник роботи _____

доц. Голян В. В

(підпис)

(посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Кваліфікаційна робота магістра містить: 60 с., 11 рис., 2 табл., 20 джер.

ПРЕДИКТИВНЕ ОБСЛУГОВУВАННЯ, MACHINE LEARNING, PYTHON,,
TENSORFLOW, VISUAL STUDIO CODE.

Об'єктом дослідження є задача прогнозування критичного стану обладнання з використанням машинного навчання.

Метою роботи є підвищення ефективності виробництва на підприємствах.

Методи розробки базуються на технологіях Python, Tensorflow, Keras. Дані з датчиків обладнання отримано з відкритих джерел.

В результаті роботи було проведено аналіз та моделювання предметної області, досліджено методи та моделі машинного навчання, вивчено особливості архітектур нейронних мереж для подальшої програмної реалізації, програмно реалізовано прогнозування критичного стану обладнання.

The object of the study is the problem of predicting critical state of equipment using machine learning.

The aim of the work is to increase the efficiency of production at facilities.

Development methods are based on technologies such as Python, Tensorflow, Keras. Data from equipment sensors were obtained from open sources.

As a result, the analysis and modeling of the subject area was performed, the methods and models of machine learning were studied, the features of neural network architectures for further software implementation were studied, the forecast of critical equipment state was implemented.

Умови публікації пояснювальної записки

Я, Любченко Юрій Михайлович

(прізвище, ім'я, по батькові)

студент групи ППЗМ-20-1, здобувач вищої освіти на другому (магістерському) рівні

кафедра програмної інженерії,

(повна назва кафедри)

заявляю: моя кваліфікаційна робота на тему

Дослідження методів та моделей предиктивного обслуговування з використанням машинного навчання,

(назва роботи)

що буде представлена до ЕК для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

| | |
|--|----|
| Перелік скорочень..... | 7 |
| Вступ..... | 8 |
| 1. Аналіз предметної галузі..... | 9 |
| 1.1 Аналіз предметної галузі..... | 9 |
| 1.2 Виявлення проблем та актуалізація рішень..... | 10 |
| 1.3 Постановка задачі..... | 17 |
| 2. Обґрунтування методів досліджень..... | 18 |
| 2.1 Обґрунтування методів досліджень..... | 18 |
| 2.2 Етапи предиктивного обслуговування, що орієнтоване надані..... | 19 |
| 2.2.1 Передобробка..... | 20 |
| 2.2.2 Аналіз індикаторів стану..... | 21 |
| 2.2.3 Виявлення аномалій..... | 21 |
| 2.2.4 Діагностика..... | 22 |
| 2.2.5 Прогнозування..... | 24 |
| 2.2.6 Відновлення..... | 25 |
| 2.3 Методи глибокого навчання..... | 26 |
| 2.3.1 Нейронна мережа прямого поширення та багат шаровий перцептрон..... | 26 |
| 2.3.2 Згортова нейронна мережа..... | 27 |
| 2.3.3 Рекурентна нейронна мережа..... | 27 |
| 2.3.4 Обмежена машина Больцмана та глибинна мережа переконань..... | 28 |
| 2.3.5 Автокодувальник..... | 28 |
| 2.3.6 Генеративні моделі..... | 29 |
| 2.4 Методи глибокого навчання для PdM..... | 30 |
| 2.5 Опис алгоритму..... | 31 |

| | |
|---|----|
| | 6 |
| 2.6 Практична реалізація..... | 36 |
| 2.6.1 Прийом даних та їх передобробка..... | 36 |
| 2.6.2 Створення моделі нейронної мережі та її оцінка..... | 38 |
| Висновки..... | 43 |
| Перелік джерел..... | 44 |
| Додаток А..... | 47 |
| Додаток Б..... | 48 |
| Додаток В..... | 50 |
| Додаток Г..... | 52 |

ПЕРЕЛІК СКОРОЧЕНЬ

ККД — коефіцієнт корисної дії,
RCM — reliability centered maintenance,
KPI — (від англ.) key performance indicator,
PdM — (від англ.) predictive maintenance,
SCADA system — (від англ.) supervisory control and data acquisition system,
HMI — (від англ.) huma-machine interface,
PLC — (від англ.) programmable logic controler,
RTU — (від англ.) remote transmission unit,
ІЧ — інфрачервоний,
RUL — (від англ.) remaining useful life,
RCA — (від англ.) root cause analysis,
ML — (від англ.) machine learning,
DL — (від англ.) deep learning,
ANN — (від англ.) artificial neural network,
ReLU — (від англ.) rectified linear unit,
RNN — (від англ.) reccurent neural network,
NLP — (від англ.) natural language processing,
GAN — (від англ.) generative adversarial network,
LSTM — (від англ.) long short-term memory,

ВСТУП

На сьогоднішній день підприємства як ніколи стурбовані підвищенням ефективності. Проте, найчастіше дороге устаткування працює на 30-40% своєї потужності. Зазвичай причина проста: працівники не зацікавлені в результатах своєї праці. Зрештою, неможливо підвищити ефективність використання обладнання, не маючи автоматичних інструментів контролю завантаження верстата: чи знаходиться він в стані холостого ходу, вимкнений або працює під навантаженням. Створення “цифрових двійників” [1] для відстежування фактичного стану дозволяють підвищити ККД верстата на 25-30%. Ще одним напрямком є предиктивне обслуговування обладнання. Наприклад, спеціальні датчики відслідковують показники вібрації, і досягнення ними граничних значень свідчить про те, що верстат треба обстежити. Зникає необхідність раз в квартал оглядати обладнання, замінювати масло і проводити роботи з обслуговування. Можна робити це не за застарілим регламентом, а за фактичною необхідності. Дана робота відображає хід і результати магістерського дослідження проблеми прогнозування критичного стану обладнання з ціллю покращення ефективності виробництва.

ЧАСТИНА 1

1.1 Аналіз предметної галузі

Для забезпечення максимального рівня надійності недостатньо застосовувати традиційний підхід до обслуговування — вести графік робіт, контролювати закупівлі запчастин, утримувати досвідчених фахівців [2]. Методологія Industry 4.0 [3] передбачає абсолютно новий підхід — віддалений моніторинг та контроль, і предиктивне обслуговування обладнання. Сама технологія інтелектуального обслуговування ґрунтується на методології обслуговування на підставі надійності [4] (RCM).

RCM — стратегія управління основними виробничими фондами, головним принципом якої є недопущення відхилення параметрів стану обладнання до значень, які призводять до порушення функціонування об'єкта або системи в конкретному виробничому оточенні. Один з ключових KPI - зниження витрат на обслуговування обладнання за рахунок збільшення його ефективності, зниження простоїв і збільшення продуктивності технічних фахівців.

Методологія RCM заснована на концепції, згідно з якою метою обслуговування є не підтримка кожної одиниці обладнання в бездоганному стані, а забезпечення надійності критичних для діяльності підприємства виробничих і технологічних процесів.

Але поломки не завжди відбуваються через однаковий час. Іноді вузол може продовжувати роботу в штатному режимі набагато довше, а іноді виходить з ладу раніше терміну. Методологія RCM полягає не в розрахунку середнього терміну служби, а в пошуку тієї точки часу, в якій показники роботи починають падати. Коли починає спостерігатися погіршення показників роботи, то можна зробити висновок, що потрібно готуватися до відмови обладнання та проведенню ремонту.

У багатьох математичних моделях деградація компонентів представляється на графіку P-F кривої [5], що зображений на рисунку 1.

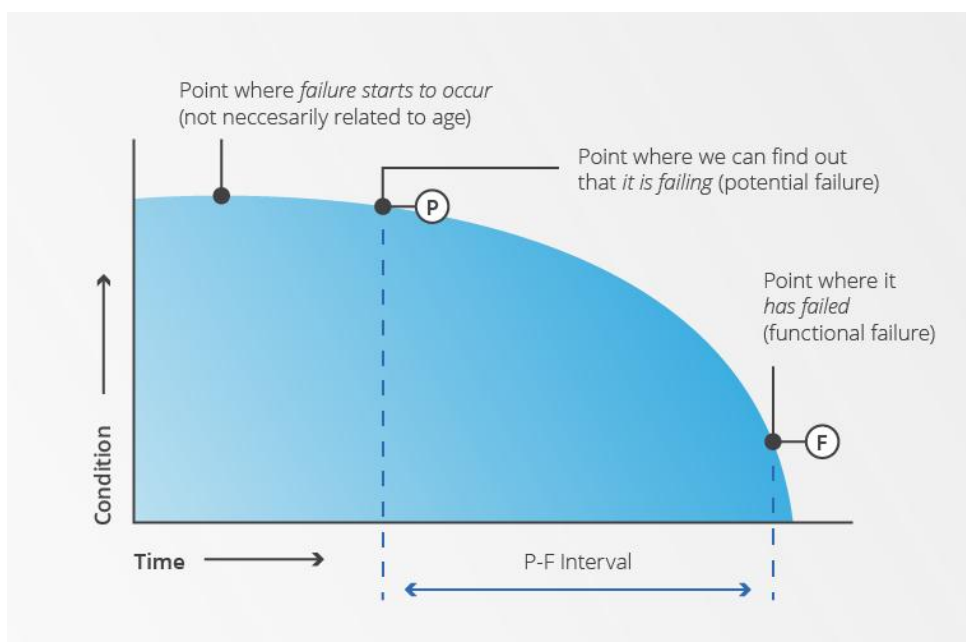


Рисунок 1 — Приклад P-F кривої

Крива P-F — це спосіб відображення стану працездатності обладнання, де з плином часу або машинних циклів, стан погіршується від здорового робочого до відмови.

