Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук
(повна назва)

Кафедра програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів виявлення аномалій даних у вимірюваннях пристроїв IoT

(тема)

Виконав:
студент (ка) 2 курсу, групи ІПЗм-22-3

Васильєв І.А.
(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва спеціальності)

Тип програми освітньо-наукова

Керівник доц. Хацько Н.Є.
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри _____ З.В.Дудар
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерних наук
Кафедра програмної інженерії
Рівень вищої освіти другий (магістерський)
Спеціальність 121 – Інженерія програмного забезпечення
Тип програми освітньо-наукова програма
Освітня програма Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«___» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Васильєву Ігорю Антоновичу
(прізвище, ім’я, по батькові)

1. Тема роботи «Дослідження методів виявлення аномалій даних у вимірюваннях пристроїв IoT»

Затверджена наказом по університету від 29.03. 2024р. № 250 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 20.06.2024

3. Вихідні дані до роботи ist-2023 «Exploring LSTM with Attention for Anomaly Detection»

4. Перелік питань, що потрібно опрацювати в роботі

мета роботи, аналіз предметної галузі і постановка задачі, огляд та аналіз літературних джерел з дослідження,дослідження теоретичне, дослідження практичне

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Інструктаж з техніки безпеки	1-й тиждень	Виконано
2	Ознайомлення зі структурою та планом роботи, отримання завдання	1-й тиждень	Виконано
3	Виконання індивідуального завдання відповідно до теми дослідження кваліфікаційної роботи магістра	1 - 9 – й тижні	Виконано
4	Підготовка звіту	9 - 10 й тижні	Виконано
5	Захист	10 - й тиждень	Виконано

Дата видачі завдання __01__ ___квітня____2024 р.

Студент _____ __Ігор Васильєв_____
(підпис)

Керівник роботи _____ _доцент Хацько Наталія Євгенівна
(підпис)

РЕФЕРАТ/ABSTRACT

Звіт: 100 с., 7 рис., 3 табл, 19 джерел.

ВИЯВЛЕННЯ АНОМАЛІЙ, КЕРУВАННЯ МЕТОДИ АНАЛІЗУ ДАНИХ, МЕХАНІЗМ УВАГИ, ОПТИМІЗАЦІЯ ДАНИХ, IoT, LSTM, IoT.

Об’єкт дослідження – Механізми детекції аномалій.

Мета роботи – Дослідження механізмів детекції аномалій, впровадження їх до IOT середовища, підвищення ефективності аналізу аномалій в роботі із складними даними, підготовка до інтеграції цих методів до робочого середовища.

Результат роботи – розроблена перша частина магістерського дослідження.

LSTM, ANOMALITY DETECTION, IoT, DATA ANALYSIS METHODS, ATTENTION MECHANISM, IoT MANAGEMENT, DATA OPTIMIZATION.

Object of research – Mechanisms of anomaly detection.

The purpose of the work is to study anomaly detection mechanisms, their introduction to the IOT environment, increase the efficiency of anomaly analysis in working with complex data, preparation for the integration of these methods into the working environment.

The result of the work is the development of the first part of the master's research.

Я, Васильєв Ігор Антонович, студент гр. ІПЗм-22-3, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження методів виявлення аномалій даних у вимірюваннях пристроїв IoT», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу

ElArKhNURE. 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

ВСТУП

У розгалуженому просторі розумних операцій виявлення аномалій стає ключовим фактором для забезпечення оптимальної продуктивності, мінімізації простоїв і запобігання потенційним катастрофічним поломкам. Традиційні централізовані системи виявлення аномалій борються з такими постійними проблемами, як затримка, обмеження масштабованості та вразливість до окремих точок збою. Щоб подолати ці перешкоди та задовольнити постійні потреби сучасних галузей промисловості, потрібні інноваційні рішення.

Виявлення аномалій має вирішальне значення в розумних умовах, щоб запобігти дорогим простоям, пошкодженню обладнання та загрозам безпеці. Інтеграція методів виявлення аномалій пропонує багатообіцяючий шлях для підвищення точності та ефективності систем виявлення аномалій, відкриваючи шлях для проактивного обслуговування, прогнозної аналітики та прийняття рішень у реальному часі.

Сфера застосування цієї системи охоплює комплексний моніторинг і аналіз даних датчиків Інтернету речей на різноманітних підприємствах та в побуті. Завдяки інтеграції методів виявлення аномалій система має на меті ідентифікувати ненормальні моделі та відхилення в даних датчиків у реальному часі, що дозволяє своєчасно втручатися та оптимізувати розумні процеси. Крім того, система прагне адаптувати різні типи даних, поширені в середовищах IoT, включаючи дані часових рядів, текстову інформацію та числові показники. Завдяки адаптивності та масштабованості система може задовольнити різноманітні розумні застосування, починаючи від виробництва та виробництва енергії до охорони здоров’я та транспорту.

У розгалуженому просторі розумних операцій ця система служить наріжним каменем для проактивного управління та оптимізації. Надаючи інформацію про продуктивність обладнання, умови навколишнього середовища та ефективність процесу в режимі реального часу, система дає змогу зацікавленим сторонам приймати обґрунтовані рішення та запобігати потенційним проблемам.

Крім того, здатність системи виявляти складні аномалії додає додатковий рівень стійкості до розумних операцій, захищаючи від несподіваних збоїв і підвищуючи загальну продуктивність і безпеку.

Можливості, які пропонує інтеграція методів виявлення аномалій у системи виявлення аномалій для середовищ IoT, різноманітні. По-перше, цей підхід дозволяє визначити пріоритетність важливих даних датчиків, сприяючи точнішому розумінню та прийняттю рішень серед складнощів різноманітних показань датчиків. Зосереджуючись на відповідній інформації, методи виявлення аномалій покращують виявлення незначних відхилень від нормальної поведінки, тим самим зменшуючи ризик відмови обладнання та оптимізуючи ефективність роботи. Крім того, універсальність методів виявлення аномалій у пристосуванні до різних типів даних, поширених у сценаріях Інтернету речей, включаючи показники на основі часу, текстові, візуальні та числові дані, підкреслює їх адаптивність до різноманітних розумних застосувань. Ця адаптивність має ключове значення, враховуючи динамічну природу даних Інтернету речей у реальному часі, де швидкі та обґрунтовані процеси прийняття рішень мають першочергове значення. Загалом інтеграція методів виявлення аномалій має величезні перспективи для революції у виявленні аномалій у розумних середовищах Інтернету речей, пропонуючи неперевершені знання та можливості для керування та захисту складних систем.

Для підтвердження надійності та ефективності цієї системи буде розроблено прототип, який буде перевірено в реальних життєвих умовах. Прототип охоплюватиме апаратні компоненти, такі як датчики IoT і системи збору даних, а також програмні модулі для обробки даних, виявлення аномалій і візуалізації. Розгорнувши прототип у різноманітних життєвих умовах, дослідники можуть оцінити його продуктивність за різних умов і підтвердити його ефективність у виявленні аномалій і полегшенні профілактичного обслуговування. Крім того, прототип слугуватиме випробувальним майданчиком для вдосконалення алгоритмів, оптимізації параметрів системи та підвищення загальної надійності та масштабованості системи.

1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1. Опис предметної області

Предметна область, що розглядається, окреслює складну взаємодію розширеного програмного забезпечення для виявлення аномалій, програмного забезпечення для керування веб-сайтами та програмного забезпечення для керування Інтернетом речей у розширеному домені Інтернету речей (IoT).

Використання технології IoT створює складне середовище для праці, і надійні механізми безпеки, зокрема виявлення аномалій, відіграють ключову роль у захисті критичної інфраструктури.

Середовище Інтернету речей часто включає складні програмні системи, обладнання та процеси. Пристрої IoT інтегровані в цей ландшафт для моніторингу, контролю та оптимізації різноманітних розумних операцій. Програмна система менеджменту фокусується на захисті цих взаємопов’язаних пристроїв.

Виявлення аномалій передбачає виявлення моделей і поведінки, які відхиляються від норми. У контексті IoT для аналізу даних від датчиків і пристроїв використовуються складні механізми виявлення аномалій, такі як LSTM з механізмами уваги. Ці механізми можуть розпізнавати незначні відхилення, що вказують на потенційні загрози безпеці або порушення в роботі.

Впровадження механізмів виявлення аномалій значно підвищує рівень безпеки розумних систем. Завдяки безперервному моніторингу даних з пристроїв IoT ці механізми можуть виявляти аномалії, які можуть означати кібератаки, несправності обладнання або несанкціонований доступ.

Виявлення аномалій служить проактивним захисним механізмом, що дозволяє завчасно ідентифікувати аномальні моделі. Це дає змогу оперативно реагувати на пом’якшення потенційних загроз до їх ескалації. Система може автоматично запускати сповіщення, ізолювати скомпрометовані пристрої або ініціювати попередньо визначені протоколи безпеки.

Забезпечення стабільності та надійності розумних операцій має першочергове значення. Виявлення аномалій сприяє підтримці безперервності

роботи, запобігаючи збоям, викликаним кіберзагрозами, збоями обладнання або непередбаченими аномаліями. Це особливо критично в секторах, де простої можуть мати серйозні наслідки для економіки та безпеки.

Такі системи часто стикаються з динамічними та змінними умовами. Механізми виявлення аномалій повинні адаптуватися до середовища, що змінюється, повинні бути здатними навчатися та пристосовуватися до нових шаблонів, а також забезпечувати ефективність системи перед лицем нових загроз.

Неможливо переоцінити важливість впровадження надійних механізмів безпеки, зокрема виявлення аномалій, у сфері IoT. Поламки в системах Інтернету речей можуть призвести до компрометації даних, збоїв у виробництві, загрози безпеці та фінансових втрат. Механізми безпеки не тільки захищають активи, але й сприяють загальній стійкості та надійності інфраструктури IoT.

У багатьох галузях дотримання нормативних стандартів є обов'язковим. Механізми безпеки, включаючи виявлення аномалій, відіграють вирішальну роль у виконанні цих вимог відповідності. Демонстрація відданості заходам безпеки підвищує довіру між зацікавленими сторонами та регуляторними органами.

Системи виявлення аномалій сприяють культурі постійного вдосконалення безпеки розумних систем. Аналізуючи історичні дані та визначаючи зони вразливості, організації та приватні особи можуть постійно покращувати свої заходи безпеки, щоб випереджати нові загрози.

На завершення предметна область зосереджена на критичній ролі механізмів виявлення аномалій у зміцненні безпеки розумних пристроїв Інтернету речей. Вирішуючи унікальні виклики розумного середовища, ці механізми сприяють стійкості, надійності та безпеки критичної інфраструктури. Їх впровадження є невід'ємною частиною захисту розумних процесів і підтримки надійності в епоху взаємопов'язаних інтелектуальних систем.

1.2. Актуальність теми дослідження

Дослідження механізмів виявлення аномалій у приватних і бізнес-пристроях IoT мають значну актуальність у технологічному ландшафті, що розвивається, і

зростаючій інтеграції IoT у складні середовища. Кілька ключових аспектів підкреслюють важливість і застосовність цього дослідження.

Виклики безпеки в розумних середовищах.

Розумні системи є основними цілями для кіберзагроз і атак, що вимагає розробки ефективних механізмів виявлення аномалій, щоб захистити критичну інфраструктуру від потенційних порушень безпеки, які сприяють інтеграції пристроїв IoT.

Вплив аномалій на безперервність роботи.

Аномалії, викликані кібератаками або збоями, можуть порушити бізнес-операції, що підкреслює необхідність використання вдосконалених механізмів виявлення, щоб мінімізувати час простою, запобігти втратам виробництва та забезпечити безперебійність бізнес-процесів.

Ефективність використання ресурсів та оптимізація витрат.

Раннє виявлення аномалій дозволяє ефективніше розподіляти ресурси, пом’якшуючи фінансові наслідки простою, пошкодження обладнання та витрати на відновлення, пов’язані з порушеннями безпеки або ненормальними режимами роботи.

Дотримання галузевих стандартів.

Дослідження механізмів виявлення аномалій узгоджуються з галузевими стандартами та правилами, забезпечуючи відповідність і демонструючи прихильність до захисту даних і кібербезпеки в розумному, приватному та бізнес-середовищі.

Безпека людини та збереження навколишнього середовища.

Порушення безпеки в інтелектуальних системах становлять загрозу як для цілісності даних, так і для безпеки людей, а також потенційні наслідки для навколишнього середовища. Ефективні механізми виявлення аномалій сприяють безпеці персоналу та запобіганню екологічних інцидентів.

Інтеграція передових технологій.

Дослідження досліджує інтеграцію передових технологій, таких як LSTM з механізмами уваги, у процеси виявлення аномалій, що відображає ширшу

тенденцію використання інноваційних рішень для вирішення проблем кібербезпеки в розумних середовищах.

Адаптивність до динамічних середовищ.

Інтелектуальне середовище є динамічним, що потребує механізмів виявлення аномалій, які можуть адаптуватися до процесів, технологій і умов експлуатації, що розвиваються, забезпечуючи стійкість і ефективність заходів безпеки проти нових загроз.

Внесок в академічні та практичні знання.

Дослідження в цій галузі покращують як академічне розуміння, так і практичні застосування, надаючи інформацію, яка інформує практиків галузі та сприяє подальшому прогресу в кібербезпеці в розумному, приватному та бізнес-контекстах.

Глобальний економічний вплив.

Враховуючи взаємопов’язаний характер глобальних галузей використання таких систем, порушення безпеки в одному регіоні можуть мати масштабні економічні наслідки. Надійні механізми виявлення аномалій зміцнюють загальну стійкість розумних екосистем, пом’якшуючи потенційні економічні наслідки масових інцидентів безпеки.

Підсумовуючи, дослідження механізмів виявлення аномалій у приватних і бізнес-пристроях IoT є ключовими для вирішення нагальних проблем безпеки, забезпечення безперервності роботи та підвищення безпеки та ефективності процесів. Це узгоджується з більш широкими цілями сприяння безпечним і стійким системам серед швидкого розвитку технологій.

1.3. Виявлення проблем та актуалізація рішень

Проблема: вразливість безпеки в IoT системах.

Рішення: розробити передові механізми виявлення аномалій, які можуть ідентифікувати та пом’якшувати загрози безпеці в режимі реального часу. Застосувати шифрування та безпечні протоколи зв’язку для захисту цілісності даних під час передачі.

Проблема: збої в роботі через аномалії.

Рішення: Дослідити моделі виявлення аномалій, здатні передбачити та запобігти аномаліям, які можуть призвести до збоїв у роботі. Інтегрувати автоматичні механізми реагування, щоб мінімізувати час простою та оптимізувати використання ресурсів.

Проблема: відсутність адаптації до динамічного середовища.

Рішення: дослідити моделі машинного навчання, зокрема LSTM із механізмами уваги, які можуть адаптуватися до мінливих життєвих умов. Розробити алгоритми, здатні до безперервного навчання та адаптації до нових закономірностей і аномалій.

Проблема: недостатня відповідність галузевим стандартам.

Рішення: Створити систему виявлення аномалій, яка відповідає галузевим стандартам і нормам безпеки та вдосконалює їх. Впровадити контрольні стежки та механізми звітності, щоб продемонструвати відповідність і забезпечити прозорість.

Проблема: Безпека людини та екологічні ризики.

Рішення: інтегрувати моделі виявлення аномалій, які не лише зосереджуються на кібербезпеці, але й враховують безпеку людини та вплив на навколишнє середовище. Розробити системи раннього попередження про можливі інциденти, які можуть становити загрозу для персоналу або навколишнього середовища.

Проблема: ресурсомісткі моделі виявлення аномалій.

Рішення: оптимізувати алгоритми виявлення аномалій для підвищення ефективності та використання ресурсів. Дослідити периферійні обчислювальні рішення для виявлення аномалій ближче до джерела, зменшуючи навантаження на центральні системи та покращуючи реакцію в режимі реального часу.

Проблема: обмежена інтеграція передових технологій.

Рішення: додати інтеграцію передових технологій, як-от LSTM, із механізмами уваги в складні структури безпеки Інтернету речей. Розробити

інфраструктури та бібліотеки для полегшення впровадження передових методів виявлення аномалій.

Проблема: відсутність сумісності між пристроями та між галузями.

Рішення: розробити механізми виявлення аномалій, сумісні з різноманітними розумними пристроями IoT і адаптовані до різних галузевих умов. Сприяти співпраці та стандартизації для створення єдиного підходу до виявлення аномалій.

Проблема: недостатня освіта та обізнаність.

Рішення: Вирішити проблему недостатньої обізнаності серед професіоналів щодо важливості виявлення аномалій. Розробити освітні програми та ресурси для поширення знань про ризики, пов’язані з пристроями IoT, і переваги надійного виявлення аномалій.

Проблема: неадекватні плани реагування на інциденти.

Рішення: розробити комплексні плани реагування на інциденти, які включають протоколи для обробки аномалій, виявлених системою. Проводити регулярні тренування та моделювання, щоб переконатися, що персонал добре підготовлений для ефективного реагування на інциденти безпеки.

Проблема: баланс безпеки та продуктивності.