1.2 Виявлення проблем та актуалізація рішень

Насьогодні існує декілька видів обслуговування обладнання, а саме: аварійне, регламентоване та предиктивне.

Аварійне обслуговування — обслуговування обладнання після виходу з ладу. Зазвичай цей вид застосовується лише для простого та дешевого обладнання,

коли заміна обійдеться дешевше, ніж кваліфіковане обслуговування для відновлення працездатності.

Регламентоване обслуговування — це планове обслуговування згідно регламенту. Воно проводиться відповідно до рекомендацій виробника через проміжки часу. Роботи проводяться з певним періодом, що визначається за допомогою статистичного аналізу та відповідно до регламентів, складає час, коли 98% обладнання працює без збоїв.

Доцільно, що можливо скористуватись гарантією виробника та проводити обслуговування за регламентом. Проте, не менш ніж у 50% випадків, немає необхідності у фактичному ремонті обладнання. Несвоєчасне обслуговування промислового обладнання — це не тільки прямі втрати від поломки у вигляді витрат на нові запчастини. Це ще і скорочення обсягів виробництва, витрати на реорганізацію і перепланування процесів, що веде до ще більших витрат у майбутньому. Більш того, для багатьох видів обладнання, ремонт не знижує вірогідність виходу зі строю.

Існує ще один вид обслуговування — предиктивне, тобто за фактичним технічним станом. При такому виді обслуговування, обладнання контролюється неперервно та в залежності від результатів роботи складається передбачення стану обладнання та формується план технічного обслуговування. Але виходи з ладу не завжди відбуваються через однаковий час. Іноді вузол може продовжувати роботу в штатному режимі набагато довше, а іноді виходить з ладу раніше терміну.

Системи, що реалізуються предиктивне обслуговування, здатні на основі поточного стану обладнання прогнозувати ті точки часу, коли показники системи почнуть падати. SCADA системи є одними з таких систем.

Система SCADA — це система управління автоматизацією (див. рис. 2), яка використовується в таких галузях, як енергетика, видобуток нафти та газу,

водопостачання та багато інших. Система має централізовану природу — вона виконує моніторинг та контроль цілих ділянок, починаючи від промислового заводу і закінчуючи комплексом заводів по всій країні. Система SCADA працює шляхом контролю сигналів, які обмінюються даними по каналах, щоб забезпечити користувача дистанційним управлінням будь-яким обладнанням у системі. Дана система також реалізує розподілену базу даних, яка містить точки зі всієї установи. Ці точки представляють собою єдине вхідне або вихідне значення, яке контролюється або контролюється та зберігаються у розподіленій базі даних як пари міток часу та значення точки. Дуже поширеним є налаштування підсистем також для отримання метаданих. Наприклад, програмований логічний контролер може збирати метадані про статистику сигналізацій.

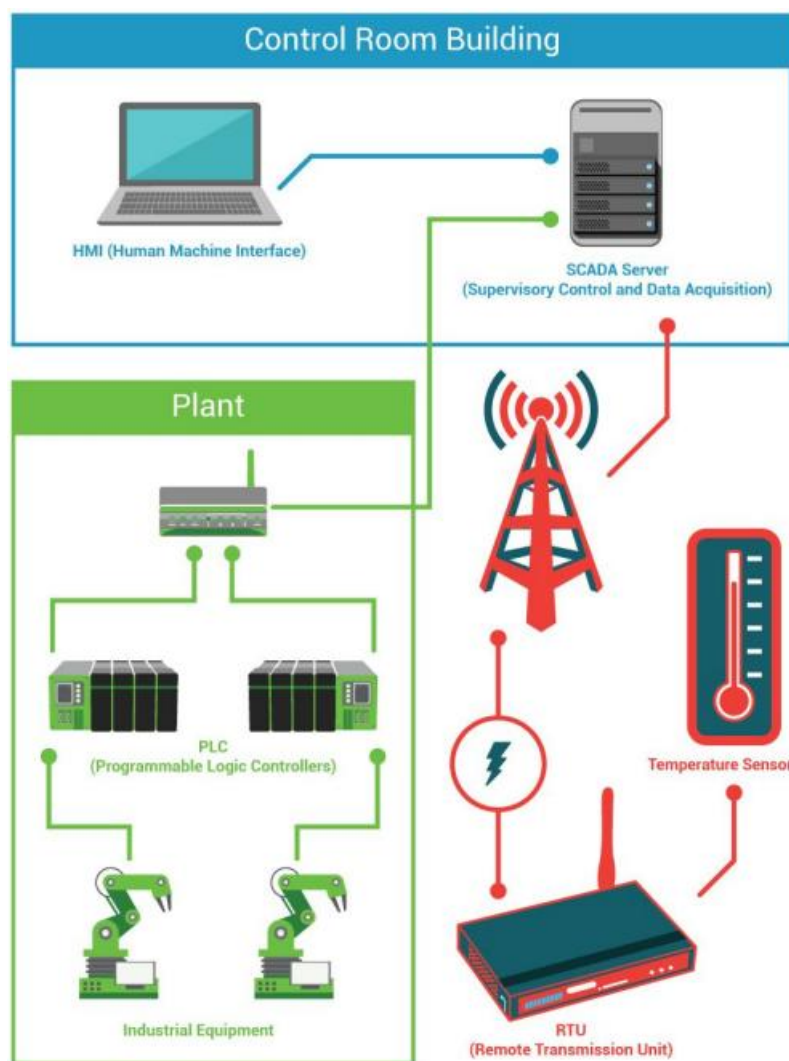


Рисунок 2 — Схема SCADA системи

Хоча SCADA системи спрощують певну інфраструктуру, їх компоненти досить складні. Існує п'ять важливих складових частин таких систем:

- людинно-машинний інтерфейс;
- система моніторингу та контролю;
- модулі для віддаленої передачі;
- програмовані логічні контролери;
- інфраструктури зв'язку.

Основним завданням платформ SCADA є надання користувачам швидкого доступу до PLC / RTU та спрощена інтеграція елементів керування обладнанням в інтерфейси користувача. Ці системи можуть бути ефективним інструментом, але їх потрібно суттєво контролювати за допомогою НМІ.

Цей напівавтоматичний підхід не враховує більш складних динамічних моделей поведінки машин або контекстуальних даних, що стосуються виробничого процесу в цілому. Для вирішення комплексних проблем використовуються моделі нейронних мереж. Вони можуть моделювати взаємозв'язки між вхідними та вихідними даними, які є нелінійними та складними; робити узагальнення та виведення висновків; розкривати приховані взаємозв'язки та закономірності; і моделювати дуже нестабільні дані та відхилення, необхідні для прогнозування рідкісних подій. Як результат, нейронні мережі можуть покращити процеси прийняття рішень у сфері предиктивного обслуговування обладнання.

У даній роботі [6] було описано 10 бажаних властивостей PdM системи:

- швидке виявлення проблем та діагностика;
- ізольованість (розрізняють різні типи відмов);
- міцність системи;
- можливість ідентифікації нових несправностей;

- оцінка помилки класифікації;
- адаптивність;
- інструмент для роз'яснення результатів;
- мінімальні вимоги до моделювання;
- обчислення та зберігання в режимі реального часу;
- можливість ідентифікації множинних несправностей.

Основні проблеми варіантів промислового використання PdM систем — це їхня поведінка та мінливість даних. Це трапляється навіть у активах, що працюють за однаковими характеристиками, враховуючи механічні допуски, налаштування кріплення та інші фактори. Ці фактори ускладнюють можливість повторного використання моделі PdM. Іншими проблемами є збір якісних даних, виконання правильної попередньої обробки та розробка функцій, щоб отримати репрезентативний набір даних. Крім того, кожне спостереження пов'язане з попередніми, тому їх слід аналізувати разом, що збільшує розмірність даних і складність моделювання.

Для предиктивного обслуговування найчастіше використовуються наступні методи: аналіз механічного ультразвуку, аналіз вібрацій, магнітно-порошкова дефектоскопія, термографія, аналіз струму сигналу двигуна, неруйнівне тестування. Проте також є додаткові методи моніторингу крутного моменту, напруги, акустичної емісії та тиску.

Розглянемо їх більш детально:

- інфрачервона термографія, відома як технологія неруйнівного чи непроникного тестування, в інтелектуальному обслуговуванні широко використовується інфрачервона термографія. За допомогою ІЧ-камер персонал здатний визначати високі температури в обладнанні. Зношені компоненти, включаючи несправні електричні ланцюги, зазвичай випромінюють тепло, яке відобразатиметься як пікова точка на тепловому зображенні. Шляхом швидкого

визначення гарячих точок, інфрачервоні інспекції можуть визначити проблеми та допомогти уникнути дорогого ремонту та простою. Також іншими способами використання інфрачервоної технології є виявлення теплових аномалій та проблем з технологічними системами, що покладаються на утримання та / або передачу тепла;

- акустичний моніторинг, за допомогою акустичних технологій персонал може виявити витіки газу, рідини або вакууму в обладнанні на звуковому або ультразвуковому рівні. Звукова технологія, яка вважається дешевшою за ультразвукову технологію, корисна для механічного обладнання, але обмежена у використанні. Ультразвукова технологія має більше застосувань і є більш надійною у виявленні механічних проблем. Вона дозволяє інженеру раніше виявити тертя і напругу в обертових машинах, що можуть бути індикаторами погіршення стану обладнання за допомогою контрольних-вимірних приладів, що перетворюють звуки в діапазоні від 20 до 100 кілогерц [7] (частоти, що генеруються зношеними та недомазаними підшипниками, несправним електрообладнанням, негерметичними клапанами, тощо) у слухові або зорові сигнали, які можна почути або побачити;

- аналіз вібрацій, використовується в основному для високошвидкісного обертового обладнання, дозволяє інженеру спостерігати вібрації машини за допомогою ручного аналізатора або датчиків реального часу, вбудованих в обладнання. Машина, що працює в пікових умовах, демонструє певний вібраційний паттерн. Коли такі компоненти, як підшипники та вали, починають зношуватися та виходити з ладу, машина почне генерувати інший вібраційний паттерн. Проактивно проводячи моніторинг обладнання, навчений інженер може порівняти показання з відомими режимами відмов, щоб визначити, де виникають проблеми. Серед проблем, які можна виявити за допомогою аналізу вібрацій є: відхилення, зігнуті вали, невривноважені компоненти, нещільні механічні

компоненти та проблеми з двигуном. Забезпечення підготовки персоналу буде життєво важливим, оскільки може бути важко передбачити несправність машини за допомогою аналізу вібрацій. Єдиним недоліком використання вібраційного аналізу є витрати, пов'язані з його реалізацією;

- аналіз нафти, є ефективним інструментом для предиктивного обслуговування. Це дозволяє техніку перевірити стан оливи та визначити, чи присутні інші частинки та забруднення. Деякі тести нафтового аналізу можуть виявити в'язкість, наявність води або зношених металів, кількість частинок та кислотне чи основне число. Однією з переваг використання аналізу масла є те, що початкові тести встановлюють базову лінію для нової машини. При правильному виконанні, аналіз масла може дати безліч результатів, які допоможуть зробити предиктивне обслуговування успішним.

Навколишні та експлуатаційні умови — це такі умови, за яких працює промислове обладнання. До навколишньої умов відносяться зовнішні умови, що можуть впливати на роботу обладнання, такі як температура навколишнього середовища та навколишні вібрації. На відміну від цього, експлуатаційні умови призначені робочим процесом за технічними характеристиками, такі як бажана швидкість, сила або положення. Крім того, з машинних датчиків надходять додаткові вимірювання. Ці дані, що надходять протягом часу, створюють набір даних у формі часового ряду. Його аналіз з використанням методів моніторингу стану дозволяє визначати стан компонентів і станків шляхом порівняння моделей і тенденцій з історичними даними.

1.3 Постановка задачі

У теоретичній частині роботи буде вивчено найбільш релевантні методи машинного навчання, що орієнтовані на дані для застосування у контексті сфери предиктивного обслуговування, розглядається придатність моделей глибокого навчання і порівнюються їх переваги та недоліки зі статистичними та класичними моделями машинного навчання.

У практичній частині даної роботи буде реалізовано один із сучасних алгоритмів машинного навчання для прогнозування RUL авіаційних двигунів.

Перед початком подальшого проектування програмної частини системи необхідно визначитися з вимогами до неї. Під вимогами розуміється сукупність властивостей, якостей та функцій до програмної системи, що реалізується. В даному випадку необхідно зазначити функціональні властивості.

Результатом роботи повинна стати система, що включає в себе такі функції:

- 1) Прийом даних та їх передобробку;
- 2) Створення моделі нейронної мережі;
- 3) Оцінка результатів роботи нейронної мережі.

ЧАСТИНА 2

2.1 Обґрунтування методів досліджень

Більшість досліджень, що вивчають PdM, можна класифікувати за трьома підходами: на основі фізичної моделі, на основі даних і гібридний підхід. Методи фізичної моделі використовують знання систем для побудови математичного опису їх деградації. Їхнє фізичне значення легко зрозуміти, але важко реалізувати в складних системах. Методи, керовані даними, передбачають стан систем, відстежуючи їх стан за допомогою рішень, засвоєних на основі історії даних. Вони складаються зі статистичних методів, функцій надійності та методів штучного інтелекту. Вони підходять для складних систем, оскільки їм не потрібно розуміти, як вони працюють. Однак зв'язати їх вихід з фізичним змістом важче. Нарешті, гібридний підхід поєднує два вищезгадані підходи. Методи, керовані даними та глибокого навчання, набули популярності в промисловості останніми роками через збільшення машинного збору даних, що дозволяє розробляти точні моделі PdM у складних системах.

Метод дослідження у даній роботі, що був застосований щодо використання моделей машинного навчання для предиктивного обслуговування пояснюється в цьому параграфі. Спочатку було проаналізовано контекст PdM та сферу його застосування. Після цього були досліджені різні типи моделей. Потім було проаналізовано моделі, що засновуються на даних. Нарешті, було ретельно переглянуто моделі глибокого навчання. Ця методологія дозволила отримати загальне уявлення про обсяг, а потім зосередитися на конкретних темах дослідження. Крім того, аналіз сучасних методів дозволив порівняти методи та обговорити проблеми та перспективи моделей DL для PdM. Аналіз було проведено шляхом запитів до пошукових систем щодо вищезазначених тем.

Спочатку було використано пошукову систему Google. Проте, після того, як дослідження наблизилося до більш специфічних термінів, було використано пошукову систему Google Scholar.

Незважаючи на те, що вже існує декілька опублікованих оглядів щодо моделей машинного навчання та глибокого навчання для предиктивного обслуговування, ця робота надає такі додаткові сучасні внески: розглядаються та пояснюються найбільш релевантні методи, орієнтовані на дані, на архітектурах глибокого навчання із застосуванням до PdM, що надає широкий погляд на доступні методи спрощеним і структурованим способом; розглядається придатність моделей глибокого навчання для PdM і порівнюються їх переваги та недоліки зі статистичними та класичними моделями машинного навчання.

2.2 Етапи предиктивного обслуговування, що орієнтоване на дані

Більшість моделей глибокого навчання для PdM засновані на тих же принципах, що й інші методи машинного навчання та статистичні методи. А саме, більшість методів, дотримуються інкрементних кроків, що представлені у стандарті OSA-CBM [8].

Ці кроки представлено у наступному порядку:

- 1) Виявлення аномалій;
- 2) Діагностика;
- 3) Прогнозування;
- 4) Відновлення.

Зазвичай перед вищезгаданими кроками виконуються два додаткові — попередня обробка та аналіз індикаторів стану, що є ключовими для підвищення

точності моделі на етапах PdM шляхом створення репрезентативного набору даних для проблеми. Усі етапи мають бути розроблені, адаптовані та впроваджені відповідно до вимог випадків використання та їх характеристики даних. Крім того, розвиток систем PdM відбувається поступово, тому методи, алгоритми та рішення, прийняті на кожному етапі, впливатимуть на наступні. У наступних підрозділах розглядаються найбільш загальні методи для виконання кожного етапу PdM.

2.2.1 Передобробка

Цей крок складається з підготовки зібраних даних для подальших етапів. Кожна модель PdM відрізняється за вимогами, та їх необхідно враховувати при виборі відповідних методів попередньої обробки для підвищення продуктивності моделі. Далі коротко наведено найпоширеніші методи: валідація даних з сенсорів переконується, що зібрані дані є коректними; синхронізація індикаторів стану використовується для збору сигналів, відібраних на різних часових мітках, щоб створити дані на основі часових рядів/циклів, які легше обробляти; очищення даних видаляє або інтерполює недоступні та відсутні значення; наддискретизація застосовується для обробки дисбалансу даних, щоб підвищити точність на зазвичай дефіцитних класах даних або для роботи з невеликими наборами даних; кодування та дискретизація змінюють індикатори стану шляхом проекції на новий простір, де моделі легше їх обробляти; сегментація розбиває великий набір даних на частини для подальшого аналізу та надання можливості розпаралелювання; масштабування функцій, наприклад нормалізація або стандартизація, масштабує всі ознаки до однакового або схожого простору, для надання можливості порівняння; обробка шуму полегшує моделювання даних.

2.2.2 Аналіз індикаторів стану

Цей крок складається з виділення відповідної підмножини ознак, яка буде використовуватися як вхідні дані для моделей на наступних етапах. Це може підвищити ефективність статистичної моделі та моделі машинного навчання, незважаючи на те, що вона не є обов'язковою для моделей глибокого навчання, з огляду на це, вони можуть автоматично витягувати нові репрезентативності індикатори, які відповідають задачі. Найпоширеніші методи можна згрупувати в наступні групи: виділення індикаторів як статистичних ознак у часовій та частотній областях, які виділяють відношення час/частота індикаторів; на основі проєкції на новий простір, наприклад аналіз головних компонентів, який зменшує розмірність, зберігаючи відповідну інформацію; методи конкатенації та злиття створюють нові індикатори стану шляхом поєднання наявних; відбір ознак зменшує розмірність, відкидаючи ознаки з низькою дисперсією, надлишкові та некорельовані з метою, враховуючи той факт, що вони збільшують складність, не додаючи додаткової інформації.