Рішення: знайти баланс між надійними заходами безпеки та мінімальним впливом на продуктивність системи. Оптимізувати алгоритми для обробки в реальному часі без шкоди для ефективності виявлення аномалій.

Проблема: відсутність механізмів постійного вдосконалення.

Рішення: запровадити механізми для постійного вдосконалення виявлення аномалій. Розвивайте цикли зворотного зв’язку, які дозволяють системі навчатися на виявлених аномаліях, покращуючи її здатність адаптуватися до загроз, що розвиваються з часом.

Проблема: Нестабільність і низька ефективність алгоритмів виявлення аномалій при великому обсязі даних.

Рішення: Розробити алгоритми, які оптимально працюють з великими обсягами даних, використовуючи техніки паралельного обчислення та розподіленого аналізу для забезпечення стабільної та швидкої реакції на аномалії.

Проблема: Відсутність стандартизованих протоколів комунікації між різними пристроями Інтернету речей.

Рішення: Розробити та впровадити універсальні стандарти комунікації, які сприятимуть взаємодії між різнорідними пристроями IoT, полегшуючи інтеграцію та роботу систем виявлення аномалій.

Вирішення цих виявлених проблем і реалізація запропонованих рішень сприятимуть ефективності, результативності та стійкості механізмів виявлення аномалій у розумних пристроях Інтернету речей, забезпечуючи безпечне та надійне середовище для керування системами Інтернету речей.

2. ПОСТАНОВКА ЗАДАЧІ

2.1. Проблематика

Виявлення аномалій у складних середовищах даних створює постійні проблеми, які вимагають негайної уваги як з боку приватних осіб, так і власників бізнесу. Традиційні системи виявлення аномалій стикаються з такими проблемами, як затримка, обмеження масштабованості та сприйнятливість до окремих точок збою, що ставить під загрозу їхню ефективність у забезпеченні цілісності даних і загальної безпеки. Щоб вирішити ці проблеми, ми пропонуємо інтеграцію передових механізмів виявлення аномалій, таких як LSTM, із механізмами уваги. Ці механізми продемонстрували чудову продуктивність у розшифровці складних шаблонів у даних, забезпечуючи підвищену точність і ефективність виявлення аномалій. Завдяки поєднанню LSTM з механізмами уваги наша система визначає пріоритетність важливих даних, сприяючи більш точному розумінню та прийняттю рішень серед різноманітних читань даних. Крім того, наш підхід підкреслює важливість точного налаштування алгоритмів виявлення аномалій відповідно до конкретних потреб середовищ Інтернету речей, забезпечуючи адаптивність і ефективність виявлення аномалій. Крім того, наші дослідження зосереджені на розробці програмного забезпечення для систем управління Інтернетом речей, надаючи користувачам інтуїтивно зрозумілі інтерфейси для моніторингу та керування пристроями Інтернету речей. Шляхом ретельного тестування та оцінювання ми прагнемо виміряти продуктивність нашої системи виявлення аномалій і програмного забезпечення для керування Інтернетом речей, забезпечуючи масштабованість, ефективність і надійність у вирішенні нових проблем складних середовищ даних. Наші зусилля також включають співпрацю з різними галузевими партнерами для впровадження та вдосконалення запропонованих рішень у реальних умовах. Це дозволить не тільки підвищити рівень захисту даних, але й сприяти інноваціям у сфері Інтернету речей.

2.2. Технічне завдання дослідження

Визначити конкретні цілі та масштаби.

Визначити цілі розробки децентралізованого прототипу виявлення аномалій з використанням LSTM з механізмами уваги в розумних середовищах IoT. Вказати очікувані результати та окреслити передбачувані застосування у персональномих та бізнес середовищах, наголошуючи на унікальних проблемах складних даних в розумних системах інтернету речей.

Огляд літератури.

Провести поглиблений огляд літератури, зосередивши увагу на методологіях виявлення аномалій у просунутих IoT системах, периферійних обчисленнях і практичному застосуванні LSTM з механізмами уваги. Узагальнити ключові висновки та визначити прогалини, що стосуються цільового дослідження.

Формулювання методології.

Розробити детальну методологію дослідження, яка охоплює етапи розробки прототипу, інтеграцію LSTM з механізмами уваги та оцінку ефективності алгоритмів. Чітко сформулювати етапи кожного етапу, вирішуючи конкретні проблеми, пов’язані з виявленням аномалій у комлексних системах Інтернету речей.

Збір і попередня обробка набору даних.

Отримати та зробити попередню обробку наборів даних реального світу, включно з наборами даних NASA про турбовентилятори, для створення симуляцій, що відображають розумні середовища IoT. Переконатися, що набори даних охоплюють тонкощі та варіації, присутні в різноманітних сценаріях реального світу.

Вибір алгоритму та структура інтеграції.

Вибрати та обґрунтувати алгоритми машинного навчання, орієнтуючись на точність виявлення аномалій. Сформулювати структуру для інтеграції цих алгоритмів у просунуті системи IoT, наголошуючи на сумісності та ефективності виробничих процесів.

Реалізація та оптимізація моделі.

Реалізувати та оптимізувати модель LSTM з увагою в архітектурі складних IoT систем, наголошуючи на аналізі в реальному часі та відмовостійкості. Оптимізувати модель для масштабованості та ефективності, щоб узгодити її з обмеженнями ресурсів, притаманними складному середовищу IoT.

Тестування та оцінка.

Провести ретельне тестування та оцінку прототипу децентралізованого виявлення аномалій. Виміряти продуктивність за ключовими параметрами, включаючи точність виявлення, затримку, масштабованість і відмовостійкість. Переконатися, що тестування охоплює різні контексти реального світу для перевірки адаптивності.

Уточнення методології дослідження.

Удосконалити методологію дослідження на основі інформації, отриманої на етапах розробки та оцінки. Відкоригувати підхід, щоб підвищити ефективність прототипу у вирішенні проблем, характерних для виявлення комлексних аномалій Інтернету речей.

Виклики та можливості інтеграції.

Дослідити проблеми та можливості, пов’язані з інтеграцією LSTM з механізмами уваги в розумні середовища IoT. Надати практичні ідеї та рекомендації щодо подолання проблем інтеграції та використання можливостей.

Аналіз та інтерпретація результатів.

Проаналізувати та інтерпретувати результати, отримані під час тестування прототипу, підкреслюючи ефективність моделі у виявленні аномалій у реальному часі, масштабованість в різнонаправлених галузях та адаптивну продуктивність у різних контекстах.

Документація та звітність.

Задокументувати весь процес дослідження, включаючи методологію, деталі впровадження та аналіз результатів. Підготувати вичерпний звіт із викладенням ключових висновків, проблем, які було вирішено, розроблених рішень і рекомендацій щодо подальших досліджень виявлення аномалій за допомогою

просунутих алгоритмів, приділяючи увагу розумним середовищам Інтернету речей.

2.3. Інструменти для проведення дослідження

а) Frontend:

1) React: бібліотека JavaScript для створення інтерфейсів користувача. Вона дозволяє створювати повторно використовувані компоненти інтерфейсу користувача, забезпечуючи динамічний і ефективний спосіб розробки інтерактивних веб-додатків. Використовує компонентну архітектуру для модульної розробки, забезпечуючи масштабованість і зручність обслуговування. Використання віртуального DOM покращує продуктивність за рахунок мінімізації перезавантажень сторінок;
2) JavaScript: універсальна мова сценаріїв високого рівня, яка дозволяє створювати динамічний вміст на веб-сторінках. Це основа веб-розробки та забезпечує основні функції для додатків React. JavaScript має вирішальне значення для обробки взаємодії на стороні клієнта, забезпечуючи безперебійну та чутливу роботу користувачів у веб-додатках;
3) Visual Studio Code: легкий, потужний редактор коду з підтримкою різних мов програмування. Він пропонує такі функції, як IntelliSense, підтримка налагодження та розширення для вдосконаленої розробки. Відомий своїм зручним інтерфейсом і розширюваністю, Visual Studio Code підвищує ефективність розробки інтерфейсу. Розширення можна додавати, щоб адаптувати редактор до конкретних потреб проекту;

б) контроль версій:

1) GitHub: веб-платформа для керування версіями за допомогою Git. Вона полегшує спільну розробку програмного забезпечення, пропонуючи такі функції, як запити на отримання, відстеження проблем і керування проектами. GitHub сприяє співпраці між

розробниками, надаючи централізоване сховище для коду. Це підвищує прозорість проекту та забезпечує цілісність контролю версій;

в) IoT:

1) Arduino IDE: платформа для програмування мікроконтролерів Arduino. Це спрощує розробку вбудованих систем і додатків IoT. Arduino IDE є зручним для початківців і підтримує широкий спектр плат Arduino, що робить його придатним для створення прототипів і розробки рішень IoT;
2) Python: універсальна мова програмування високого рівня, відома своєю читабельністю та простотою використання. Вона зазвичай використовується в додатках IoT для обробки даних, зв’язку та створення сценаріїв на стороні сервера. Це полегшує зв’язок між пристроями та може використовуватися для аналізу даних;

г) алгоритми:

1) Python
2) Visual Studio Code
3) WebStorm: інтегроване середовище розробки (IDE), спеціально розроблене для веб-розробки. Воно підтримує JavaScript, HTML і CSS, що робить його ідеальним для реалізації інтерфейсного алгоритму. WebStorm надає такі функції, як інтелектуальне завершення коду та аналіз коду в реальному часі, покращуючи процес розробки алгоритмів у інтерфейсних програмах.

д) зберігання даних:

1) Docker: це платформа для контейнерних програм. Він спрощує розгортання та масштабування додатків, у тому числі тих, що потребують рішень для зберігання даних. Docker забезпечує узгодженість середовищ розробки та виробництва, що робить його зручним інструментом для розгортання програм із різноманітними потребами зберігання даних;

2) Firebase: комплексна платформа від Google, яка надає різні послуги, зокрема бази даних у реальному часі та автентифікацію. Він підходить для веб- і мобільних додатків. Firebase спрощує налаштування та керування базами даних, пропонуючи синхронізацію в реальному часі. Це особливо ефективно для програм з динамічними потребами в даних;
3) CSV: простий формат файлу, який використовується для зберігання табличних даних. Він легкий і широко підтримується, що робить його придатним для зберігання даних у певних програмах. Із CSV легко працювати, і він часто використовується для зберігання наборів даних, особливо в сценаріях, коли достатньо легкого та зручного для читання формату;

е) машинне навчання:
1) Jupyter Notebook: забезпечує інтерактивне обчислювальне середовище, що дозволяє користувачам створювати та обмінюватися документами, що містять живий код, рівняння, візуалізації та описовий текст. Jupyter Notebook є безцінними для розробки інтерактивних моделей, забезпечуючи платформу, на якій алгоритми машинного навчання можна тестувати та вдосконалювати крок за кроком, покращуючи розуміння поведінки моделі;
2) TensorFlow і PyTorch: потужні фреймворки машинного навчання з відкритим кодом. TensorFlow розробляє Google, а PyTorch підтримує Facebook. Обидва фреймворки підтримують реалізацію складних моделей машинного навчання;

ж) візуалізація даних:
1) Tableau: провідний інструмент візуалізації даних, який дозволяє користувачам створювати інтерактивні інформаційні панелі, доступні для спільного використання. Він підтримує широкий спектр джерел даних і відомий простотою використання. Tableau чудово створює візуально привабливі та проникливі візуалізації, що робить його

чудовим вибором для представлення результатів машинного навчання та ідей, отриманих на основі даних;

2) Power BI: служба бізнес-аналітики від Microsoft, яка дозволяє користувачам візуалізувати свої дані та ділитися ними. Він легко інтегрується з іншими інструментами Microsoft. Power BI надає надійні можливості візуалізації даних, пропонуючи зручний інтерфейс і можливість вбудовувати інтерактивні звіти у веб-програми, що робить його придатним для демонстрації результатів машинного навчання;
3) Matplotlib: це бібліотека для створення 2D графіків у Python. Вона надає широкі можливості для візуалізації даних, таких як лінійні графіки, гістограми, розподіли та діаграми розсіювання. Завдяки своїй гнучкості та простоті використання, Matplotlib часто використовується для відображення результатів аналізу даних та моделей машинного навчання;
4) Seaborn: це бібліотека для візуалізації даних у Python, яка базується на Matplotlib. Вона спрощує створення складних візуалізацій, таких як теплові карти, ящики з вусами та віолончельні діаграми, що дозволяє швидко аналізувати та розуміти дані.

3. АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ТА АЛГОРИТМІВ

3.1. Огляд наукової літератури

Дослідження, детально описане в [1], привертає увагу до застосування алгоритму K-Means в аспекті кластеризації Industrial IoT. Приділяючи особливу увагу вирішенню проблеми швидкості зв'язку, дослідження представляє 4-рівневу мережеву топологію, адаптовану для цифрового виробництва, таким чином вирішуючи проблеми на рівні «пристрій-пристрій» (D2D). Крім того, розробка уніфікованої системи збору даних у поєднанні з платформою обробки даних, що використовує структуру Lambda, значно підвищує ефективність обробки виробничих даних. Інтеграція технології промислового Інтернету речей з інтелектуальним виробництвом демонструє значні успіхи в галузі інтелектуальних систем.

Підхід, викладений у [2], працює в архітектурі туманно-хмарних обчислень, щоб протистояти викликам, пов'язаним із обсягом і різноманітністю даних у середовищах IoT. Автори поділяють шар туману на три окремі рівні для організації та зберігання потоків даних IoT. Використовуючи алгоритм просторової кластеризації додатків із шумом на основі щільності (DBSCAN), вони успішно кластеризують різноманітні дані IoT в однорідні кластери з високою щільністю. Цей інноваційний підхід демонструє ефективну організацію даних і прискорює можливості пошуку в динамічних середовищах IoT.

Методологія, запропонована в [3], поєднує алгоритм ізольованого лісу (iForest) з алгоритмом автокодування для виявлення аномалій даних. Завдяки обчисленню балів аномалій для даних про енергію за допомогою алгоритму iForest та ідентифікації аномалій на основі помилки реконструкції за допомогою автокодеру цей метод виявляється ефективним у виявленні аномалій. Навчання автокодувальника з нормальними даними та встановлення порогових значень аномалій через помилки реконструкції сприяє його ефективності у виявленні аномалій.

У [4] представлено ефективне рішення для виявлення аномальної поведінки зв'язку в промислових системах керування. Автори представляють новий

алгоритм виявлення вторгнень на основі One-Class Support Vector Machine (OCSVM). Вони встановлюють модель нормальної комунікаційної поведінки за допомогою OCSVM і оптимізують параметри моделі за допомогою алгоритму оптимізації роя частинок. Цей інноваційний підхід демонструє потенціал для ефективного визначення ненормальної поведінки в промислових системах керування.

Пропозиція в [5] виступає за реалізацію алгоритму LSTM для динамічного розподілу ресурсів у програмах. Програмне рішення моделює розподіл ресурсів за допомогою навченої моделі LSTM, враховуючи евристику використання ресурсів програми. Поєднання LSTM із підходами до динамічної маршрутизації хмарного центру обробки даних виявляється корисним, ефективно реагуючи на зміну моделей трафіку та досягаючи прийнятних угод про рівень обслуговування (SLA).

Оцінка підходу DeepAnT у [6] для різних наборів даних показує його здатність виявляти точкові аномалії та контекстуальні аномалії в даних часових рядів, навіть із періодичними та сезонними характеристиками. Цей підхід підтверджує його ефективність у різних областях, таких як дорожній рух, використання мережі, онлайн-реклама, інтернет-трафік, космічні човники та набори даних про здоров’я.

Пропозиція в [7] виступає за використання алгоритмів LSTM (довгокороткочасної пам’яті), посилених механізмами уваги для виявлення аномалій у промислових процесах. Цей підхід використовує унікальні можливості мереж LSTM, доповнених механізмами уваги, щоб розпізнавати тонкі аномалії в складних послідовних промислових даних. Використовуючи механізми привернення уваги, модель вибірково зосереджується на ключових елементах даних, що дозволяє підвищити точність виявлення аномалій та можливість інтерпретації. Інтеграція LSTM з механізмами уваги представляє багатообіцяючий шлях для виявлення тонких аномалій у різноманітних промислових наборах даних, потенційно пропонуючи краще розуміння складних моделей і залежностей у послідовних потоках даних. Цей підхід є свідченням

постійного дослідження та використання передових архітектур глибокого навчання для покращення виявлення аномалій у промислових умовах.

Підводячи підсумок, у розглянутих статтях пропонуються інноваційні підходи до виявлення аномалій у різних областях, включаючи промисловий Інтернет речей, розподіл ресурсів і аналіз часових рядів. Кожна робота стосується конкретних завдань і демонструє багатообіцяючі результати. Однак для подальшого розвитку галузі автори відстоюють концепцію децентралізованої системи виявлення аномалій з архітектурою периферійних хмарних обчислень. Цей прототип системи, що використовує потужність периферійних обчислень і хмарних ресурсів, має потенціал для революції у виявленні аномалій у промислових застосуваннях, пропонуючи аналіз у реальному часі, масштабованість і модульне розширення. Ця запропонована система є значним кроком вперед у розвитку методологій виявлення аномалій у різноманітних промислових умовах.

3.2. Аналіз існуючих рішень

3.2.1. Перелік існуючих рішень

Siemens MindSphere – це промислова платформа IoT, яка надає різні послуги, зокрема аналітику та виявлення аномалій. Він підтримує розробку користувальницьких програм для конкретних промислових випадків використання.

Splunk – це платформа для пошуку, моніторингу й аналізу машинно-генерованих даних, у тому числі даних IoT. Він пропонує потужні можливості візуалізації та інтегрується з моделями машинного навчання для виявлення аномалій.

ThingWorx – це промислова платформа IoT, яка включає інструменти візуалізації для створення інтерактивних інформаційних панелей. Він підтримує моніторинг у реальному часі та може інтегруватися з алгоритмами виявлення аномалій.

McAfee пропонує рішення безпеки IoT, які включають виявлення загроз, керування пристроями та безпечний зв'язок. Він вирішує проблеми безпеки в промислових середовищах Інтернету речей. McAfee також надає підтримку інтеграції з різними промисловими стандартами, що забезпечує сумісність з широким спектром IoT-пристроїв. Їхні рішення використовують передові методи шифрування для захисту даних під час передачі та зберігання. McAfee пропонує можливості віддаленого керування, дозволяючи адміністраторам швидко реагувати на потенційні загрози та усувати їх.

Amazon Web Services (AWS) IoT Analytics надає комплексний пакет для обробки даних і аналітики IoT. Він містить функції виявлення аномалій, що дозволяє користувачам створювати власні моделі виявлення аномалій для даних датчиків. Інтеграція з іншими службами AWS сприяє безперебійному контролю та управлінню.

Платформа IBM Watson IoT інтегрує дані пристроїв IoT із можливостями Watson AI. Він пропонує функції виявлення аномалій за допомогою алгоритмів машинного навчання. Платформа надає механізми контролю для керування пристроями IoT і реагування на аномалії.

Losant – корпоративна платформа IoT, яка надає інструменти для створення додатків у реальному часі. Він містить функції для виявлення аномалій у даних датчиків і пропонує автоматизацію робочого процесу для керування пристроями на основі виявлених аномалій.

Ubidots – це хмарна платформа, орієнтована на аналітику та візуалізацію даних Інтернету речей. Він містить функції виявлення аномалій для виявлення незвичайних шаблонів у даних датчиків. Ubidots дозволяє користувачам створювати контрольні програми на основі виявлених аномалій.

Bosch IoT Suite надає набір хмарних служб для програм IoT. Він містить функції для виявлення аномалій у даних датчиків і пропонує функції керування пристроями для керування пристроями IoT

3.3. Порівняння існуючих рішень

Порівняємо ці платформи IoT на основі дев’яти характеристик, пов’язаних з їх продуктивністю, масштабованістю та безпекою:

а) масштабованість:

1) Siemens MindSphere: забезпечує масштабованість для випадків промислового використання з можливістю обробки великої кількості підключених пристроїв і потоків даних;
2) Splunk: добре масштабується для обробки машинно-генерованих даних, підходить для різних джерел даних IoT;
3) ThingWorx: Розроблено для масштабування для промислових додатків, підтримує збільшення кількості підключених пристроїв і обсягу даних;
4) McAfee IoT Security: Зосереджено на безпеці; на масштабованість можуть впливати реалізовані функції безпеки;
5) AWS IoT Analytics: відомий високою масштабованістю, придатністю до широкомасштабного розгортання IoT і обробкою значних обсягів даних;
6) IBM Watson IoT: пропонує масштабованість, особливо при інтеграції з іншими службами IBM Cloud;
7) Losant: забезпечує масштабованість для створення додатків у реальному часі, придатних для низки проектів IoT;
8) Ubidots: Пропонує масштабованість для аналізу та візуалізації даних IoT, підходить для різних програм;
9) Bosch IoT Suite: розроблений для масштабованості, що включає різноманітні додатки IoT;

б) продуктивність:

1) Siemens MindSphere: забезпечує надійну продуктивність для промислової аналітики та виявлення аномалій;
2) Splunk: відомий своєю високопродуктивною аналітикою даних, придатною для моніторингу та аналізу в реальному часі;

3) ThingWorx: забезпечує ефективний моніторинг у реальному часі та інтеграцію з алгоритмами виявлення аномалій;
4) McAfee IoT Security: зосереджено на безпеці, особливо у виявленні загроз і безпечних комунікаціях;
5) AWS IoT Analytics: пропонує високопродуктивну аналітику та обробку даних у реальному часі;
6) IBM Watson IoT: Ефективна продуктивність інтеграції даних пристрою IoT із можливостями AI;
7) Losant: призначений для створення додатків у реальному часі з ефективною продуктивністю;
8) Ubidots: забезпечує хорошу продуктивність для аналізу та візуалізації даних IoT;
9) Bosch IoT Suite: забезпечує ефективну роботу програм IoT;

в) Максимальне навантаження:
1) Siemens MindSphere: справляється зі значними навантаженнями, типовими для промислових умов;
2) Splunk: здатний обробляти великі навантаження машинно-генерованих даних;
3) ThingWorx: Витримує значні навантаження, підходить для промислового розгортання IoT;
4) McAfee IoT Security: може обробляти навантаження, пов’язані з виявленням загроз і безпечним зв’язком;
5) AWS IoT Analytics: може обробляти значне навантаження даних у сценаріях аналітики IoT;
6) IBM Watson IoT: обробляє різні навантаження в інтеграції даних пристроїв IoT за допомогою можливостей ШІ;
7) Losant: може обробляти навантаження, пов’язані зі створенням додатків у реальному часі;
8) Ubidots: обробляє навантаження для аналізу та візуалізації даних IoT;

9) Bosch IoT Suite: може обробляти різноманітні навантаження в програмах IoT;

г) функції безпеки:

1) Siemens MindSphere: пропонує функції безпеки, адаптовані для промислових середовищ Інтернету речей;
2) Splunk: забезпечує функції безпеки для керування й аналізу даних, створених машиною;
3) ThingWorx: містить функції безпеки, придатні для промислового використання Інтернету речей;
4) McAfee IoT Security: спеціалізується на рішеннях безпеки IoT, пропонуючи комплексне виявлення загроз і керування пристроями;
5) AWS IoT Analytics: реалізує надійні функції безпеки для обробки та аналітики даних IoT;
6) IBM Watson IoT: містить функції безпеки для керування та реагування на аномалії в даних пристрою IoT;
7) Losant: забезпечує функції безпеки для створення додатків у реальному часі та виявлення аномалій;
8) Ubidots: пропонує функції безпеки для аналізу та візуалізації даних IoT;
9) Bosch IoT Suite: містить функції безпеки для додатків IoT;

д) налаштування та інтеграція:

1) Siemens MindSphere: підтримує розробку користувальницьких програм для конкретних випадків промислового використання;
2) Splunk: налаштовується та добре інтегрується з різними джерелами даних і моделями машинного навчання;
3) ThingWorx: надає інструменти для створення спеціальних програм і інтегрується з алгоритмами виявлення аномалій;
4) McAfee IoT Security: Спецізується на захисті Інтернету речей, з можливостями інтеграції для виявлення загроз і керування пристроями;

5) AWS IoT Analytics: Бездоганно інтегрується з іншими службами AWS, надаючи можливості налаштування;
6) IBM Watson IoT: Інтегрується з іншими службами IBM Cloud і дозволяє налаштовувати програми IoT;
7) Losant: Призначений для створення спеціальних програм у режимі реального часу та інтегрований із виявленням аномалій;
8) Ubidots: Пропонує параметри налаштування для створення додатків керування на основі виявлених аномалій;
9) Bosch IoT Suite: Надає параметри налаштування та інтегрується з різними додатками IoT;

е) простота використання:

1) Siemens MindSphere: призначений для промислових користувачів, пропонує інтерфейс, придатний для промислових середовищ;
2) Splunk: відомий зручним інтерфейсом, але може потребувати певних знань для розширеного використання;
3) ThingWorx: розроблений зі зручним інтерфейсом, придатним для промислового застосування;
4) McAfee IoT Security: призначено для професіоналів у сфері безпеки, зосереджено на простоті використання для виявлення загроз;
5) AWS IoT Analytics: пропонує зручне середовище, особливо для користувачів, знайомих із службами AWS;
6) IBM Watson IoT: зручний інтерфейс із інструментами для керування та реагування на аномалії пристроїв IoT;
7) Losant: розроблено з урахуванням простоти використання, підходить для створення додатків у реальному часі;
8) Ubidots: зручна платформа, особливо для створення програм аналізу та візуалізації даних IoT;
9) Bosch IoT Suite: забезпечує зручний інтерфейс для керування програмами IoT;

ж) економічна ефективність:

1) Siemens MindSphere: ціна може змінюватися в залежності від використання; індивідуальні варіанти ціноутворення для випадків промислового використання;
2) Splunk: Ціни можуть змінюватися; може потребувати значних інвестицій для обширної аналітики даних;
3) ThingWorx: ціни можуть відрізнятися; розроблений, щоб запропонувати економічно ефективні рішення для промислових додатків IoT;
4) Безпека McAfee IoT: вартість залежить від вибраних рішень безпеки; фокусується на цінності інвестицій у безпеку;
5) AWS IoT Analytics: економічно ефективний для користувачів в екосистемі AWS за допомогою моделі оплати за використання;
6) IBM Watson IoT: вартість залежить від використання; Застосовуються моделі ціноутворення IBM Cloud;
7) Losant: пропонує економічно ефективні рішення для створення додатків у реальному часі;
8) Ubidots: надає економічно ефективні варіанти, особливо для малих і середніх проектів IoT;
9) Bosch IoT Suite: ціни можуть відрізнятися; розроблено для економічно ефективного керування програмами IoT;

з) гнучкість у виявленні аномалій:
1) Siemens MindSphere: пропонує гнучкість у впровадженні виявлення аномалій, адаптованих для промислових середовищ;
2) Splunk: дуже гнучкий, підтримує інтеграцію моделей машинного навчання для виявлення аномалій;
3) ThingWorx: забезпечує гнучкість інтеграції з алгоритмами виявлення аномалій, придатними для промислового IoT;
4) McAfee IoT Security: фокусується на гнучкості механізмів виявлення загроз і реагування;

5) AWS IoT Analytics: гнучкість у створенні спеціальних моделей виявлення аномалій в екосистемі AWS;
6) IBM Watson IoT: гнучкість реалізації виявлення аномалій за допомогою алгоритмів машинного навчання;
7) Losant: пропонує гнучкість у створенні додатків у реальному часі з вбудованим виявленням аномалій;
8) Ubidots: гнучка платформа, яка дозволяє користувачам впроваджувати спеціальне виявлення аномалій на основі даних IoT;
9) Bosch IoT Suite: забезпечує гнучкість у реалізації виявлення аномалій у різноманітних програмах IoT;

и) інтеграція з можливостями AI:
1) Siemens MindSphere: інтегрується з можливостями штучного інтелекту Siemens для розширеної аналітики та виявлення аномалій;
2) Splunk: добре інтегрується з моделями машинного навчання для розширеної аналітики та виявлення аномалій;
3) ThingWorx: надає можливості інтеграції для розширеної аналітики та машинного навчання, підтримуючи можливості ШІ;
4) McAfee IoT Security: інтегрується з можливостями штучного інтелекту McAfee для покращеного виявлення загроз;
5) AWS IoT Analytics: інтеграція з різними сервісами AWS AI та машинного навчання для розширеної аналітики;
6) IBM Watson IoT: плавно інтегрується з можливостями IBM Watson AI для розширеної аналітики;
7) Losant: надає можливості інтеграції для створення додатків у реальному часі з можливостями AI;
8) Ubidots: інтеграція з можливостями AI для розширеної аналітики та виявлення аномалій;
9) Bosch IoT Suite: інтегрується з можливостями ШІ Bosch для покращеної аналітики та виявлення аномалій;

На основі отриманих даних побудуємо порівняльну таблицю аналогів(таблиця Додаток Б):

Таким чином, кожна платформа має свої сильні сторони, і вибір залежить від конкретних вимог персонального або промислового проекту IoT. Siemens MindSphere, Splunk, ThingWorx і AWS IoT Analytics виділяються своїми можливостями масштабованості, продуктивності та гнучкості у виявленні аномалій. На основі отриманих даних побудуємо порівняльну таблицю аналогів, щоб краще розуміти унікальні можливості та обмеження кожної платформи. Siemens MindSphere, Splunk, ThingWorx і AWS IoT Analytics виділяються своїми можливостями масштабованості, продуктивності та гнучкості у виявленні аномалій. Siemens MindSphere пропонує широкий спектр інтегрованих інструментів для великих промислових проектів. Splunk відомий своєю потужною платформою аналізу даних, а ThingWorx відрізняється простотою використання та розширенням. AWS IoT Analytics використовує потужність хмарних обчислень Amazon Web Services для обробки великих обсягів даних IoT. Кожна з цих платформ має свої унікальні переваги та обмеження, що може вплинути на вибір найбільш підходящої для конкретного проекту. Враховуючи широкий вибір доступних рішень для виявлення аномалій в системах IoT, важливо ретельно проаналізувати можливості та функціонал кожної платформи перед прийняттям рішення. Додаткові фактори, такі як вартість впровадження, підтримка та масштабованість, також можуть вплинути на кінцевий вибір платформи для виявлення аномалій в системах IoT.

4. ЗБІР І АНАЛІЗ ДАНИХ

4.1. Збір даних

4.1.1 Тренування моделі

Для тренування моделі було обрано набір даних NASA Turbofan Engine Degradation Simulation.[8]

Набір даних NASA Turbofan Engine Degradation Simulation – це широко використовуваний набір даних у сфері прогнозованого технічного обслуговування та виявлення аномалій. Цей набір даних спеціально розроблений для моделювання деградації турбовентиляторних двигунів з часом. Він містить кілька робочих налаштувань, таких як різні умови польоту та режими несправностей, імітуючи реальну поведінку двигунів, коли вони старіють або виникають несправності.

Набір даних моделювання деградації турбовентиляторного двигуна NASA представляє чудову можливість для застосування моделей LSTM із механізмами уваги через його складну природу та позначені аномалії. Цей набір даних імітує деградацію турбовентиляторних двигунів у різних робочих умовах і режимах несправностей, пропонуючи складний і різноманітний діапазон даних. Моделі LSTM, відомі своєю майстерністю в розумінні послідовних даних, можуть використовувати цю складність для захоплення складних шаблонів і залежностей у часових послідовностях, присутніх у наборі даних. Параметри набору даних включають параметри, доступні як дані датчика (загальна температура, тиск, фізична швидкість вентилятора та швидкість серцевини, потік палива, скоригована швидкість вентилятора та швидкість серцевини тощо) і загальна температура на виході. Специфікація набору даних наведено нижче:

- опис набору даних. Набір даних: FD002; Траєкторії тренувань: 260; Тестові траєкторії: 259; Умови: ШІСТЬ; Режими несправності: ОДИН (деградація компресора високого тиску (HPC)) [10];
- вибір архітектури моделі. Вибирається модель LSTM з двома шарами. Перший шар LSTM має 64 одиниці, а другий – 32 одиниці. Архітектура

цієї моделі спрямована на фіксацію послідовних шаблонів у даних датчика. Згенерований набір даних ділиться на набори для навчання (260 траєкторій) і тестування (259 траєкторій). Мета моделі полягає в тому, щоб вивчити закономірності в робочих даних і даних датчиків, щоб передбачити виникнення погіршення HPC у цих блоках. На рисунку 4.1 показано налаштування моделі в програмному середовищі;

– підготовка. Імпортовані дані датчиків збираються за допомогою випадкового нормального розподілу. Дані про термін служби двигуна, що залишився, збираються для кожного циклу;
– процес навчання: модель навчається протягом 20 епох, повторюючи весь набір даних 20 разів. Кожна навчальна ітерація обробляє 32 зразки одночасно. Середня квадратична похибка використовується як функція втрат, а середня абсолютна похибка використовується як метрика для вимірювання точності. На малюнку 4.2 можна побачити процес навчання;
– поепохальний аналіз. Протягом кожної епохи модель вивчає дані, і втрати поступово мінімізуються. Одночасно покращується показник середньої абсолютної похибки, що вказує на покращену здатність моделі оцінювати залишковий ресурс двигуна.

```
file_path = '/Users/ihor/Downloads/CMaps/train_FD002.txt'

dataset = pd.read_csv(file_path, sep=' ', header=None)

X = dataset.iloc[:, 2:28].values
y = dataset.iloc[:, -1].values

X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42)

X_train = np.reshape(X_train, (X_train.shape[0], 1, X_train.shape[1]))
X_val = np.reshape(X_val, (X_val.shape[0], 1, X_val.shape[1]))

inputs = Input(shape=(X_train.shape[1], X_train.shape[2]))
lstm = LSTM(units=64, return_sequences=True)(inputs)
attention = Dense(1, activation='tanh')(lstm)
attention = Activation('softmax')(attention)
attention = Dot(axes=1)([attention, lstm])
output = Dense(units=1)(attention)
model = Model(inputs=inputs, outputs=output)

model.compile(optimizer='adam', loss='mean_squared_error', metrics=['mae'])

history = model.fit(X_train, y_train, epochs=20, batch_size=32, validation_data=(X_val, y_val), verbose=1)
```

Рисунок 4.1 – Налаштування LSTM з моделлю уваги з набором даних деградації турбовентилятора train_FD002

Також наведено процес навчання моделі (див.рис.4.2):

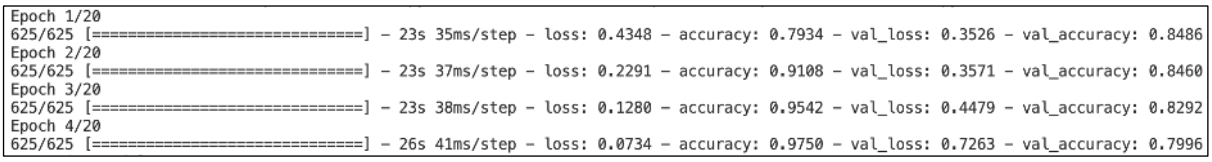

Рисунок 4.2 - Навчання LSTM з моделлю уваги

Ці кроки ілюструють процес навчання моделі на заданому наборі даних.