2.2.3 Виявлення аномалій

Цей крок спрямований на виявлення ненормальних станів обладнання. Існує три способи проведення цього кроку за допомогою моделей класифікованих за основним завданням машинного навчання: класифікація, класифікація одного класу та кластеризація. Відповідно, їх можна використовувати, коли промаркеровані дані різних класів доступні на етапі навчання, коли існують лише

дані одного класу (зазвичай дані без збоїв) і коли дані немаркеровані. Аналіз аномалій є корисним, для того щоб отримати уявлення про можливі типи збоїв на основі експертних знань, що допомагає розробити життєвий цикл аналізу даних з визначенням пріоритетності типів збоїв або аномалій, які необхідно виявити. Для виявлення аномалій аналізують наступні показники: щільність у розмірі простору особливостей; відстань між точками даних; відношення до моделей розподілу, що відповідають навчальним даним; відношення до кластерів, створених моделями машинного навчання без учителя.

Методи виявлення аномалій потребують попередньої обробки, а деякі також залежать від індикаторів стану. Після роботи над функціями наступним кроком є вибір, навчання та оптимізація правильної моделі для випадку використання. Додаток Б класифікує та підсумовує основні методи виявлення аномалій на основі наступних робіт [9, 10, 11]. Крім того, ці методи можна поєднувати для створення моделі виявлення аномалій, що компенсує недоліки однієї моделі.

У першому стовпці методи глибокого навчання для виявлення аномалій можуть відноситись до навчання без учителя, навчання з учителем, напівкерованих і комбінованих методів, та до усіх вищезгаданих.

2.2.4 Діагностика

Як тільки аномалія була виявлена, наступний етап полягає в діагностиці того, чи відноситься аномалія до несправного робочого стану і може перерости в майбутню поломку або, навпаки, немає ризику відмови. Останній випадок вказує на те, що модель виявлення аномалій не працювала належним чином, і тому її, можливо, потрібно буде переоцінити або перенавчити. Діагноз зазвичай

ґрунтується на методах аналізу першопричин, які спрямовані на виявлення справжньої причини проблеми.

Алгоритм діагностики повинен відповідати проблемі, яка вирішується. Існує кілька підходів для вирішення цього кроку, які залежать від реалізованого методу виявлення аномалій та характеристик навчальних даних: багатокласова класифікація, бінарна класифікація, однокласова класифікація та кластеризація. Відповідні методи обираються якщо: набір даних має кілька типів збоїв, чи містить спостереження про відмовну і безвідмовну роботу, чи належать спостереження лише до одного класу та якщо спостереження проводились без нагляду.

До іншого методу діагностики відноситься кількісна оцінка відхилення аномалій за індексом здоров'я. Вона спрямована на вимірювання пошкоджень активів шляхом порівняння поточних робочих даних з історичними даними під наглядом або без нагляду. Результат оцінки може показувати або рівень погіршення в числовому шасштабі, або відсоток відхилення від нормальних робочих даних.

Крок діагностики є тим простіше, чим більше є інформації про набір даних та його мітки. Основні статистичні методи та методи машинного навчання для діагностики описані в наведеному нижче списку.

- 1) Після багатокласової класифікації для виявлення аномалій виконується діагностика на основі попередніх даних про відмову оціненого класу, тому зв'язок даних з типом відмови безпосередньо отримується з моделі;
- 2) Після бінарної класифікації для виявлення аномалій виконується кластеризація з вилученими ознаками, щоб згрупувати дані за схожістю та спробувати розрізнити непозначені типи збоїв;
- 3) Після однокласової класифікації або кластеризації для виявлення аномалій використовують методи, що визначають поріг відстані до класифікованого класу

або щільності кластерів відповідно для класифікації аномалій. Діагностику можна виконати за допомогою алгоритму кластеризації в цих показниках для аналізу внутрішньокластерних і міжкластерних відносин. Знання предметної області є важливими, щоб зв'язати невиявлені зв'язки з фізичним значенням активів, що контролюються. Ці нові знання корисні для інтерпретації результатів неконтрольованих моделей, щоб виявити нові типи збоїв, використовуючи наприклад модель К-середніх.

2.2.5 Прогнозування

Як тільки аномалію виявлено та діагностовано, еволюцію деградації можна відстежувати на основі робочих умов і стану машини в цей момент, зосереджуючись на найвпливовіших етапах діагностики, які можуть відстежувати збої. Прогнозування передбачає, коли відбудеться збій на основі поточного та минулого стану машини. Алгоритм прогнозування, як правило, оцінює залишок строку корисного використання (RUL) машини або час до відмови, аналізуючи поточний стан машини. Цей спосіб може використовувати моделювання, машинне навчання або їх поєднання для прогнозування майбутніх значень показників стану. Потім ці майбутні значення використовуються для обчислення показників RUL, які визначають, чи слід проводити технічне обслуговування і коли. Проте, якщо немає достатньо даних про деградацію, одним з небагатьох методів що залишилися є прогнозування індексу здоров'я.

Моделі прогнозування, що керовані даними, можна класифікувати на 4 групи щодо їх основного методу:

- 1) Модель на основі подібності порівнює поточну поведінку з минулою поведінкою від запуску до відмови;
- 2) Статистична модель покладається на історичну статистику для оцінки деградації, наприклад, моніторинг використання життя в поєднанні з моделями середнього часу до відмови, щоб оцінити очікувану тривалість роботи.
- 3) Аналіз часових рядів: авторегресійна інтегрована ковзна середня на основі попередніх значень, фільтр Калмана для моделювання прихованого стану шумових даних, пов'язаних з часом;
- 4) Модель навчання:
 - Класифікація діагностує дані за відомим типом відмови або подібними робочими даними, а потім спрогнозує погіршення відповідно до історичних даних цього класу. Хоча для класифікації можна взяти будь-який класифікатор, загально розповсюдженими є наступні: нейронна мережа прямого поширення, метод опорних векторів, пакетна нормалізація, алгоритм випадкового лісу;
 - Регресія: пряма оцінка індексу здоров'я, відхилення аномалії або середнього часу до відмови на основі вхідних даних. Поширеними є наступні методи: лінійні та нелінійні функції можуть моделювати нелінійні взаємозв'язки; метод опорних векторів на основі баєсовській лінійній регресії; згорткові нейронні мережі, рекурентні нейронні мережі, такі як мережа з вузлами довгої короткострокової пам'яті або вентильним рекурентним вузлом.

2.2.6 Відновлення

Як тільки аномалію виявлено, діагностовано її причину та спрогнозовано термін експлуатації верстату, є достатньо інформації для виконання заходів з

технічного обслуговування, щоб зменшити кількість збоїв на ранніх етапах і таким чином запобігти виходу з ладу обладнання. Цей етап складається з проектування та виконання кроків, необхідних для відновлення належного робочого стану обладнання до того, як виникнуть збої, що також зменшує витрати на впровадження та простої.

Відновлення виконується спеціалістами з технічного обслуговування, які відповідають за створення та реалізацію плану пом'якшення в рамках процесів управління технічним обслуговуванням та виробничою операцією. Моделі PdM повинні генерувати допоміжну інформацію, надаючи спеціалістам статистику та розпорядження щодо дій. Таким чином, більш якісне відновлення досягається за допомогою комбінації знань предметної області та інформації про стан обладнання та очікувану деградацію.

2.3 Методи глибокого навчання

Найпоширеніші методи глибокого навчання, що стосуються області PdM, узагальнено в наступних параграфах. Більшість з них засновано на схемі зворотного зв'язку, але кожен з них має свої особливості. Порівняльний аналіз методів глибокого навчання наведено у додатку В.

2.3.1 Нейронна мережа прямого поширення та багатошаровий перцептрон

Найпоширеніша і найпростіша архітектура. Вона утворена накопиченими нейронами, які створюють шари, де всі нейрони шару з'єднані з усіма нейронами

наступного шару, що подавають свій вихід на вхід до інших шарів. Однак немає зв'язків з нейронами попередніх шарів або між нейронами того самого шару.

2.3.2 Згорткова нейронна мережа

Це тип мережі з прямим зв'язком, яка підтримує сусідство нейронів шляхом застосування згорткових фільтрів. Він натхненний зоровою корою тварин і має застосування в розпізнаванні зображень і сигналів, системах рекомендацій і НЛП. Згортковий шар, як правило, є лінійним та за ним слідує застосування функції активації для отримання нелінійного результату. Після цього для зменшення розмірності можна використовувати максимальний шар або шар середнього об'єднання. Нарешті, більшість архітектур мають згладжений крок для отримання репрезентативних характеристик вхідних даних, які можна використовувати з іншими мережами машинного навчання для виконання типових завдань. Вага згортки ділиться між усіма, що полегшує їх тренування.

2.3.3 Рекурентна нейронна мережа

Цей тип моделює тимчасові дані, зберігаючи стан, отриманий з попередніх входів мережі. Алгоритм зворотного поширення в часі є адаптацією традиційного зворотного поширення для тимчасових даних, які використовуються для поширення помилки мережі на попередні екземпляри часу. Однак це поширення може призвести до проблем “зникнення” та “вибуху” градієнтів. Градієнти

можуть “зникнути” (тобто вони можуть мати тенденцію наближатись до нуля) або “вибухнути” (тобто вони можуть наближатись до нескінченості), через обчислення, що беруть участь у процесі, в яких використовуються числа з кінцевою точністю. Ця проблема змусить ці мережі забути про довгострокові відносини між нейронами. RNN, що використовують блоки LSTM, частково вирішують проблему зникаючого градієнта, оскільки вони дозволяють градієнтам протікати без змін.