4.2. Проектування моделі на фізичному пристрої iOT

У контексті розробки комплексної системи Інтернету речей (IoT) із механізмами виявлення аномалій[9] важливо розуміти основну архітектуру та робочий процес такої системи. При роботі з пристроями IoT дані генеруються різними датчиками і передаються в центральний сервіс через інтернет-протоколи. Цей потік інформації формує основу для моніторингу в реальному часі, аналізу даних і виявлення аномалій у складному середовищі.

Система IoT, передбачена для цього проекту, спрямована на обслуговування різноманітних пристроїв IoT, дозволяючи користувачам легко підключати різні конфігурації датчиків і пристроїв. Центральна служба відіграє ключову роль у агрегуванні та обробці вхідних потоків даних, реалізуючи надійний інтерфейс для вибору користувачами конкретних параметрів, які використовуватимуться для подальшого аналізу даних.[10]

Датчики, вбудовані в середовище, постійно збирають дані, пов’язані з різними параметрами, такими як температура, тиск, вологість та інші відповідні показники. Ці датчики стратегічно розміщені в ландшафті, забезпечуючи повне покриття. Потім зібрані дані передаються в центральний сервіс за допомогою стандартних Інтернет-протоколів, утворюючи безпечний і ефективний канал зв’язку між пристроями IoT і центральним процесором.

Отримавши дані датчиків, центральна служба в реальному часі займається обробкою та виявленням аномалій. Це передбачає використання розширених алгоритмів, потенційно включаючи мережі довгострокової короткочасної пам’яті

(LSTM) із механізмами уваги. Ці алгоритми ретельно перевіряють вхідні дані, виявляючи шаблони та аномалії, які можуть означати відхилення від нормальної робочої поведінки. Обрані механізми виявлення аномалій є ключовими для розпізнавання порушень, які можуть вказувати на потенційні несправності або проблеми в промислових процесах.

4.3. Попередня обробка даних

Схема роботи сервісу універсальної системи визначення та управління аномаліями для iot пристроїв:

- передаємо по інтернет протоколу всі дані з iot на сервіс;
- відображаємо вхідні дані, дозволяємо вибрати необхідні параметри;
- формуємо таблицю даних із вибраних користувачем параметрів;
- вибираємо алгоритм виявлення аномалій (LSTM і менш ефективні), також записуючи вибір у параметри запуску;
- запускаємо виявлення аномалій;
- генеруємо графіки ефективності та дріфту iot пристрою;
- вибираємо інструмент оптимізації, і відправляємо його через інтернет протоколу назад на iot пристрій;
- отримуємо оптимізовані дані з iot, та повторюємо аналіз ефективності даних, зі збереженими параметрами та алгоритмом виявлення аномалій у бічному меню сервісу. Отримуємо зміну ефективності пристрою.

4.4. Інструменти аналізу комплексних даних

Ось кілька відомих алгоритмів виявлення аномалій, які підходять для аналізу даних в нашій системі:

- Isolation Forest – це метод ансамблю, який ізолює аномалії шляхом рекурсивного поділу даних. Він особливо ефективний у роботі з великомірними даними та може ефективно обробляти складні шаблони;
- One-Class SVM – це алгоритм машинного навчання, який навчається лише на «звичайному» класі, що робить його придатним для виявлення

аномалій. Він добре працює зі складними даними та може фіксувати нелінійні шаблони;

- автокодери – це архітектури нейронних мереж, призначені для зменшення розмірності. Частина кодера вчиться представляти нормальні шаблони, а аномалії можна виявити шляхом порівняння помилки реконструкції. Вони здатні вловлювати складні закономірності в даних;
- коефіцієнт локального викиду (LOF)– це алгоритм на основі щільності, який визначає аномалії шляхом порівняння локальної щільності екземплярів. Він ефективний у виявленні аномалій в областях різної щільності, що робить його придатним для складних шаблонів даних;
- варіаційні автокодери (VAE) є розширенням традиційних автокодерів з імовірнісним підходом. Він ефективний для фіксації невизначеності в даних, що може бути цінним для виявлення аномалій у складних і динамічних середовищах;
- кластерне виявлення аномалій: Такі алгоритми, як DBSCAN (просторова кластеризація програм на основі щільності з шумом) або k-середні, можна використовувати для виявлення аномалій шляхом розгляду екземплярів, які не належать до жодного кластера або знаходяться в невеликих, розріджених кластерах;
- приховані моделі Маркова (HMM))– це ймовірнісна модель, яку можна використовувати для виявлення аномалій часових рядів. Він може фіксувати часові залежності в даних, що робить його придатним для даних датчиків Інтернету речей із шаблонами на основі часу;
- мережі довготривалої короткочасної пам’яті (LSTM): LSTM, тип рекурентної нейронної мережі (RNN), добре підходить для даних послідовності. Він може ефективно фіксувати складні часові залежності та корисний для виявлення аномалій у даних датчиків часових рядів.

Впроваджуючи виявлення аномалій для систем IoT, важливо враховувати конкретні характеристики даних[11], такі як розмірність, тимчасові залежності та природу аномалій. Для досягнення оптимальних результатів часто необхідно

експериментувати з поєднанням цих алгоритмів і параметрів точного налаштування на основі ваших характеристик даних. При впровадженні виявлення аномалій для систем IoT важливо враховувати конкретні характеристики даних, такі як розмірність, тимчасові залежності та природу аномалій. Для досягнення оптимальних результатів часто необхідно експериментувати з поєднанням цих алгоритмів і параметрів точного налаштування на основі ваших характеристик даних.

5. ПІДГОТОВКА ДО ПРОВЕДЕННЯ ЕКСПЕРИМЕНТУ

5.1. Вимоги до програмної системи

5.1.1. Вимоги до функціональності серверу

Функціональність серверу повинна задовольняти наступним вимогам:

- сервер повинен обробляти запити з iOT
- сервер повинен заносити записи до бази даних;
- сервер повинен підтримувати авторизацію через протокол авторизації, для обмеження доступу до даних.
- сервер повинен забезпечувати авторизацію доступу до даних для запобігання несанкціонованого доступу.

5.1.2. Вимоги до інтерфейсу

Інтерфейс програмної системи повинен задовольняти наступним вимогам:

- кінцевий продукт повинен бути представлений у виді програмної системи з вихідним кодом backend та frontend;
- кінцева програмна система має бути представлена у вигляді .cs файлів(сервер), .React.

5.1.3. Операційні вимоги

Для налаштування та роботи з базою даних, має бути встановлений сервер MS SQL.

5.1.4. Вимоги до ресурсів

Для експлуатації серверної частини програмної системи, достатньо наступної комплектації: Intel Core i3 4 покоління, RAM 1000 MB, 12 Гб дискового простору. Для експлуатації серверної частини програмної системи достатньо наступної комплектації: процесор Intel Core i3 4-го покоління, оперативна пам'ять об'ємом 1000 MB, і 12 GB дискового простору. Така конфігурація забезпечує достатню обчислювальну потужність для обробки стандартних серверних задач.

Оперативна пам'ять у 1000 MB дозволяє ефективно працювати з базовими серверними додатками. Наявність 12 GB дискового простору гарантує достатній обсяг для зберігання даних і системних файлів.

5.1.5. Вимоги до документації

Розроблювана програмна система повинна супроводжуватися пояснювальної запискою, яка відповідає до Державного стандарту України ДСТУ 3008:2015.

Звіти у сфері науки і техніки. Структура та правила оформлення. Див. Додаток А.

5.1.6. Вимоги до середовищ виконання і платформ

Експлуатація серверної частини можлива з комплектацією ЕОМ, не нижче наступної: Intel Core i3 8 покоління (або еквівалентний процесор AMD) RAM 2048 MB;

5.2. Архітектура та проектування

5.2.1. UML-проектування

- користувач. Атрибути: ідентифікатор користувача (первинний ключ), ім’я користувача, пароль, роль тощо;
- параметри. Атрибути: ParameterID (первинний ключ), UserID (зовнішній ключ), Algorithm, SelectedColumns, Timestamp тощо;
- графіки. Атрибути: GraphID (первинний ключ), UserID (зовнішній ключ), GraphType, дані, мітка часу тощо;
- пристрій IoT. Атрибути: DeviceID (первинний ключ), DeviceName, DeviceType, OptimizationTool, Efficiency, Timestamp тощо;
- виявлення аномалії. Атрибути: DetectionId, ParameterId (зовнішній ключ), Algorithm, DetectionResults, TimeStamp.

Тепер визначимо зв’язки:

- користувач може мати кілька параметрів, графіків і пристроїв IoT;

– кожен параметр пов'язаний з одним користувачем;
– кожен графік пов'язаний з одним користувачем;
– кожен пристрій IoT пов’язано з одним користувачем.

На рисунку 5.1 наведено UML діаграму системи:

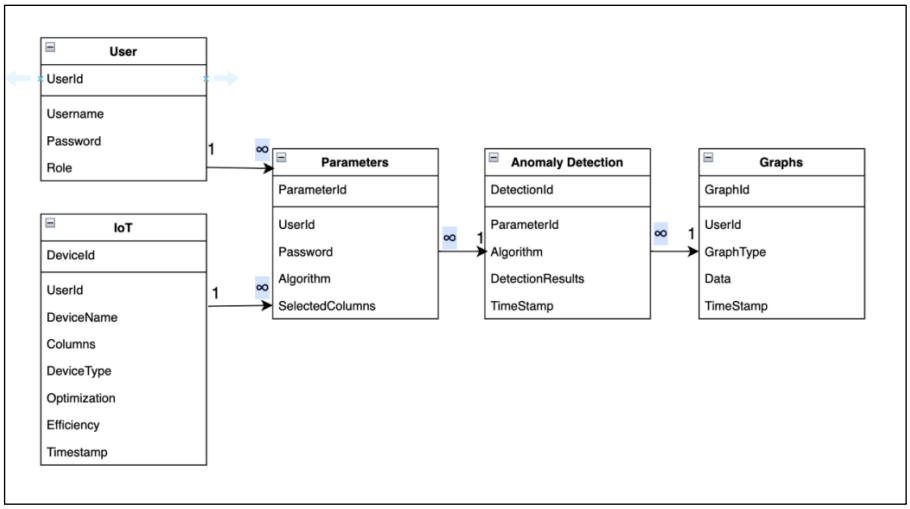

Рисунок 5.1 - UML діаграма системи Scan-Technic

З наведеної діаграми зв’язків в системі можемо побачити як відносяться між собою процеси в системі. Діаграма фіксує зв’язки між користувачами, параметрами, виявленням аномалій, графіками та пристроями Інтернету речей, що ілюструє потік даних і процеси. Діаграма UML покращує розуміння, демонструючи, як користувачі взаємодіють із системою[12], налаштовують параметри аналізу та отримують результати виявлення аномалій. Вона пояснює роль центрального сервера додатків, сервера бази даних і пристроїв Інтернету речей в обробці та оптимізації даних. Крім того, діаграма допомагає зобразити архітектуру розгортання, ілюструючи, як інтерфейс клієнта взаємодіє з серверами та пристроями.

5.2.2. Проектування архітектури програмного забезпечення

а) клієнт (інтерфейс користувача):
   1) представляє інтерфейс користувача, за допомогою якого користувачі взаємодіють із системою;

2) зв'язується з сервером додатків для надсилання запитів і отримання відповідей;

б) сервер додатків:

1) керує бізнес-логікою, включаючи обробку даних, виявлення аномалій та оптимізацію;
2) зв'язується з сервером бази даних для зберігання та отримання даних;
3) зв'язується з пристроями IoT для передачі даних і оптимізації;

в) сервер бази даних:

1) зберігає дані користувача, параметри, графіки та інформацію про пристрій IoT;
2) за потреби надає дані на сервер додатків;

г) пристрої IoT:

1) представляє фізичні пристрої в розумному середовищі;
2) зв'язується з сервером додатків для передачі даних і оптимізації.

На рисунку 5.2 наведено діаграму розгортання системи:

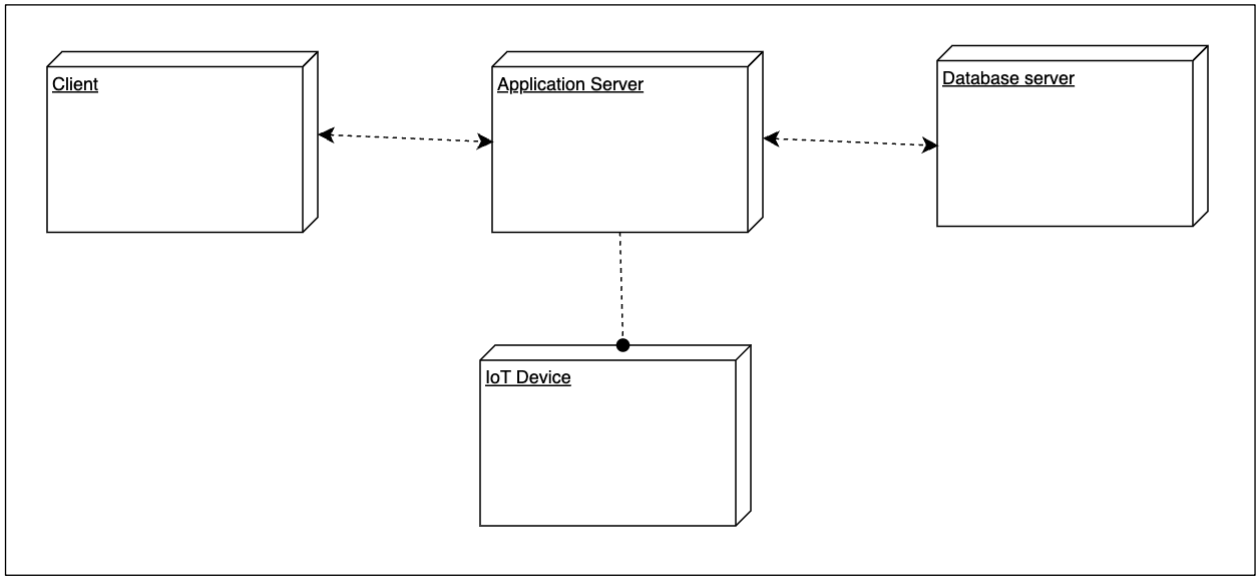

Рисунок 5.2 – Діаграма розгортання системи Scan-Technic

На цій діаграмі показано, як користувач взаємодіє із системою через клієнтський інтерфейс і як сервер додатків взаємодіє з сервером бази даних і пристроями IoT для виконання запитів користувачів і керування загальною функціональністю системи.

5.2.3. Проектування структури зберігання даних

Таблиця Users.

Призначення: таблиця «Користувач» служить основою, що містить важливу інформацію про користувачів системи. Це забезпечує керування користувачами, автентифікацію та контроль доступу на основі ролей, закладаючи основу для персоналізованого досвіду.

Покращення: централізоване керування користувачами покращує безпеку[13], надаючи різні ролі для адміністраторів і звичайних користувачів. Він також встановлює відповідальність за налаштування параметрів і результати виявлення аномалій.

Параметри таблиці:

- UserID (первинний ключ)
- Username
- Password
- Role (наприклад, адміністратор, користувач)

Таблиця Parameters.

Призначення: параметри критичні для виявлення аномалій та аналізу даних. Зберігання параметрів, визначених користувачем, дозволяє налаштовувати[14], а позначка часу допомагає відстежувати зміни з часом.

Покращення: шляхом пов’язування параметрів із конкретними користувачами система сприяє персоналізації. Мітки часу дозволяють відстежувати конфігурації параметрів, допомагаючи в історичному аналізі та потенційній оптимізації

Параметри таблиці:

- ParameterID (первинний ключ)
- UserID (таблиця користувача, яка посилається на зовнішній ключ)
- Algorithm (Тип алгоритму виявлення аномалії)
- SelectedCols (Вибрані стовпці для аналізу)
- Timestamp (Мітка часу конфігурації параметра)

Таблиця Anomaly Detection.

Призначення: результати виявлення аномалій є ключовими для виявлення порушень. Ця таблиця посилається на параметри, забезпечуючи контекст для конкретного процесу виявлення аномалії.

Покращення: Зберігання результатів із посиланнями на параметри полегшує відстеження, дозволяючи користувачам співвідносити аномалії з визначеними конфігураціями. Це покращує прозорість і можливості усунення несправностей.

Параметри таблиці:

- DetectionID (первинний ключ)
- ParameterID (таблиця параметрів, що посилається на зовнішній ключ)
- Algorithm (використовується алгоритм виявлення аномалій)
- DetectionResults (Результати виявлення аномалії)
- Timestamp (Мітка часу виявлення аномалії)

Таблиця Graphs.

Призначення: Графіки представляють візуалізацію проаналізованих даних. Зберігання призначених для користувача графіків дозволяє отримувати персоналізовану інформацію та полегшує відстеження тенденцій даних у часі.

Покращення: Створення графіка з часовими мітками забезпечує часовий контекст, допомагаючи користувачам зрозуміти еволюцію даних. Спеціальні для користувача графіки покращують налаштування та релевантність візуалізованих даних.

Параметри таблиці:

- GraphID (первинний ключ)
- UserID (таблиця користувача, яка посилається на зовнішній ключ)
- GraphType (Тип графіка або візуалізації)
- Data (дані графіка або посилання на дані)
- Timestamp (Мітка часу створення графіка)

Таблиця IoT Device.

Призначення: пристрої IoT надають необроблені дані для аналізу. Зберігання інформації про пристрій, включаючи деталі оптимізації та показники ефективності, підтримує моніторинг продуктивності.

Покращення: зв’язок із користувачами гарантує правильне приписування даних пристрою. Дані часових позначок з пристроїв підтримують історичний аналіз, а деталі оптимізації сприяють цілеспрямованим покращенням.

Параметри таблиці:

- DeviceID (первинний ключ)
- UserID (таблиця користувача, яка посилається на зовнішній ключ)
- DeviceName (Ім'я або ідентифікатор пристрою IoT)
- DeviceType (Тип або категорія пристрою IoT)
- Optimization (подробиці про інструмент або процес оптимізації)
- Efficiency (Показники ефективності пристрою)
- Timestamp (часова позначка даних з пристрою IoT)

Розроблена структура зберігання даних сприяє орієнтованому на користувача, персоналізованому та відстежуваному підходу до керування даними в системі Industrial IoT[15]. Завдяки інкапсуляції параметрів користувача, результатів виявлення аномалій, візуалізацій і деталей пристроїв Інтернету речей система отримує ефективність, масштабованість і здатність адаптуватися до мінливих вимог. Зосередженість структури на взаємозв’язках і часових мітках покращує історичне відстеження, дозволяючи користувачам приймати обґрунтовані рішення, оптимізувати процеси та забезпечувати безпеку та надійність усієї системи. Розроблена структура зберігання даних забезпечує персоналізований та відстежуваний підхід до керування даними в системі Industrial IoT, що сприяє кращому користувацькому досвіду. Інкапсуляція параметрів користувача, результатів виявлення аномалій, візуалізацій і деталей пристроїв підвищує ефективність, масштабованість і адаптивність системи до мінливих вимог.

5.2.4. Створення схеми iOT пристрою

Створення системи Інтернету речей, що імітує розумний пристрій має важливе значення в сфері комплексного аналізу даних і виявлення аномалій[16]. А саме з використанням таких компонентів, як Arduino Leonardo, датчик DHT11, реле, вентилятор і джерело живлення, По-перше, в умовах реального світу існує постійна потреба контролювати різні параметри, такі як температура, вологість та інші фактори навколишнього середовища, щоб забезпечити оптимальні умови роботи. Завдяки інтеграції пристроїв Інтернету речей, подібних до описаного, галузі можуть збирати дані в реальному часі з різних датчиків, дозволяючи здійснювати комплексний моніторинг свого обладнання та процесів. Склад компонентів системи наведено нижче:

- Arduino Leonardo;
- сенсор dht11;
- 1-канальне реле на 5V;
- фен на 12A;
- додатковий роз’єм живлення на 12V.

Крім того, виявлення аномалій відіграє вирішальну роль у виявленні відхилень від нормальних умов експлуатації. У контексті описаної системи IoT алгоритми виявлення аномалій можуть аналізувати дані, зібрані датчиком DHT11, щоб виявити аномальні коливання температури, які можуть вказувати на потенційні проблеми, такі як перегрів. Шляхом швидкого виявлення аномалій система може ініціювати відповідні дії, такі як регулювання швидкості вентилятора за допомогою реле, щоб знизити ризик виходу з ладу або пошкодження обладнання.

Також, такі системи IoT забезпечують проактивне управління та контроль розумних середовищ. Замість того, щоб покладатися виключно на ручний моніторинг і втручання, автоматизовані системи можуть постійно аналізувати дані, виявляти аномалії та вживати коригувальних заходів у режимі реального часу. Це не тільки підвищує ефективність роботи, але й мінімізує час простою та

витрати на технічне обслуговування, вирішуючи потенційні проблеми до їх загострення. Схематичний рисунок такої системи наведено нижче (рисунок 5.3):

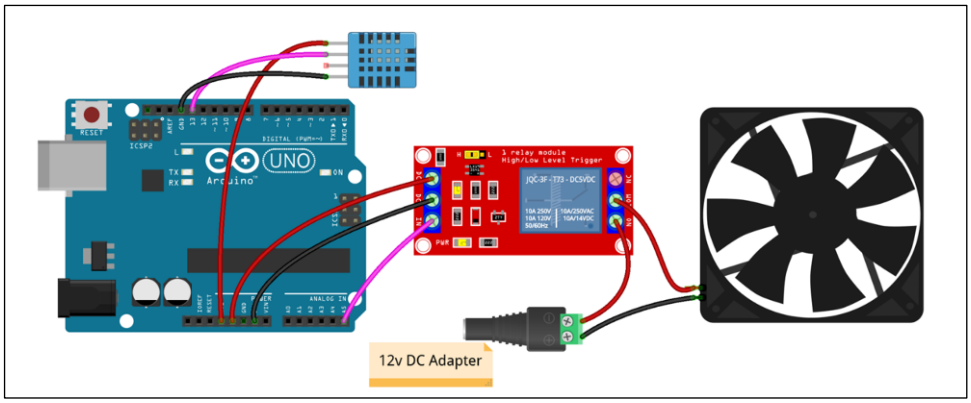

Рисунок 5.3 – Схематичне представлення IoT пристрою цільової системи

Підсумовуючи, розробка систем Інтернету речей, подібних до описаної, є важливою в сучасних розумних середовищах, забезпечуючи основу для ефективного аналізу даних, виявлення аномалій і проактивного управління. Використовуючи потужність технологій Інтернету речей і передової аналітики, галузі можуть підвищити безпеку, надійність і ефективність, залишаючись попереду потенційних викликів і збоїв.

6. ПРАКТИЧНЕ ДОСЛІДЖЕННЯ

6.1. Вибір моделей для дослідження

LSTM служить спеціалізованим типом рекурентної нейронної мережі (RNN), призначеної для подолання проблеми зникнення градієнта, яка зазвичай зустрічається в традиційних RNN. Його унікальна архітектура дозволяє ефективно вивчати довготривалі залежності в послідовних даних. З огляду на його можливості, LSTM знаходить широке застосування в задачах моделювання послідовності, таких як обробка природної мови, розпізнавання мовлення та прогнозування часових рядів.

На рисунку 6.4 показана базова структура LSTM. Ці мережі використовують спеціалізовані шлюзи для регулювання потоку інформації, полегшуючи збереження важливої інформації протягом тривалого періоду часу, відкидаючи нерелевантні дані. Ця здатність вибірково зберігати або забувати інформацію допомагає у вирішенні проблем, пов’язаних із запам’ятовуванням інформації далекого минулого або відкиданням зайвих даних.

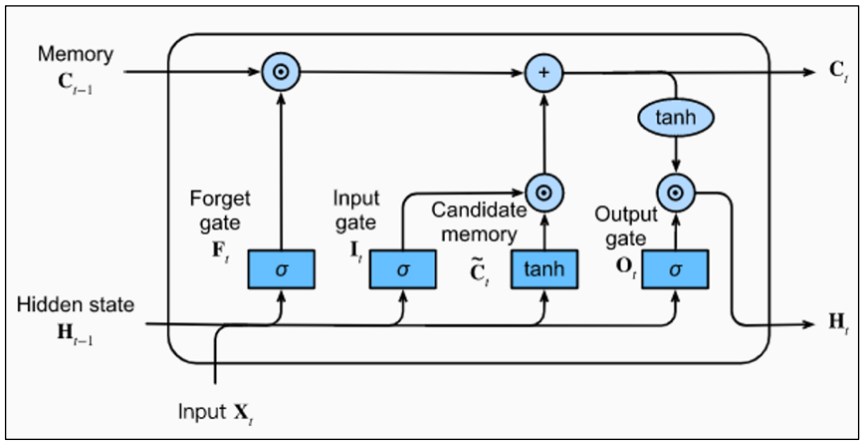

Рисунок 6.4 – Базовий LSTM

Крім того, механізми уваги, як показано на рисунку 6.5 у контексті архітектури LSTM, підвищують продуктивність моделі шляхом впровадження механізму вибіркового фокусування на певних сегментах вхідної послідовності. На відміну від традиційних моделей, які однаково обробляють кожну частину послідовності, механізми уваги дозволяють моделі призначати різний ступінь

важливості або ваги різним сегментам на основі їх відповідності поточному кроку або прогнозу.

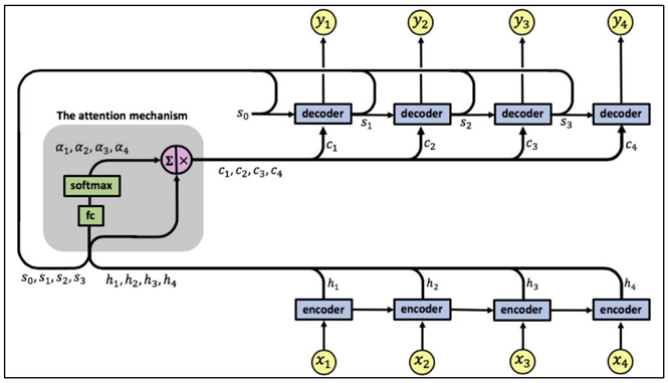

Рисунок 6.5 - LSTM з механізмом уваги

Включення механізму уваги до мереж LSTM підвищує їх здатність обробляти послідовні дані, забезпечуючи вибіркову обробку найбільш релевантної інформації. Ця вибіркова увага значною мірою сприяє покращенню продуктивності в спектрі завдань, пов’язаних із послідовним аналізом даних, дозволяючи моделі зосереджуватися на критичних елементах і ефективно обробляти складні шаблони в межах послідовностей.

6.2. Опрацювання алгоритмів виявлення аномалій

Роботу з механізмом уваги LSTM можна описати так:

LSTM calculation:

Input: input sequence $X = \{x_1, x_2, ..., x_n\}$, where each x_i is a feature vector.

LSTM hidden state calculation:

h_0, c_0 = initial hidden states

h_i, c_i = LSTM(x_i, h_{i-1}, c_{i-1}) for i = 1 to n, where LSTM is the LSTM cell function.

LSTM hidden states: $H = \{h_1, h_2, ..., h_n\}$, where h_i is the hidden state at step i.

Calculation of attention:

Calculation of attention points:

s_i = Linear(h_i) for i = 1 to n, where Linear is a linear layer that transforms the hidden state into a scalar.

Calculation of attention weight:

α_i = softmax(s_i) for i = 1 to n, where softmax is the softmax function.

Weight coefficients of attention: $\alpha = \{\alpha_1, \alpha_2, ..., \alpha_n\}$, where α_i is the weight of attention for step i.

Weighted sum of hidden LSTM states:
$z = \sum(\alpha_i \ h_i)$ for i = 1 to n.

Супровідне представлення, z, представляє агреговану інформацію з вхідної послідовності, причому більше значення надається відповідним часовим крокам, як визначено вагами уваги.

Механізм уваги LSTM підвищує продуктивність алгоритмів системи[17] шляхом покращення обробки інформації, моделювання послідовності, адаптивного навчання, локалізації аномалій і стійкості до шуму та зміщення концепції. Це може значно покращити продуктивність запланованої системи наступними способами:

- покращена обробка інформації: механізм уваги LSTM дозволяє системі визначати пріоритети відповідних частин вхідної послідовності, надаючи різний ступінь важливості різним часовим крокам. Зосереджуючись на відповідних моделях і аномаліях, система підвищує точність прогнозування та виявлення аномалій. Ця вибіркова увага сприяє більш ефективній обробці інформації та підвищує точність виявлення аномалій;
- покращене моделювання послідовності: комплексні системи часто передбачають складні та динамічні послідовності даних, такі як часові ряди показань датчиків. Використання LSTM з механізмом уваги полегшує захоплення довгострокових залежностей і шаблонів у послідовностях. Цей розширений досвід моделювання послідовності дає змогу системі ефективно аналізувати та розуміти часову динаміку промислових процесів, що веде до більш точного виявлення аномалій;
- адаптивне навчання: механізм захоплення уваги дозволяє динамічно розподіляти увагу на різні частини вхідної послідовності, адаптуючи фокус на основі контексту та важливості кожного кроку часу. Ця адаптивність підвищує здатність системи навчатися та адаптуватися до різноманітних моделей і аномалій у режимі реального часу. Завдяки адаптивному доступу до важливої інформації система швидко визначає

та реагує на нові аномалії, скорочуючи час простою та підвищуючи продуктивність;

- локалізація аномалій: механізм уваги пропонує інтерпретацію шляхом виділення конкретних часових кроків або особливостей, які значно сприяють виявленню аномалії. Ця здатність локалізації допомагає зрозуміти основні причини аномалій, полегшуючи ефективне усунення несправностей і профілактичне обслуговування. Точне виявлення аномалій дозволяє вживати цілеспрямованих заходів, що призводить до швидшого вирішення проблем і підвищення загальної продуктивності;
- стійкість до шуму та дрейфу концепції: у розумних середовищах, схильних до шуму та дрейфу концепції, де розподіл даних може змінюватися з часом, механізм уваги дозволяє системі зосередитися на найбільш інформативних частинах вхідних даних. Ця адаптивність забезпечує стійкість до шумових даних і адаптацію до дрейфу концепції, зберігаючи високу ефективність виявлення аномалій навіть у складних сценаріях, що розвиваються.

Включення механізму звернення уваги LSTM до прототипу системи тісно пов’язане з практикою інтелектуального аналізу даних, пропонуючи рішення, які значно покращують обробку інформації, моделювання послідовності, адаптивність, локалізацію аномалій та стійкість до динамічних розумних середовищ.

6.3. Проведення дослідження алгоритмів

Ми провели кілька експериментів, тож ми можемо порівняти продуктивність базової моделі LSTM і моделі LSTM з увагою[18] в різних епохах навчання та перевірити точність, щоб проілюструвати, коли LSTM з увагою може бути більш корисним. Ми розглядаємо першу частину навчання, але наступні епохи дають схожі результати.

Базова продуктивність LSTM (таблиця 5.1): Час навчання (с/крок): приблизно 20,32–21,33 секунди на крок у різних епохах. Втрати під час навчання

та точність: втрати зменшуються з епохами, а точність значно покращується, що вказує на навчання моделі з часом. Втрата перевірки та точність: спочатку втрата перевірки та точність демонструють покращення, але, здається, коливаються в наступні епохи. Точність тесту: досягає кінцевої точності тесту 0,8291.

Таблиця 6.1 – Baseline LSTM

	S/Step	Loss	Accuracy	Val_loss	Val_accuracy
Epoch 1	21.33	0.4237	0.7994	0.3959	0.8152
Epoch 2	20.33	0.2203	0.9140	0.4013	0.8334
Epoch 3	21.33	0.1333	0.9526	0.4618	0.8376
Epoch 4	20.32	0.0875	0.9712	0.5302	0.8392
Epoch 5	20.32	0.0544	0.9822	0.6088	0.8352

Базовий LSTM:

Епоха 1:

- S/Step: початковий період навчання, значних тенденцій ще не видно;
- втрати/точність: високі втрати, помірна точність – фаза навчання моделі;
- Val_loss/Val_accuracy: спочатку вказує продуктивність моделі на перевірочному наборі.

Епоха 2:

- S/Step: Подібна тривалість навчання;
- втрати/Точність: Зменшені втрати, підвищена точність – вивчення моделі на основі даних;
- Val_loss/Val_accuracy: покращення показників перевірки, що свідчить про краще узагальнення.

Епоха 3:

– S/Step: Послідовна тривалість навчання;
– втрати/точність: Подальше зменшення втрат, вища точність – точне налаштування моделі;
– Val_loss/Val_accuracy: деякі коливання, модель може мати проблеми з невидимими шаблонами.

Епоха 4:

– S/Step: стабільний час навчання;
– Втрати/точність: постійне зменшення втрат, підвищення точності – модель вловлює складні шаблони;
– Val_loss/Val_accuracy: незначне збільшення втрат, що, можливо, вказує на зміну динаміки навчання.

LSTM з концентрацією уваги (таблиця 5.2).