2.3.4 Обмежена машина Больцмана та глибинна мережа переконань

Обмежена машина Больцмана — це дводольний, повністю зв’язний, неорієнтований граф, що складається з видимого та прихованого шарів. Це тип стохастичної ANN, яка може вивчати розподіл ймовірностей за даними. Його можна навчати з учителем та без учителя, і його основні застосування полягають у зменшенні розмірності та класифікації. У свою чергу, глибинна мережа переконань є ANN, де кожні два послідовні шари розглядаються як обмежені машини Больцмана.

2.3.5 Автокодувальник

Цей тип заснований на концепції декомпозиції одиничних значень для виділення нелінійних ознак, які найкраще представляють вхідні дані в меншому просторі. Він складається з двох частин: кодувальника, який відображає вхідні

дані в закодованій, прихованій простір, і декодувальника, який проектує дані прихованого простору в відновлений простір, який має той же розмір, що і вхідні дані. Мережа навчена мінімізувати помилку реконструкції — похибкою між входом і виходом. Автокодувальники можуть бути класифіковані відповідно до їх розмірності прихованого простору на недоповні та надповні, які відповідно відповідають прихованому простору, меншому, більшому або рівному вхідному розміру. Ці прості архітектури розширені та адаптовані для виконання різних завдань і проблем. Ванільні автокодувальники — це найпростіші автокодувальники, які належать до недоповного типу. Наступні варіації отримані шляхом застосування регуляризації та модифікації типів автокодувальника. Автокодувальник, що знижує шум використовується для відновлення пошкоджених даних. Ще одна модифікація — розріджений автокодувальник, що обмежений на етапі навчання на основі обмеженого штрафу, який ґрунтується на концепції розходження Кульбака-Лейблера.

2.3.6 Генеративні моделі

Ця група включає до себе варіативні автокодувальники та генеративні змагальні мережі. Обидві моделі були розроблені для роботи без учителя. Варіативний автокодувальник є генеративною і, отже, недетермінованою модифікацією ванільного автокодувальника, де латентний простір є безперервним.

GAN — це ще один тип генеративної нейронної мережі, яка складається з двох частин: генератора та дискримінатора. Генератор навчений генерувати вихідні дані, які належать до конкретного розподілу даних, на основі вектору представлення, що подається на вхід. Дискримінатор навчений класифікувати свої

вхідні дані незалежно від того, належать вони до певного розподілу даних чи ні. Вихід генератора підключений до входу дискримінатора, і вони тренуються разом. Метою генератора є систематичне відхилення дискримінатора шляхом генерації вихідних дані з випадкових введених даних і змушення дискримінатора класифікувати його так, як він належить до конкретного навченого розподілу. Роль дискримінатора полягає в тому, щоб розрізнити синтетичні, згенеровані дані, від несинтетичних, реальні дані від навченого розподілу. Вони навчаються разом, щоб кожна частина вчилася в іншій, змагаючись за упередження іншої частини, подібно до теорії ігор.

2.4 Методи глибокого навчання для PdM

У цьому розділі представлені основні методи глибокого навчання та його найпоширеніші архітектури, що застосовуються в області PdM. Сьогодні моделі глибокого навчання перевершують статистичні та традиційні моделі ML у багатьох областях, включаючи PdM, коли є достатньо історичних даних. Термін глибокого навчання (DL) відноситься до штучних нейронних мереж (ANN).

ANN утворені нейронами, які обчислюють лінійні регресії вхідних даних з ваговими показниками, а потім обчислюють нелінійні функції активації, такі як сигмовидна, випрямлена лінійна одиниця (ReLU) або tan-h для отримання результатів. Параметри мережі зазвичай ініціалізуються випадковим чином, а потім вони коригуються, щоб зіставити вхідні дані з вихідними даними з урахуванням набору навчальних даних. Цей процес навчання відбувається шляхом запуску алгоритму градієнтного спуску в поєднанні з алгоритмом зворотного поширення. Вони дозволяють обчислити коригування кожного

нейрона щодо помилки, створеної мережею, щоб зменшити її, де помилка розраховується на основі визначеної користувачем функції вартості.

Робота Гудфеллоу та ін. [12] надає вичерпну інформацію про DL, і багато дослідників у цій галузі вважають її зразком. А саме, книга знайомить із математичними основами машинного і глибокого навчання. Після цього зосереджується на оптимізації DL, регуляризації, різних типах архітектур, їх математичному визначенні та поширених застосуваннях. Простіший, але потужний огляд поля зроблено Літ'єнцом в огляді DL, застосованому до медицини [13], який додатково доповнюється візуальною схемою, що збирає основні архітектури. Додатково, практичне використання було описано Жероном [14] на основі таких інструментів, як Scikit-Learn [15], Keras [16] та Tensorflow[17].

2.5 Опис алгоритму

Далі наведемо хід роботи для створення моделі мережі довгої короткострокової пам'яті (LSTM) для предиктивного обслуговування. Для прогнозування RUL стає доцільним використання архітектури за типом RNN, оскільки найпростіше всього представити вхідні дані із датчиків у вигляді часового ряду.

Усі рекуррентні нейронні мережі мають вигляд ланцюга повторюваних модулів нейронної мережі. У стандартних RNN [18] (див. рис. 3) цей повторюваний модуль матиме дуже просту структуру, наприклад, один шар із функцією активації \tanh .

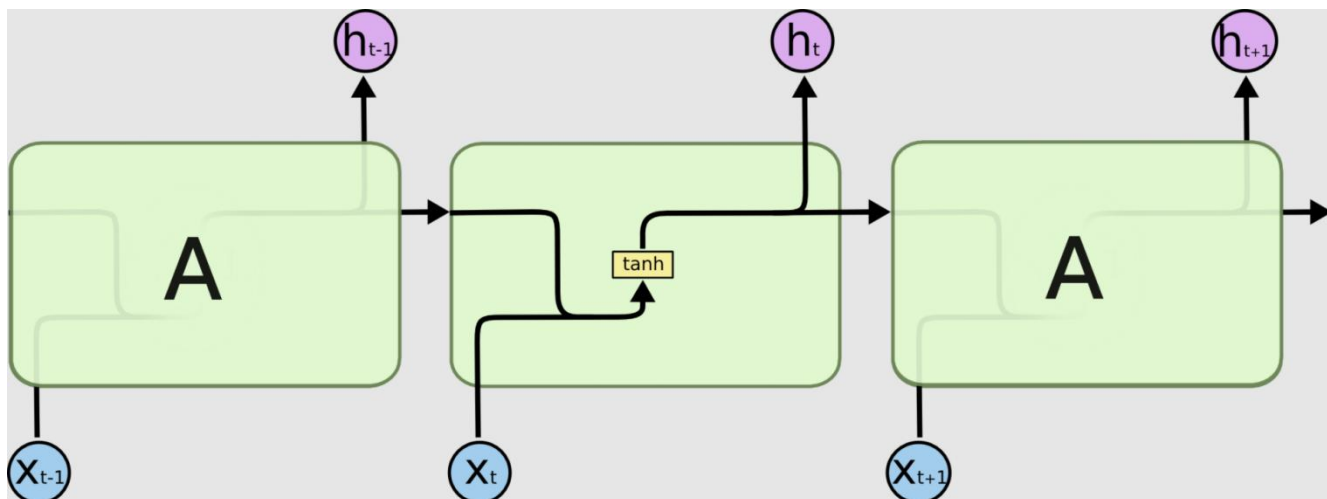


Рисунок 3 — Приклад структури RNN-мережі

Мережа довгої короткострокової пам'яті (LSTM) — різновид архітектури рекурентних нейронних мереж, запропонована в 1997 році Зеппом Хохрайтером і Юргеном Шмідхубером. На відміну від традиційних рекурентних нейронних мереж, LSTM-мережа добре пристосована до навчання на задачах класифікації, обробки і прогнозування часових рядів у випадках, коли важливі події розділені часовими лагами з невизначеною тривалістю і строками.

LSTM мережі також мають ланцюжкову структуру, але повторюваний модуль є більш складним [18] (див. рис. 4).