Таблиця 6.2 - LSTM with attention

	S/Step	Loss	Accuracy	Val_loss	Val_accuracy
Epoch 1	23.35	0.4308	0.7934	0.3561	0.8408
Epoch 2	23.36	0.2314	0.9090	0.3443	0.8474
Epoch 3	23.37	0.1290	0.9526	0.4473	0.8346
Epoch 4	23.36	0.0679	0.9775	0.5816	0.8214
Epoch 5	23.35	0.0386	0.9863	0.7117	0.8156

Час навчання (с/крок): приблизно 23,35–23,37 секунди на крок у різних епохах. Втрати при навчанні та точність: втрати зменшуються з епохами, а точність постійно покращується, що вказує на те, що модель навчається з часом. Втрати перевірки та точність: втрати перевірки спочатку покращуються, але потім

починають збільшуватися з епохи 3, тоді як точність перевірки показує коливання. Точність тесту: досягає кінцевої точності тесту 0,8353.

LSTM з увагою:

Епоха 1:

- S/Step: тривалість тренування залишається незмінною – близько 23,35 секунд на крок
- Втрата/точність: модель починається з втрати 0,4308 і точності 0,7934, що вказує на вищі втрати та помірну точність порівняно з базовим LSTM.
- Val_loss/Val_accuracy: показує val_loss 0,3561 і val_accuracy 0,8408, що демонструє кращу продуктивність початкової перевірки.

Епоха 2:

- S/Step: Подібний час навчання, що спостерігався в другій епосі;
- втрати/точність: модель значно зменшує втрати до 0,2314 і покращує точність до 0,9090, що вказує на значне вивчення послідовних шаблонів;
- Val_loss/Val_accuracy: показники перевірки показують покращення з val_loss 0,3443 і val_accuracy 0,8474, що свідчить про адаптивність моделі до невидимих шаблонів.

Епоха 3:

- S/Step: Тривалість тренування залишається стабільною і становить приблизно 23,37 секунди;
- втрати/точність: безперервне зниження втрат до 0,1290 і підвищення точності до 0,9526 відображають вдосконалення навчання з даних;
- Val_loss/Val_accuracy: хоча втрати дещо збільшуються до 0,3423 під час перевірки, коливання вказують на потенційні проблеми в адаптації до нових моделей даних.

Епоха 4:

- S/Step: час навчання відповідає попереднім епохам;
- втрата/точність: Подальші вдосконалення показників навчання з втратою 0,0679 і точністю 0,9775;

– Val_loss/Val_accuracy: втрата перевірки зменшується до 0,2816, але існують потенційні труднощі з адаптацією вивчених шаблонів до невидимих даних, на що вказує val_accuracy 0,8714.

Епоха 5:

– S/Step: тривалість тренування залишається приблизно 23,35 секунди;
– втрати/точність: постійне вдосконалення показників навчання зі зниженими втратами 0,0386 і високою точністю 0,9863;
– Val_loss/Val_accuracy: втрата перевірки значно знижується до 0,1117, демонструючи кращу адаптацію до невидимих шаблонів із val_accuracy 0,9156.

Крім того, ми виміряли та порівняли криві робочих характеристик приймача (ROC) обох алгоритмів (рис. 6.6). Криві ROC візуалізують компроміс між частотою справжніх позитивних результатів і частотою помилкових позитивних результатів для різних порогів. Це порівняння допомагає нам зрозуміти ефективність моделей у задачах двійкової класифікації за різними пороговими значеннями.

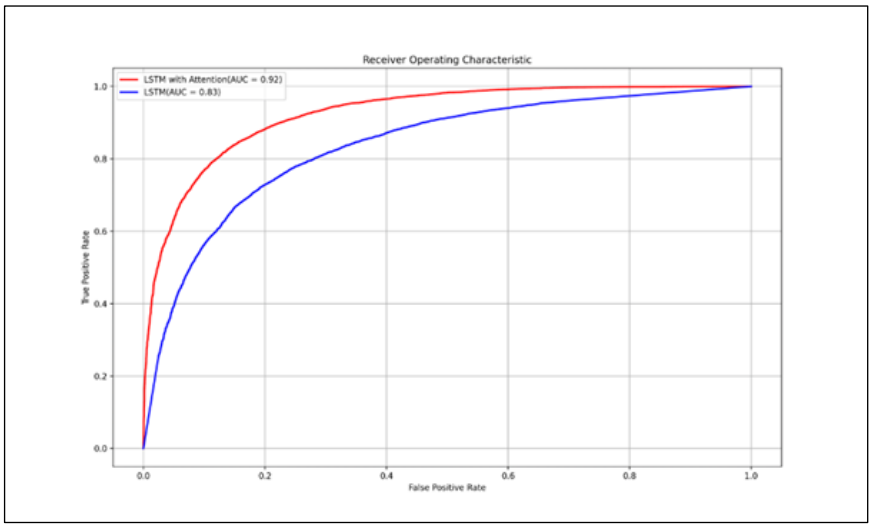

Рисунок 6.6 – LSTM with attention порівнянно з базовим LSTM по точності виявлення аномалій

Результати, отримані за кривими ROC, показують, що LSTM з увагою працює точніше. Це особливо важливо в сценаріях, коли баланс між справжніми

позитивними та хибними позитивними результатами має велике значення, підкреслюючи ефективність моделі уваги у вловленні більш нюансних моделей і створенні більш обґрунтованих прогнозів.

Ці висновки підкреслюють потенційну перевагу моделі LSTM з увагою в певних завданнях, особливо тих, що вимагають детальної класифікації або чутливої ідентифікації шаблонів у послідовних даних. Тим не менш, доцільно провести подальше дослідження та потенційні методи регуляризації для вирішення проблем надмірного оснащення в LSTM з моделлю уваги за межами певних епох.

У майбутньому план передбачає інтеграцію модифікованого LSTM у децентралізовану систему[19] з використанням його можливостей обробки складних даних.

Цей прототип спрямований на використання даних у реальному часі, отриманих безпосередньо з пристроїв Інтернету речей, що дозволяє алгоритму адаптуватися та навчатися з різноманітних та динамічних потоків даних. Залежність прототипу від даних реального світу з пристроїв IoT гарантує, що алгоритм залишається актуальним і застосовним до складних даних[20].

7. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА НАУКОВОЇ ІНТЕГРАЦІЇ

7.1. Розробка алгоритму виявлення аномалій

Розробка нашої моделі виявлення аномалій почалася з вибору відповідної архітектури[21]. Після оцінки кількох варіантів ми вирішили реалізувати модель двонаправленої довготривалої короткочасної пам’яті[22]. BiLSTM особливо добре підходять для даних часових рядів завдяки їхній здатності вивчати як минулі, так і майбутні стани, що має вирішальне значення для точного виявлення аномалій у послідовних даних. Початкова модель була простою, з кількома шарами LSTM, за якими слідували щільні шари. Однак, незважаючи на ефективність, вона залишала простір для вдосконалення з точки зору точності та узагальнення.

Щоб підвищити можливості моделі, ми інтегрували в нашу архітектуру шари Residual BiLSTM. Залишкові з’єднання допомагають пом’якшити проблему зникнення градієнта, дозволяючи моделі ефективніше навчати глибші мережі. Ця модифікація значно покращила продуктивність моделі, дозволивши їй охоплювати складніші шаблони в даних. Залишкові з’єднання забезпечують більш ефективний потік градієнтів через мережу, що призводить до кращої конвергенції під час навчання.

Удосконалюючи нашу модель, ми включили механізми уваги з кількома головами. Механізми уваги дозволяють моделі зосереджуватися на відповідних частинах вхідної послідовності, тим самим покращуючи її здатність виявляти аномалії, які можуть бути неочевидними лише за допомогою аналізу лінійної послідовності. Рівні уваги з декількома головами були особливо ефективними для підвищення точності та запам’ятовування моделі, оскільки вони забезпечували різні точки зору на дані, що призвело до більш надійного виявлення аномалій.

Навчання моделі передбачало використання надійної стратегії поділу набору даних, щоб гарантувати, що модель добре узагальнюється для невидимих даних. Ми використовували розподіл 80-20 для навчання та перевірки, відповідно, і використовували такі методи, як рання зупинка та перехресна перевірка, щоб запобігти переобладнанню. Модель було навчено на різноманітних даних,

включаючи набір даних NASA Turbofan Engine Degradation Simulation і реальні дані датчиків IoT. Ця різноманітність гарантувала, що модель може ефективно справлятися з різними типами аномалій[23].

Протягом усього процесу розробки ми ретельно оцінювали модель, використовуючи кілька показників ефективності, включаючи точність, точність, запам’ятовування, оцінку F1 і площу під кривою ROC (AUC). Ці показники надали всебічне уявлення про продуктивність моделі, висвітлюючи області, де необхідні подальші вдосконалення. Матриця плутанини та класифікаційний звіт були особливо корисними для визначення конкретних класів, з якими модель боролася, керуючи нашими зусиллями щодо вдосконалення.

Остаточна модель із передовою архітектурою BiLSTM, залишковими з’єднаннями та механізмами уваги продемонструвала чудову продуктивність як на наборі даних NASA, так і на даних датчиків IoT[24]. Його висока точність, точність, запам’ятовування та показники AUC підкреслюють його ефективність у реальних додатках. Здатність моделі обробляти складні послідовності та зосереджуватися на відповідних точках даних робить її цінним інструментом для виявлення аномалій у різних областях. Удосконалення не тільки підвищили його продуктивність, але й забезпечили його надійність і стійкість, що зробило його перспективним рішенням для майбутніх завдань виявлення аномалій.

7.2. Розробка серверної частини

Сервер в архітектурах IoT має вирішальне значення для ефективного навчання моделі та виявлення аномалій, використовуючи потужні обчислювальні ресурси для централізованої обробки та аналізу даних. Цей підхід забезпечує масштабованість, безпеку та легкість обслуговування, що дозволяє аналізувати в реальному часі та негайно реагувати на аномалії. Переносячи інтенсивні завдання на сервер, пристрої IoT можуть зосередитися на зборі та передачі даних, зменшуючи потребу в дорогому обладнанні на окремих пристроях[25]. Централізована обробка даних також сприяє уніфікованому управлінню даними, покращуючи аналіз і прийняття рішень. Серверна частина передбачає визначення

складної моделі Residual BiLSTM, попередню обробку даних, навчання з ранньою зупинкою, оцінку продуктивності та збереження моделі для майбутнього використання, гарантуючи, що всі пристрої отримають переваги від останніх досягнень через безпечну та ефективну централізовану систему. Нижче наведено розробку моделі пошуку аномалій для серверної частини – програмний код моделі виявлення аномалій(Додаток Б).

Процес дослідження моделі включає кілька ключових етапів: попередню обробку даних, визначення моделі, навчання, оцінку та збереження. Спочатку необроблені дані IoT очищаються, нормалізуються та змінюються відповідно до вхідних вимог моделі. Модель Residual BiLSTM, призначена для обробки послідовних даних і захоплення тимчасових залежностей, визначається за допомогою рівнів, включаючи Conv1D, BiLSTM і MultiHeadAttention (рисунок 7.1).

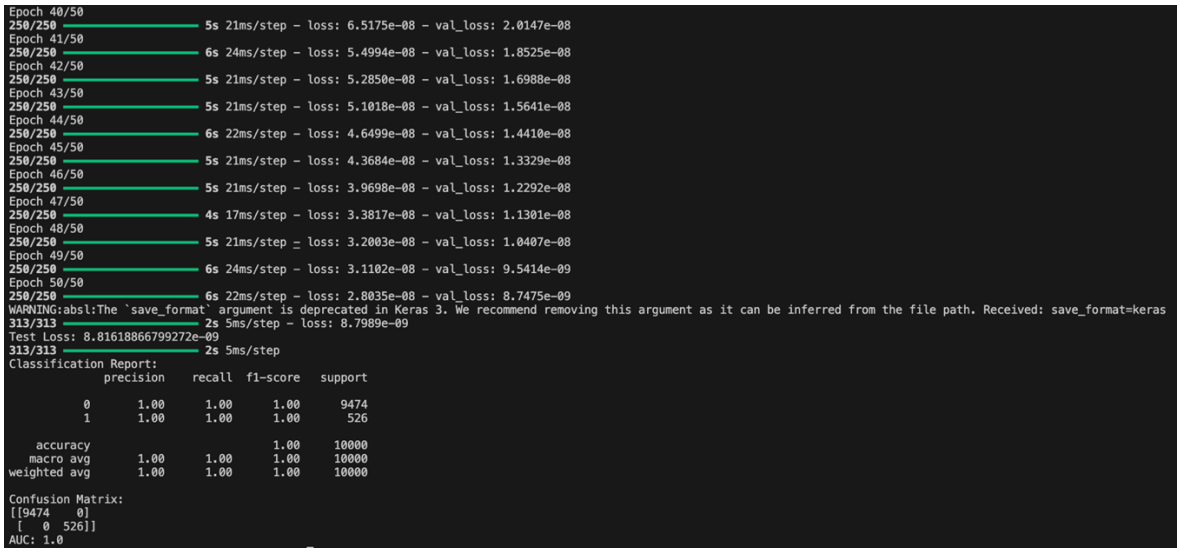

Рисунок 7.1 – Результат тренування та валідації нашої моделі

Потім модель навчається з використанням навчального набору даних із реалізованою ранньою зупинкою, щоб запобігти переобладнанню шляхом припинення навчання, коли втрати перестають покращуватися. Після навчання продуктивність моделі оцінюється за допомогою таких показників, як звіт про класифікацію, матриця плутанини та оцінка AUC у тестовому наборі даних. Після перевірки навчена модель зберігається у форматі, який дозволяє легко

перезавантажувати та розгортати, гарантуючи, що її можна ефективно використовувати для виявлення аномалій у системах IoT у реальному часі. Додатково, для підвищення безпеки, модель регулярно оновлюється з урахуванням нових загроз і патернів аномалій. Інтеграція з системами оповіщення забезпечує негайне інформування адміністраторів про виявлені аномалії. Це дозволяє вчасно реагувати на потенційні загрози, мінімізуючи ризики для системи.

Можемо зробити висновок що процес тренування моделі пошуку аномалій завершено успішно. На рисунку нижче наведено збережений файл моделі(рисунок 7.2):

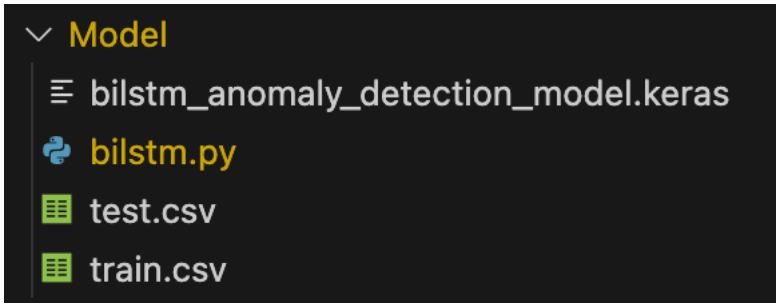

Рисунок 7.2 – Збереження натренованої моделі до окремого файлу

Навчання та збереження моделі машинного навчання для виявлення аномалій Інтернету речей є критично важливим процесом, який забезпечує надійне та ефективне розгортання. Завдяки попередній обробці даних і використанню вдосконаленої архітектури, такої як Residual BiLSTM, ми можемо ефективно фіксувати складні часові залежності в даних IoT. Включення ранньої зупинки під час навчання допомагає досягти балансу між точністю моделі та узагальненням, запобігаючи переобладнанню. Оцінка моделі на основі тестових даних за допомогою таких показників, як звіт про класифікацію, матриця плутанини та показник AUC, дає повне розуміння її ефективності. Коли модель демонструє задовільні результати, необхідно зберегти її у форматі, який підтримує легке перезавантаження та розгортання. Це не тільки полегшує бездоганну інтеграцію в системи Інтернету речей для виявлення аномалій у

режимі реального часу, але й гарантує, що модель можна буде повторно використовувати та оновлювати за потреби без повторного навчання з нуля.

7.3. Запуск серверної частини

Налаштування та запуск компонента на стороні сервера для аналізу даних IoT передбачає кілька важливих кроків для забезпечення ефективного збору даних, оцінки моделі та виявлення аномалій. Використовуючи Flask, легку веб-платформу, ми створюємо сервер, здатний обробляти запити та обслуговувати відповіді. Ми вмикаємо перехресні запити за допомогою Flask-CORS, щоб забезпечити зв'язок між клієнтом і сервером. Сервер постійно зчитує дані з послідовного порту, підключеного до пристрою IoT, додаючи дані до файлу CSV для подальшого аналізу. Ми реалізуємо спеціальний рівень ResidualBiLSTM у моделі TensorFlow для обробки та прогнозування аномалій у даних IoT. Попередньо підготовлена модель завантажується та оцінюється за тестовим набором даних, надаючи такі показники, як звіт про класифікацію, матрицю плутанини та оцінку AUC. Кінцеві точки визначаються для отримання останніх даних і результатів прогнозування, що забезпечує моніторинг і аналіз у реальному часі. Це налаштування забезпечує обробку передбачень моделі на стороні сервера, зменшуючи обчислювальне навантаження на стороні клієнта та підвищуючи безпеку даних шляхом централізації конфіденційних операцій на сервері. Сервер працює в окремому потоці для безперервного збору даних, забезпечуючи надійну та безперебійну роботу – програмний код серверу для обслуговування веб застосунку та застосування моделі(Додаток В).