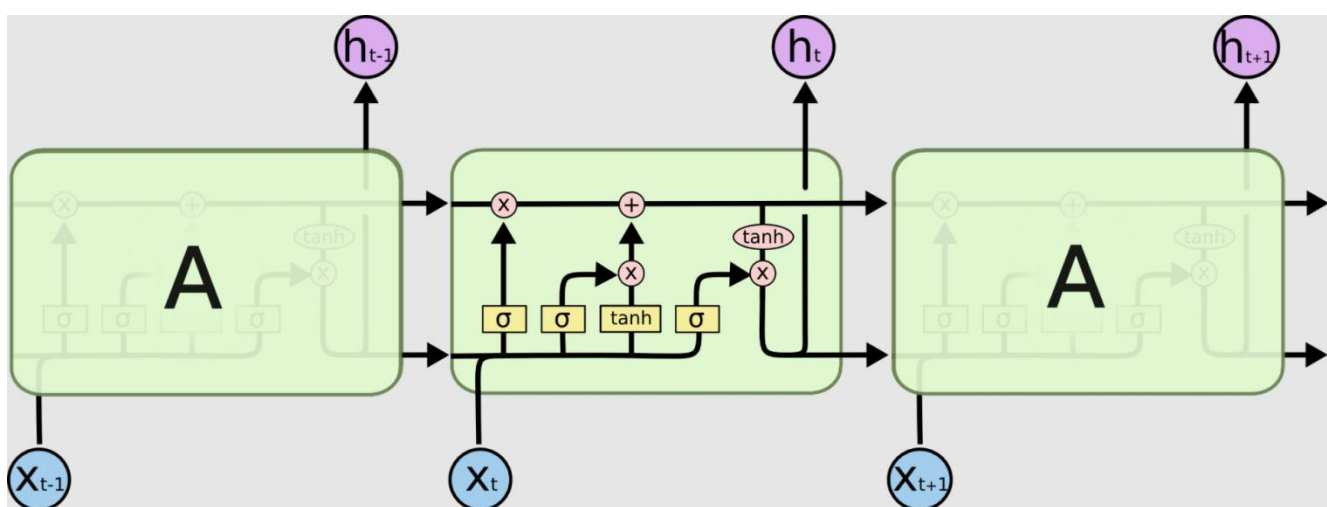


Рисунок 4 — Приклад структури LSTM-мережі

Розглянемо позначення більш детально [18] (див. рис. 5):

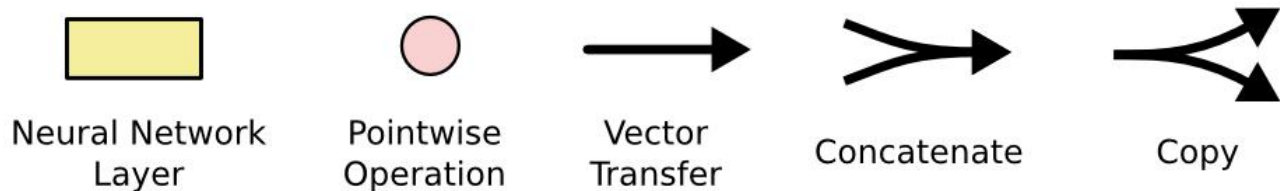


Рисунок 5 — Спеціальні позначення

На схемі кожна лінія переносить цілий вектор від виходу одного вузла до входу іншого. Рожевими кружками позначені покоординатні операції, такі, як додавання векторів, а жовті прямокутники — це навчені шари нейронної мережі. Лінії, що зливаються, означають об'єднання, а стрілки, що розгалужуються, говорять про те, що дані копіюються в різні компоненти мережі.

Ключовий компонент LSTM — це стан осередку — горизонтальна лінія, що проходить у верхній частині схеми [18] (див. рис. 6). Стан осередку нагадує конвеєрну стрічку, що проходить безпосередньо через весь ланцюжок, беручи участь лише у кількох лінійних перетвореннях. Інформація може легко проходити по ній, не змінюючись.

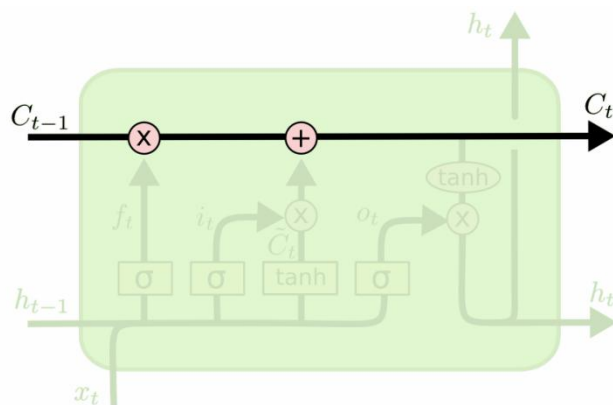


Рисунок 6 — Ключовий компонент LSTM, осередок

Тим не менш, LSTM може видаляти інформацію із осередку. Цей процес регулюється структурами, які називаються фільтрами. Фільтри дозволяють пропускати інформацію на підставі певних умов. Вони складаються з шару сигмоїдальної нейронної мережі та операції поточкового множення. Сигмоїдальний шар повертає числа від нуля до одиниці, що позначають, яку частку кожного блоку інформації слід пропустити далі через мережу. Нуль означає “не пропускати нічого”, одиниця — “пропустити все”. У LSTM є три такі фільтри.

Перший крок у LSTM [18] (див. рис. 7) — визначити, яку інформацію можна викинути зі стану комірки. Це рішення приймає сигмоїдальний шар, що називається “шаром фільтра забування”. Він дивиться на h_{t-1} і x_t і повертає число від 0 до 1 для кожного числа зі стану комірки C_{t-1} . Одиниця означає "повністю зберегти", а нуль — "повністю викинути".

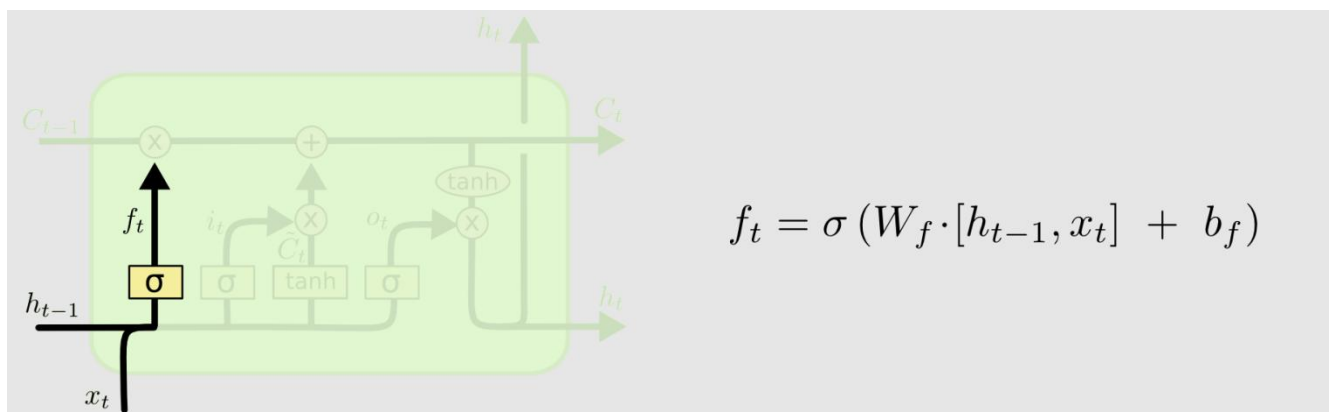


Рисунок 7 — Перший логічний крок LSTM

Наступний крок [18] (див. рис. 8) вирішує яка нова інформація буде зберігатися у стані комірки. Цей етап складається з двох частин. Зпочатку сигмоїдальний шар під назвою “шар вхідного фільтра” визначає які значення слід

оновити. Потім \tanh -шар будує вектор нових значень-кандидатів \tilde{C}_t , що можна потім додати у стан комірки.

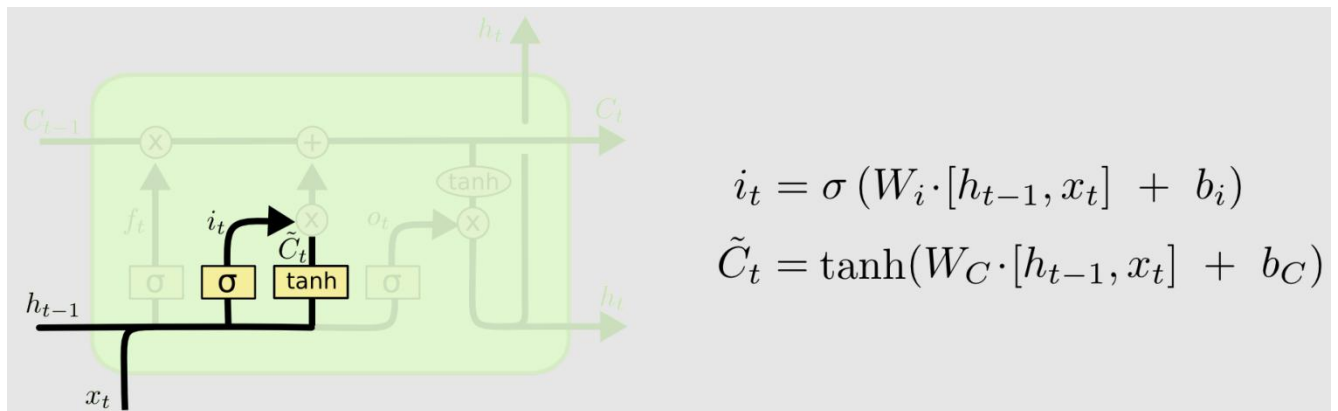


Рисунок 8 — Другий логічний крок LSTM

Потім старе значення стану комірки C_{t-1} замінюється на новий стан C_t .

На наступному кроці [18] (див. рис. 9) старий стан помножується на f_t та попередній стан забувається. Потім до цього прибавляється $i_t * \tilde{C}_t$.