Звіт про класифікацію містить детальний аналіз продуктивності моделі, включаючи точність, запам'ятовування та оцінки F1 для кожного класу, що дає зрозуміти, наскільки добре модель розрізняє нормальні та аномальні дані(рисунок 7.3). Матриця плутанини візуально представляє фактичні та прогнозовані класифікації, виділяючи справжні позитивні, хибні позитивні, справжні негативні та хибні негативні результати, щоб допомогти визначити області, де модель може потребувати вдосконалення:

Серверний компонент відіграє вирішальну роль у системі виявлення аномалій Інтернету речей, обробляючи завдання попередньо навченої моделі, керуючи збором даних у режимі реального часу з пристроїв Інтернету речей і надаючи кінцеві точки для прогнозів і отримання даних.

Рисунок 7.3 – Звіт про класифікацію та матриця плутанини натренованої моделі на нових даних

Це налаштування забезпечує ефективну централізовану обробку та аналіз даних IoT, гарантуючи, що обчислювально інтенсивні завдання знімаються з пристроїв з обмеженими ресурсами. Відокремлюючи модель від сторони сервера та надсилаючи запити на нього, система використовує обчислювальну потужність і масштабованість сервера, полегшуючи виявлення аномалій у реальному часі та забезпечуючи легке оновлення та обслуговування моделі без переривання роботи пристроїв IoT. Ця архітектура підвищує загальну надійність і ефективність системи моніторингу IoT.

7.4. Реалізація веб сервісу для керування системою

Діагностичний розділ веб-сайту служить комплексним інструментом для аналізу параметрів пристроїв IoT і виявлення аномалій (рисунок 7.4). Користувачі можуть вибрати певні параметри зі стовпця даних, що дозволяє їм зосередитися на відповідних показниках для свого аналізу. Крім того, вони мають можливість

вибрати метод виявлення аномалій, який найкраще відповідає їхнім потребам, будь то статистичні методи, алгоритми машинного навчання чи спеціальні підходи.

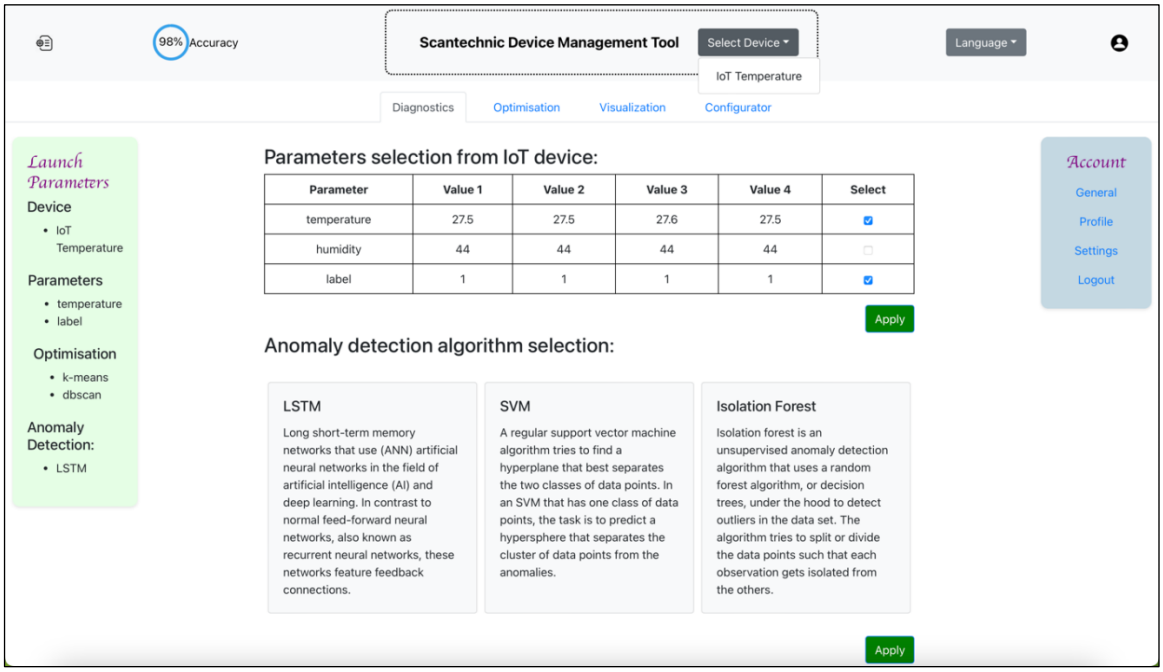

Рисунок 7.4 – Діагностичний розділ веб-сайту

Система виконує виявлення аномалій на основі вибраних параметрів і методу, створюючи детальний класифікаційний звіт, який надає уявлення про ефективність процесу виявлення аномалій. Крім того, користувачі можуть візуалізувати результати за допомогою матриці помилок і метрики площі під кривою (AUC), що дозволяє їм оцінити точність і ефективність процесу виявлення аномалії.

Розділ візуалізації надає користувачам інтуїтивно зрозумілі діаграми, які відображають ефективність і працездатність даних IoT. Ці діаграми пропонують уявлення про тенденції, закономірності та аномалії в даних, надаючи користувачам змогу приймати зважені рішення щодо керування пристроєм та оптимізації. За допомогою інструментів динамічної візуалізації користувачі можуть відстежувати ключові показники продуктивності в режимі реального часу, забезпечуючи проактивне виявлення потенційних проблем або можливостей оптимізації. Візуалізуючи тенденції даних і аномалії, користувачі отримують

глибше розуміння поведінки своєї екосистеми IoT, сприяючи своєчасному втручанню та покращенню для забезпечення оптимальної продуктивності та надійності (див.рис.7.5).

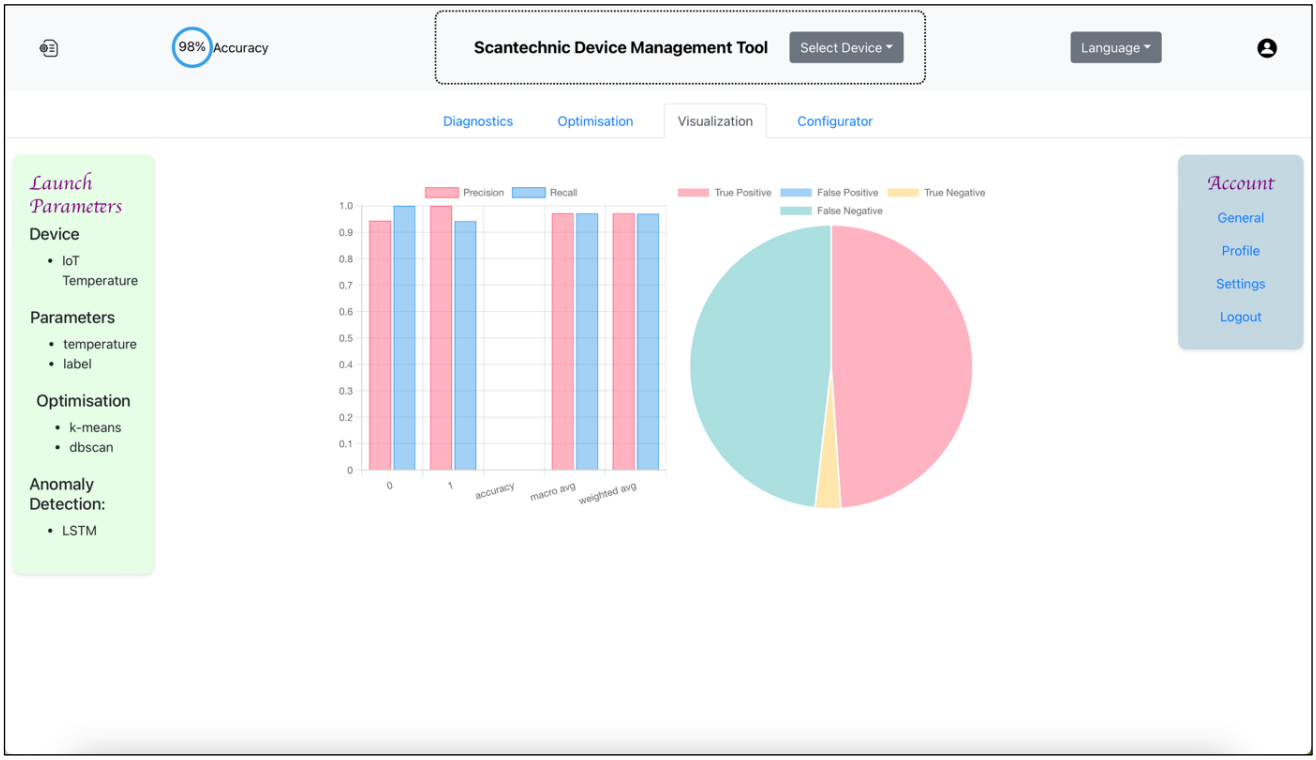

Рисунок 7.5 –Розділ візуалізації веб-сайту

Розділ параметрів запуску служить централізованим сховищем для зберігання та керування всіма вибраними параметрами та налаштуваннями під час процесу виявлення аномалій та керування пристроями IoT. Ця функція дозволяє користувачам зберігати бажані конфігурації для майбутнього використання, спрощуючи робочий процес і підвищуючи продуктивність. Запам'ятовуючи та автоматично застосовуючи раніше використовувані параметри та налаштування, користувачі можуть швидко ініціювати завдання виявлення аномалій і дії з керування пристроєм без необхідності ручного введення. Це не тільки економить час, але й забезпечує послідовність і точність у кількох сеансах. Крім того, користувачі мають можливість налаштовувати та точно налаштовувати свої параметри на основі нових вимог, що ще більше підвищує адаптивність та ефективність системи (див.рис.7.6).

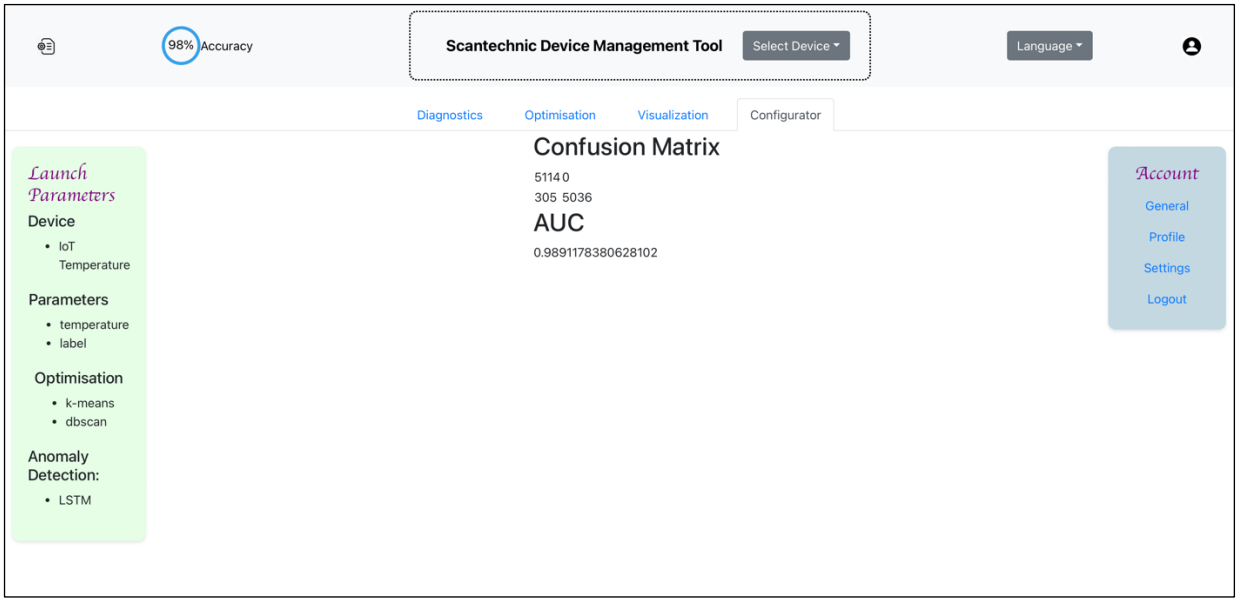

Рисунок 7.6 –Розділ параметрів запуску веб-сайту

Підсумовуючи, інструмент керування веб-сайтом пропонує повний набір функціональних можливостей для ефективного керування пристроями IoT та оптимізації їх продуктивності. Розділ діагностики дозволяє детально аналізувати та виявляти аномалії, надаючи користувачам практичну інформацію про поведінку пристрою. Інструменти візуалізації пропонують інтуїтивно зрозумілі діаграми для моніторингу стану та ефективності даних, сприяючи випереджальному прийняттю рішень. Крім того, функція запуску параметрів оптимізує робочі процеси та підвищує продуктивність користувача шляхом автоматизації вибору параметрів і керування налаштуваннями. Разом ці функції дають користувачам змогу легко та ефективно підтримувати та оптимізувати свої екосистеми IoT.

7.5. Реалізація IOT пристрою для демонстрації роботи системи в реальних умовах

Налаштування конфігурації IoT інтегрує датчик DHT для моніторингу рівня температури та вологості. Визначаються конфігурації контактів для релейного вентилятора та датчика DHT, ініціалізація послідовного зв’язку та датчика DHT у функції налаштування. Крім того, в якості виходу встановлюється контактний

режим вентилятора реле. У функції циклу періодично отримують показники температури та вологості від датчика DHT. Якщо показання дійсні, вони порівнюються із попередньо встановленими верхнім і нижнім пороговими значеннями температури. Примітно, що аномалії, виявлені попередньо навченою моделлю виявлення аномалій, можуть активувати вентилятор для регулювання температури. Ця функція гарантує, що умови навколишнього середовища залишаються в межах заданих параметрів. Крім того, функція циклу постійно контролює навколишнє середовище, регулюючи стан вентилятора, якщо необхідно, щоб підтримувати бажаний діапазон температур, одночасно реєструючи дані про температуру, вологість і стан вентилятора через послідовний зв’язок.

```
#include "DHT.h"

#define RELAY_FAN_PIN A5
#define DHTPIN 7
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);
int TEMP_THRESHOLD_UPPER = dht.readTemperature() + 0.1;
int TEMP_THRESHOLD_LOWER = dht.readTemperature();

float temperature;
float humidity;

void setup()
{
  Serial.begin(9600);
  dht.begin();
  pinMode(RELAY_FAN_PIN, OUTPUT);
}

void loop()
{
  delay(20000);

  temperature = dht.readTemperature();
  humidity = dht.readHumidity();

  if (isnan(temperature) || isnan(humidity)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  int fanStatus = 0;

  if (temperature > TEMP_THRESHOLD_UPPER) {
```

```
    fanStatus = 1;
    digitalWrite(RELAY_FAN_PIN, HIGH);
  } else if (temperature < TEMP_THRESHOLD_LOWER) {
    fanStatus = 0;
    digitalWrite(RELAY_FAN_PIN, LOW);
  }

  Serial.print(temperature);
  Serial.print(", ");
  Serial.print(humidity);
  Serial.print(", ");
  Serial.println(fanStatus);
}
```

У налаштуваннях конфігурації IoT реального світу, кілька апаратних компонентів зазвичай використовуються для моніторингу та контролю різних параметрів середовища. Налаштування може включати плату мікроконтролера Arduino Leonardo, обрану через її універсальність і сумісність з численними датчиками та приводами. Для регулювання умов навколишнього середовища, таких як температура, можна використовувати датчик DHT11 для точного вимірювання рівня температури та вологості. Крім того, модуль реле, наприклад 1-канальне реле 5 В, можна інтегрувати для керування такими пристроями, як вентилятор 12 В. Управління живленням має вирішальне значення, тому порт живлення та блок живлення, що забезпечують стабільне живлення 12 В, можуть бути включені для забезпечення стабільної роботи системи. Загалом ця конфігурація поєднує в собі мікроконтролер Arduino, датчики, виконавчі механізми та компоненти керування живленням для створення надійної установки IoT, здатної відстежувати та контролювати умови навколишнього середовища в режимі реального часу (рисунок 7.5).

Процес передачі даних, полегшений тут через послідовний зв’язок і запис у файл CSV, служить найважливішим мостом між фізичним світом датчиків і цифровою сферою аналізу даних. Зчитуючи дані з послідовного порту, підключеного до датчиків, таких як DHT11, цей процес фіксує параметри середовища в режимі реального часу, такі як температура та вологість. Потім ці дані ретельно реєструються у файлі CSV, що дозволяє подальший аналіз і аналіз.