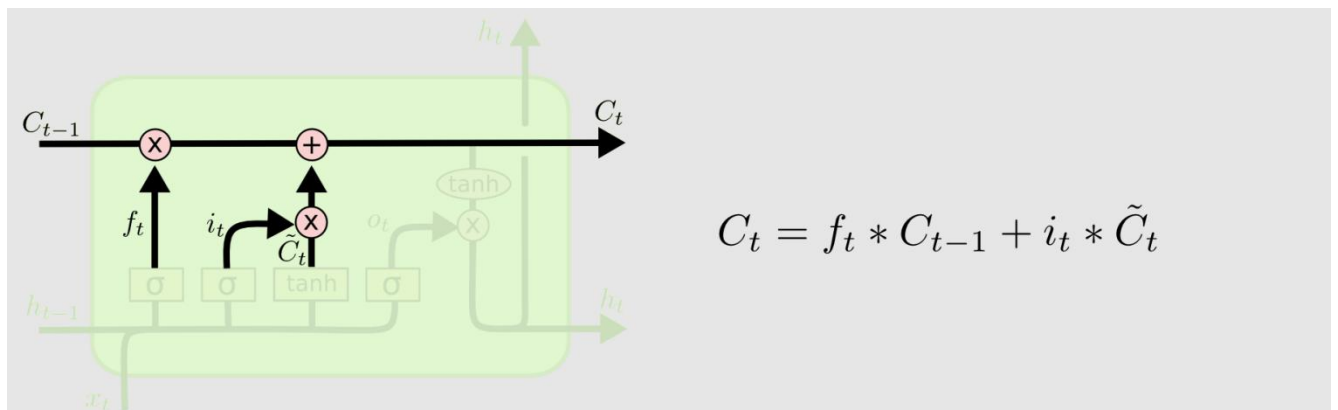


Рисунок 9 — Третій логічний крок LSTM

Наслідок потрібно вирішити яку інформацію передавати на вихід. Вихідні дані будуть засновані на стані осередку, до якого будуть використані деякі фільтри [18] (див. рис. 10).

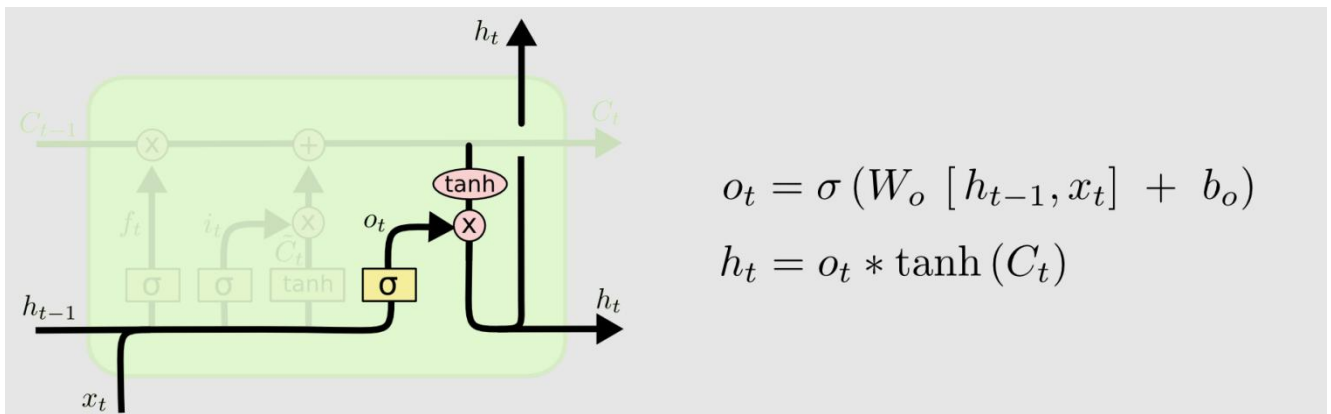


Рисунок 10 — Останній логічний крок LSTM

На цьому кроці застосовується сигмоїдальний шар, який вирішує, яку інформацію зі стану осередку виводиться. Потім значення стану осередку проходить через tanh-шар, щоб отримати на виході значення діапазону від -1 до 1, і перемножується з вихідними значеннями сигмоїдального шару, що дозволяє виводити тільки необхідну інформацію.

2.6 Практична реалізація

Далі наведемо хід роботи для створення рішення із використанням мережі довгої короткострокової пам'яті (LSTM) для предиктивного обслуговування авіаційних двигунів.

2.6.1 Прийом даних та їх передобробка

У даному розділі ми використовуємо навчальні, тренувальні та тестові набори даних NASA Turbofan Jet Engine Data Set [19], що знаходяться у відкритому доступі. Дані навчання складаються з кількох багатовимірних часових рядів із циклом як одиницею часу, разом із 21 показаннями датчиків та 3 налаштуваннями для кожного циклу. Можна вважати, що кожен часовий ряд

створений з іншого механізму того самого типу. Дані тестування мають ту ж схему даних, що й навчальні дані. Єдина відмінність полягає в тому, що дані не вказують, коли відбувається збій. Наслідок, основні правдиві дані надають кількість робочих циклів, що залишилися для двигунів у даних тестування.

Навчальні дані складаються із показників датчиків від 100 двигунів (колонка id) у вигляді багатовимірних часових рядів із колонкою cycle як одиницею часу з 21 показаннями датчика від s1 до s21 та 3 функціями робочих налаштувань (колонка settings) для кожного циклу. У цих змодельованих даних передбачається, що двигун працює нормально на початку кожного часового ряду. Погіршення двигину прогресує і зростає до тих пір, поки не буде досягнуто попередньо визначеного порогу, коли двигун вважається небезпечним для подальшої роботи. У цьому моделюванні останній цикл у кожному часовому ряді можна розглядати як точку відмови відповідного двигуна.

Далі створюються мітки для даних навчання. Оскільки останнє спостереження вважається точкою відмови, ми можемо обчислити залишок корисного часу (RUL) для кожного циклу в даних:

```
w1 = 30
w0 = 15
train_df['label1'] = np.where(train_df['RUL'] <= w1, 1, 0 )
train_df['label2'] = train_df['label1']
train_df.loc[train_df['RUL'] <= w0, 'label2'] = 2
```

У шаблоні прогнозного технічного обслуговування стовпець cycle також використовується для навчання, тому ми також включимо його. Далі нормалізуємо стовпці в даних навчання за допомогою min-max нормалізації:

```
train_df['cycle_norm'] = train_df['cycle']
```

```

cols_normalize =
train_df.columns.difference(['id', 'cycle', 'RUL', 'label1', 'label2'])
min_max_scaler = MinMaxScaler()
norm_train_df =
pd.DataFrame(min_max_scaler.fit_transform(train_df[cols_normalize]),
              columns=cols_normalize,
              index=train_df.index)

join_df =
train_df[train_df.columns.difference(cols_normalize)].join(norm_train_df)
train_df = join_df.reindex(columns = train_df.columns)
train_df.head()

```

2.6.2 Створення моделі нейронної мережі та її оцінка

Побудова нейронної мережі вимагає визначення архітектури мережі. У даній роботі буде створено мережу із 6 шарами, з яких 4 є прихованими. Перший шар LSTM із 100 одиницями, по одній для кожної вхідної послідовності, за яким слідує шар випадання у 20%. Потім шар LSTM із 50 одиницями, за яким знову шар випадання у 20%. Шари випадання було застосовано щоб контролювати надмірне навчання. Останній щільний вихідний шар використовує сигмоїдну функцію активації, що відповідає вимогам бінарної класифікації — чи вийде обладнання з ладу за певний встановлений проміжок часу. У якості функції втрати для мережі використовується функція бінарної кросентропії. У якості алгоритму оптимізації для тренування моделі використовується алгоритм adam.

Наведемо схему мережі у вигляді UML-діаграми [20].

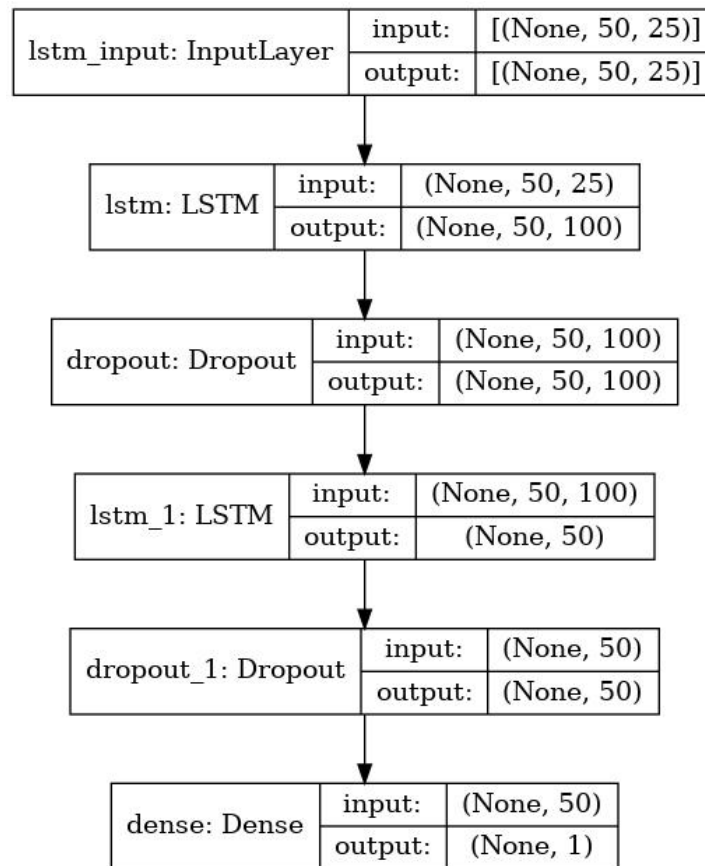


Рисунок 11 — Схема шарів нейронної мережі

Далі наведемо код моделі:

```

model = Sequential()
model.add(LSTM(
    input_shape=(sequence_length, nb_features),
    units=100,
    return_sequences=True))
model.add(Dropout(0.2))
model.add(LSTM(
    units=50,
    return_sequences=False))
model.add(Dropout(0.2))
model.add(Dense(units=nb_out, activation='sigmoid'))
  
```

```
model.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['accuracy'])
```

Для оцінки тренувального набору даних використовується матриця невідповідностей даної моделі. Точні прогнози лежать уздовж діагоналі матриці, похибки — на позадіагоналі.

Наведемо приклад коду для оцінки:

```
y_pred = model.predict_classes(seq_array, verbose=1, batch_size=200)
y_true = label_array
cm = confusion_matrix(y_true, y_pred)
```

Результат даної оцінки було виведено у консоль:

```
Training Confusion matrix
- x-axis is true labels.
- y-axis is predicted labels
array([[12448,    83],
       [  328,  2772]])
```

Оскільки точні прогнози складають більшу частину, також можна обчислити влучність та повноту моделі на стадії тренування. У результаті влучність складає 0.97, а повнота складає 0.89.

```
precision = precision_score(y_true, y_pred)
recall = recall_score(y_true, y_pred)
f1 = 2 * (precision * recall) / (precision + recall)
```

```
Training precision = 0.9709281961471103
Training recall   = 0.8941935483870967
```

Далі було розглянуто продуктивність моделі на тестовому наборі даних. Для цілей тестування зберігаються лише дані останнього циклу для кожного ідентифікатора двигуна в даних тестування. Щоб порівняти результати з шаблоном, ми вибираємо останню послідовність для кожного ідентифікатора в даних тесту:

```
y_mask = [len(test_df[test_df['id']==id]) >= sequence_length
for id in test_df['id'].unique()]
label_array_test_last = test_df.groupby('id')['label1'].nth(-1)
[y_mask].values
label_array_test_last =
label_array_test_last.reshape(label_array_test_last.shape[0],1).astype
(np.float32)
```

Далі перевіримо модель на тестових даних: порівняємо точність моделі на тестовому наборі з точністю на наборі навчання. За визначенням, точність навчання має бути оптимістичною, оскільки модель була оптимізована для цих спостережень. Точність тестового набору є більш загальною і моделює те, як модель передбачалося використовувати для прогнозування вперед у часі. Це число використовується для звіту про ефективність моделі та складає 0,96:

```
scores_test = model.evaluate(seq_array_test_last,
label_array_test_last, verbose=2)
```

```
Accuracy: 0.9569892415436365
```

Мережа LSTM складається з двох компонентів: архітектури та вагів моделі. Ці компоненти буде збережено у двох файлах. Архітектуру буде збережено у файлі json, який пакет keras може використовувати для перебудови моделі. Ваги

моделі буде збережено у форматі HDF5, що використовується для перебудови точної моделі.

ВИСНОВКИ

В результаті роботи було проведено аналіз та моделювання предметної області, досліджено методи та моделі машинного навчання, вивчено особливості архітектур нейронних мереж для подальшої програмної реалізації, програмно реалізовано прогнозування критичного стану обладнання. Об'єктом дослідження була задача прогнозування критичного стану обладнання з використанням машинного навчання з метою підвищення ефективності виробництва на підприємствах. Методи розробки базуються на технологіях Python, Tensorflow, Keras. Дані з датчиків обладнання отримано з відкритих джерел.

У теоретичній частині роботи буде вивчено найбільш релевантні методи машинного навчання, що орієнтовані на дані для застосування у контексті сфери предиктивного обслуговування, розглядається придатність моделей глибокого навчання і порівнюються їх переваги та недоліки зі статистичними та класичними моделями машинного навчання. Спочатку було проаналізовано контекст PdM та сферу його застосування. Після цього були досліджені різні типи моделей. Потім було проаналізовано моделі, що засновуються на даних. Нарешті, було ретельно переглянуто моделі глибокого навчання. Ця методологія дозволила отримати загальне уявлення про обсяг, а потім зосередитися на конкретних темах дослідження. Крім того, аналіз сучасних методів дозволив порівняти методи та обговорити проблеми та перспективи моделей DL для PdM.

У практичній частині було реалізовано алгоритм нейронної мережі із блоками LSTM для прогнозування RUL авіаційних двигунів. Отримана точність прогнозування складає 96%.

ПЕРЕЛІК ДЖЕРЕЛ

1. Д. С. Кокорев, А. А. Юрин. Цифровые двойники: понятие, типы и преимущества для бизнеса // Colloquium-journal. 2019. №10 (34). URL: <https://cyberleninka.ru/article/n/tsifrovye-dvoyniki-ponyatie-tipy-i-preimuschestva-dlya-biznesa> (дата звернення: 10.12.2021).
2. Голян В. В., Бітюкова Є. І. Дослідження технологій та інструментів управління проектами // Монографія: SCIENCE, RESEARCH, DEVELOPMENT#27/ NECHNICS AND TECHNOLOGY – 2020. – 33 с.
3. Четвёртая промышленная революция: интернет вещей, циркулярная экономика и блокчейн [Электронный ресурс] // [furfur.me](http://www.furfur.me). FURFUR - ежедневное молодёжное интернет-издание. URL: <http://www.furfur.me/furfur/changes/changes/216447-4-aya-promyshlennaya-r-evolyutsiya> (дата звернення: 9.12.2021).
4. И. Н. Антоненко. Методология RCM: ретроспектива и перспектива надёжностно-ориентированного технического обслуживания [Электронный ресурс] // trim.ru. НПП СпецТек | Системы управления активами. URL: https://trim.ru/sites/default/files/files/pdf/rcm_methodology_history.pdf (дата звернення: 10.12.2021).
5. International Society of Automation. Improving maintenance by adopting a P-F curve methodology [Электронный ресурс]. URL: <https://www.isa.org/intech-home/2019/march-april/features/improving-maintenance-by-adopting-a-p-f-curve-meth> (дата звернення: 15.02.2022).
6. Venkat Venkatasubramanian, Raghunathan Rengaswamy, Kewen Yin, and Surya N. Kavuri. 2003. A review of process fault detection and diagnosis part I: Quantitative model-based methods. *Computers and Chemical Engineering* 27, 3 (2003), 293–311.

7. Pramod K. Behera, Bhabani S. Sahoo. (2016) Leverage of Multiple Predictive Maintenance Technologies in Root Cause Failure Analysis of Critical Machineries [Текст] // *Procedia Engineering*. Volume 144, P. 351-359.
8. Mitchell Lebold, Karl Reichard, Carl S Byington, and Rolf Orsagh. 2002. OSA-CBM architecture development with emphasis on XML implementations. In *Maintenance and Reliability Conference (MARCON)*. pp. 6–8.
9. Thyago P. Carvalho, Fabrizzio A.A.M.N. Soares, Roberto Vita, Roberto da P. Francisco, João P. Basto, and Symone G.S. Alcalá. 2019. A systematic literature review of machine learning methods applied to predictive maintenance. *Computers and Industrial Engineering* 137 (2019), 106024.
10. Marco A.F. Pimentel, David A. Clifton, Lei Clifton, and Lionel Tarassenko. 2014. A review of novelty detection. *Signal Processing* 99 (2014), 215–249.
11. Weiting Zhang, Dong Yang, and Hongchao Wang. 2019. Data-Driven Methods for Predictive Maintenance of Industrial Equipment: A Survey. *IEEE Systems Journal* 13, 3 (2019), 2213–2227.
12. Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. [Электронный ресурс]. URL: <http://www.deeplearningbook.org> (дата звернення: 15.02.2022).
13. Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghahfoorian, Jeroen A.W.M. van der Laak, Bram van Ginneken, and Clara I. Sánchez. 2017. A survey on deep learning in medical image analysis. *Medical Image Analysis* 42 (2017), 60–88.
14. Aurélien Géron. 2017. *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems*. [Текст] // O'Reilly Media. 572 pages.
15. Scikit-Learn. *Machine Learning in Python* [Электронный ресурс]. URL: <https://scikit-learn.org/> (дата звернення: 15.02.2022).

16. Keras. Deep learning for humans. [Электронный ресурс]. URL: <https://keras.io/> (дата звернения: 15.02.2022).
17. Tensorflow. Комплексная платформа машинного обучения с открытым исходным кодом. [Электронный ресурс]. URL: <https://www.tensorflow.org/> (дата звернения: 15.02.2022).
18. Colah's blog [Электронный ресурс]. URL: <https://colah.github.io/> (дата звернения: 15.02.2022).
19. Kaggle. NASA Turbofan Jet Engine Data Set [Электронный ресурс]. URL: <https://www.kaggle.com/datasets/behrad3d/nasa-smaps> (дата звернения: 15.02.2022).
20. Jordan K. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. – NY: «Williams», 2006. – 736 с.