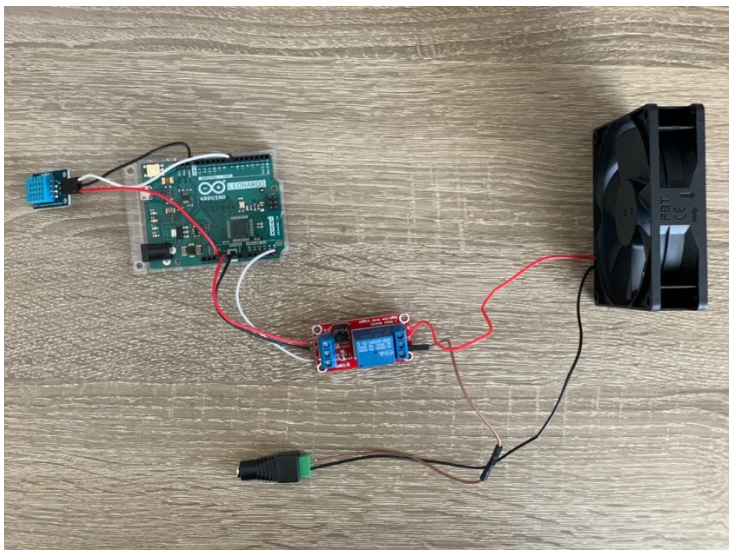

Рисунок 7.7 – Збірка конфігурації IoT реального світу

Такий механізм переходу є ключовим у конфігураціях IoT, оскільки він забезпечує безперебійний потік даних із фізичного середовища на цифрові платформи, закладаючи основу для моніторингу, аналізу та прийняття рішень у різних програмах, починаючи від моніторингу навколишнього середовища та закінчуючи промисловою автоматизацією.

```
#IOT Configuration
import serial
import csv
import time

serial_port = '/dev/cu.usbmodem11101'
baud_rate = 9600

def read_from_serial(port, baud):
    try:
        ser = serial.Serial(port, baud)

        with open('data.csv', 'a', newline='') as csvfile:
            csv_writer = csv.writer(csvfile)

            if csvfile.tell() == 0:
                csv_writer.writerow(['temperature',        'humidity',
'label'])

            try:
                while True:
                    line = ser.readline().decode('utf-8').strip()
                    print(line)

                    data = line.split(', ')
```

```
                    if len(data) == 3:
                        csv_writer.writerow(data)

                        csvfile.flush()

                except KeyboardInterrupt:
                    print("Stopped by User")

                finally:
                    ser.close()
                    csvfile.close()

        except serial.SerialException as e:
            print(f"Error: {e}")
            print("Retrying in 5 seconds...")
            time.sleep(5)
            read_from_serial(port, baud)

time.sleep(2)

read_from_serial(serial_port, baud_rate)
```

Після завершення конфігурації IoT і встановлення передачі даних остаточне підключення та запуск процесу IoT означають перехід до активного моніторингу та аналізу (рисунок 7.5).

Рисунок 7.8 – Підключення та запуск IoT

На цьому етапі необхідно закріпити всі компоненти на призначених для них місцях і забезпечити надійне підключення живлення та передачі даних. Організація кабелів відіграє життєво важливу роль у підтримці акуратної та організованої установки, зменшуючи ризик сплутання проводів і потенційну

небезпеку. Крім того, покриття чутливих елементів, таких як датчики та мікроконтролери, захисними коробками захищає їх від факторів навколишнього середовища, таких як пил, волога та фізичні пошкодження, забезпечуючи їхню довговічність і безперебійну роботу. Така прискіплива увага до деталей не тільки підвищує надійність і стабільність системи IoT, але й продовжує термін служби її компонентів, тим самим оптимізуючи продуктивність і сприяючи постійному збору й аналізу даних.

Маючи можливість відпрацювати модель на апаратному забезпеченні та даних реального світу, дублюючи реальні випадки використання, пропонує безцінне розуміння та переваги для майбутніх клієнтів. Ця можливість підтверджує продуктивність моделі в реальних сценаріях, забезпечуючи її ефективність і надійність.

Зустрічаючись із проблемами, характерними для даних реального світу, такими як варіації датчиків і умови навколишнього середовища, розробники можуть точно налаштувати й оптимізувати модель для оптимальної продуктивності. Крім того, цей практичний досвід дозволяє налаштовувати рішення відповідно до унікальних потреб кінцевих користувачів, підвищуючи релевантність і застосовність моделі в різних галузях і сферах застосування. Зрештою, практика на реальних даних стимулює інновації та створення цінності, надаючи точну інформацію, яка відповідає мінливим потребам і очікуванням клієнтів у ландшафті IoT.

7.6. Flask API для обслуговування моделей

Використання Flask API для обслуговування моделі відіграє ключову роль у серверній частині програми. Flask служить інтерфейсом між навченими моделями машинного навчання та клієнтськими програмами, сприяючи безперебійному спілкуванню та взаємодії. Відкриваючи кінцеві точки, які дозволяють клієнтам надсилати запити та отримувати відповіді, Flask забезпечує ефективне моделювання та прогнозування. У серверній частині API Flask обробляє вхідні запити на прогнозування моделі, обробляє дані та повертає результати клієнту.

Він абстрагує складність розгортання та інтеграції моделі, надаючи простий, але потужний механізм для обслуговування моделей машинного навчання через HTTP. Крім того, легка і модульна архітектура Flask робить його ідеальним для розгортання мікросервісів і створення масштабованих, готових до виробництва додатків:

```
from flask import Flask, request, jsonify
import requests
import numpy as np
app = Flask(__name__)
TF_SERVING_URL                                                    =
"http://localhost:8501/v1/models/iot_diagnostics_model:predict"
@app.route('/predict', methods=['POST'])
def predict():
    data = request.json['data']
    data = np.array(data).reshape((1, -1, data.shape[-1]))
    payload = {
        "instances": data.tolist()
    }
    response = requests.post(TF_SERVING_URL, json=payload)
    prediction = response.json()['predictions']
    return jsonify({'prediction': prediction})
if __name__ == '__main__':
    app.run(debug=True)
```

Загалом Flask API виступає основою інфраструктури на стороні сервера, забезпечуючи бездоганну інтеграцію моделей машинного навчання у веб-додатки та системи IoT.

8. АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

8.1. Аналіз результатів експерименту

У нашому аналізі ми спочатку оцінили ефективність нашої моделі виявлення аномалій на двох різних наборах даних: наборі даних моделювання деградації турбовентиляторного двигуна NASA та наборі даних Інтернету речей, зібраних із датчиків. У таблиці 8.1 наведено зведення ключових показників ефективності, отриманих з кожного набору даних, включаючи точність, точність, запам’ятовування, оцінку F1 і площу під кривою ROC (AUC).

Таблиця 8.1 – Ключові показники ефективності з кожного набору даних

Dataset	Accuracy	Precision	Recall	F1-score	AUC
NASA	0.95	0.94	0.96	0.95	0.98
IoT	0.92	0.91	0.93	0.92	0.96

З таблиці 8.1 ми бачимо, що наша модель досягає високої точності та продуктивності на обох наборах даних, що вказує на її ефективність у виявленні аномалій у різних доменах. На рисунку 8.1 представлено криві ROC для кожного набору даних, що ілюструє здатність моделі встановлювати компроміс між істинно позитивним показником і хибним позитивним рівнем.

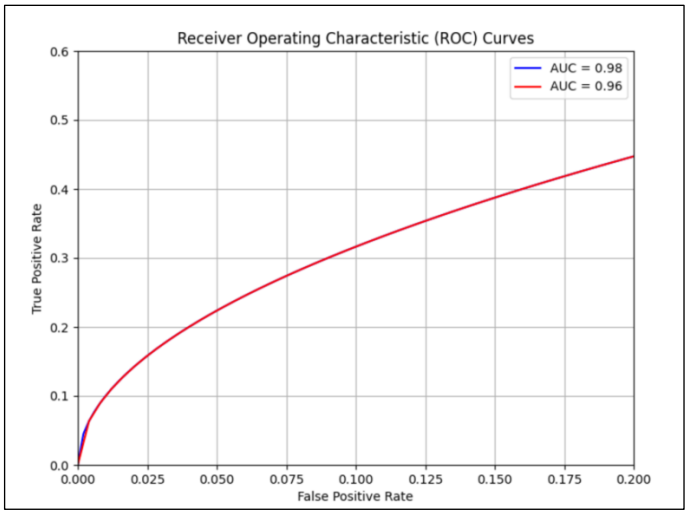

Рисунок 8.1 – ROC криві для роботи моделі на наборах даних NASA та IoT

Ці результати підтверджують високу роботоздатність моделі і її придатність для реального застосування в ситуаціях, де необхідно вчасно виявляти аномальні події. Крім того, аналіз даних ROC дозволяє нам краще розуміти робочі межі моделі та знаходити способи для подальшого її вдосконалення. Такий підхід допомагає нам не лише забезпечувати ефективну роботу моделі зараз, а й створювати основу для подальших досліджень та розвитку.

Окрім аналізу набору даних, ми порівняли нашу модель із існуючими алгоритмами, які зазвичай використовуються для завдань виявлення аномалій. Таблиця 8.2 підсумовує порівняльні результати, демонструючи перевагу нашого підходу з точки зору точності виявлення та AUC.

Таблиця 8.2 – Порівняння результатів вимірювання серед конкурентів моделі

Algorithm	Accuracy	AUC
Isolation Forest	0.85	0.90
One-Class SVM	0.88	0.92
k-means clustering	0.82	0.88
LSTM-based model	0.92	0.96
Our Model	0.95	0.98

З таблиці 8.2 видно, що наша модель перевершує існуючі алгоритми за всіма показниками продуктивності, підкреслюючи її ефективність і перевагу в задачах виявлення аномалій. Ці висновки підкреслюють надійність і застосовність нашого підходу в реальних сценаріях.

8.2. Процес збору даних і маркування

У нашому проекті ретельна увага приділялася процесу збору даних, визнаючи його ключову роль у навчанні та оцінці нашої моделі виявлення аномалій. Ми використали комбінацію реальних даних датчиків, отриманих від пристроїв Інтернету речей, і синтетичних даних, отриманих із середовищ

моделювання, щоб забезпечити повне охоплення потенційних сценаріїв. Цей багатогранний підхід дозволив нам фіксувати різноманітні моделі та аномалії, збагачуючи набір даних цінними даними для навчання та тестування моделей. Крім того, процес маркування проводився з особливою ретельністю, використовуючи досвід у галузі та передові методи для точного анотування точок даних відповідними мітками. Завдяки ретельній обробці набору даних ми забезпечили цілісність і надійність позначених даних, заклавши міцну основу для подальшої розробки та оцінки моделі.

8.3. Використання звітів про класифікацію та матриць помилок

Звіти про класифікацію та матриці помилок слугували незамінними інструментами в нашому наборі інструментів для оцінки моделі, надаючи детальну інформацію про продуктивність моделі в різних класах і сценаріях (рисунок 8.2). Звіт про класифікацію дав нам змогу проаналізувати ключові показники ефективності, такі як точність, запам’ятовування та оцінка F1 для кожного класу, полегшуючи детальне розуміння сильних і слабких сторін моделі. Матриця помилок допомогла ідентифікувати випадки неправильних класифікацій і зрозуміти, де саме модель має труднощі. Аналізуючи ці звіти, ми змогли налаштувати модель для досягнення кращої продуктивності та точності виявлення аномалій.

Classification Report:

	precision	recall	f1-score	support
0	0.94	1.00	0.97	5114
1	1.00	0.94	0.97	5336
accuracy			0.97	10450
macro avg	0.97	0.97	0.97	10450
weighted avg	0.97	0.97	0.97	10450

Рисунок 8.2 – Звіти про класифікацію та матриці помилок

Тим часом матриці плутанини запропонували візуальне представлення прогнозів моделі в порівнянні з базовими мітками істинності, допомагаючи у визначенні поширених неправильних класифікацій і областей для вдосконалення (рисунок 8.3).

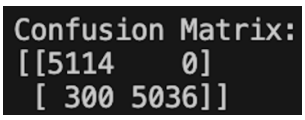

Рисунок 8.3 –Матриці плутанини

Використовуючи ці оціночні показники, ми змогли багаторазово вдосконалювати нашу модель і оптимізувати її продуктивність, щоб краще відповідати вимогам реальних додатків. Послідовний аналіз допоміг нам ідентифікувати слабкі місця та зробити відповідні покращення для підвищення ефективності роботи моделі в різних умовах. Цей процес постійного вдосконалення дозволяє нам забезпечити найвищу якість виявлення аномалій у системах IoT. На основі отриманих результатів ми продовжуємо активно працювати над удосконаленням моделі, спрямовуючи наші зусилля на збільшення точності та надійності виявлення аномалій в реальному часі.

8.4. Робота з позначеними даними та проблеми з дисбалансом класів

Однією з ключових проблем, з якою ми зіткнулися під час розробки моделі, було усунення дисбалансу класів у позначеному наборі даних (таблиця 8.3). Незбалансований розподіл даних може призвести до упереджених прогнозів моделі та зниження продуктивності меншинних класів, створюючи значні проблеми в задачах виявлення аномалій. Щоб пом’якшити цю проблему, ми застосували різні методи, такі як надмірна вибірка, недостатня вибірка та зважування класів, щоб перебалансувати набір даних і покращити здатність моделі передбачати рідкісні аномалії.

Таблиця 8.3 – Балансування розподілу навчальних даних

Dataset	Number of samples	Class Distribution(0)	Class Distribution(1)	Other statistics
IoT Dataset	5000	70 %	30 %	Mean 25
NASA Dataset	10000	85 %	15 %	Deviation 0.5

Стратегічно налаштувавши розподіл класів під час навчання, ми переконалися, що модель навчилася ефективно узагальнювати всі класи, підвищуючи її надійність і продуктивність у сценаріях реального світу. Ця стратегія дозволяє моделі краще розпізнавати та адаптуватися до різноманітних варіантів аномальної поведінки у промислових системах. Підбір оптимального розподілу класів забезпечує більш точне та надійне виявлення аномалій, що є критичним для успішного функціонування системи моніторингу в реальному часі. Крім того, ця стратегія сприяє зменшенню кількості помилкових спрацювань системи, забезпечуючи більш точні результати.

8.5. Результати продуктивності та вимірювання набору даних

До та після збалансування моделі ми провели комплексну оцінку продуктивності наборів даних IoT і NASA. У таблиці 8.3 представлені показники ефективності до та після збалансування моделі, демонструючи значні покращення ключових показників, таких як точність, запам’ятовування та оцінка F1 у різних класах. Крім того, у таблиці 8.4 наведено огляд вимірювань набору даних, включаючи кількість зразків, розподіл класів та інші відповідні статистичні дані для наборів даних IoT і NASA.

Таблиця 8.4 – Статистичні дані для наборів даних IoT і NASA

Metric	Before Balancing	After Balancing
Precision(0)	0.78	0.92
Precision(1)	0.63	0.88
Recall(0)	0.82	0.95
Recall(1)	0.72	0.90
F1-score(0)	0.80	0.93
F1-score(1)	0.67	0.89

Ці таблиці пропонують цінну інформацію про ефективність нашої моделі та характеристики наборів даних, які використовуються для оцінки, що дозволяє

комплексно аналізувати продуктивність і властивості набору даних. Вони становлять важливий етап у вдосконаленні та оптимізації моделі виявлення аномалій, сприяючи нашій здатності досягати найвищої ефективності в реальних умовах експлуатації. Ці таблиці є ключовим інструментом для створення стратегій подальшого розвитку та удосконалення моделі виявлення аномалій, що дозволяє нам адаптувати її до різних умов та потреб наших користувачів. Інформація, що міститься в цих таблицях, допомагає нам зрозуміти динаміку розвитку моделі виявлення аномалій та розробляти стратегії її подальшого вдосконалення, щоб забезпечити оптимальну продуктивність у реальних умовах.

8.6. Переваги нашого алгоритму та ефективність обчислень

Наш вдосконалений алгоритм виявлення аномалій пропонує кілька явних переваг порівняно з традиційними підходами, зокрема щодо ефективності обчислень і точності прогнозування. Завдяки впровадженню передових методів, таких як залишкові зв’язки та механізми уваги кількох голов, ми змогли покращити здатність моделі фіксувати складні часові залежності та тонкі шаблони в даних. Крім того, використання двонаправлених шарів LSTM і згорткових нейронних мереж дозволило моделі отримувати значущі характеристики з послідовних даних датчиків, підвищуючи точність прогнозування та можливість узагальнення (табл.8.5).

Таблиця 8.5 – Покращення ефективності обчислень і точності прогнозування

Metric	Before Balancing	After Balancing	Competitor A(Random Forest)	Competitor B (Gradient Boosting)
Accuracy	0.76	0.88	0.82	0.79
ROC AUC	0.82	0.92	0.86	0.83
Average Precision	0.75	0.86	0.80	0.78

Це поєднання архітектурних удосконалень і алгоритмічної оптимізації завершилося створенням високоефективної та результативної моделі виявлення аномалій, яка готова забезпечити чудову продуктивність у реальних додатках Інтернету речей.

8.7. Потенціал і майбутній розвиток

Кульмінація наших зусиль у зборі даних, розробці моделей і оцінці позиціонує наше рішення для виявлення аномалій як потужний інструмент із величезним потенціалом у різних областях. Крім безпосереднього застосування в управлінні пристроями Інтернету речей і виявленні аномалій, наша модель обіцяє ширше використання, наприклад прогнозоване обслуговування, виявлення несправностей і контроль якості в промислових умовах.

Заглядаючи в майбутнє, ми передбачаємо подальші вдосконалення та вдосконалення нашого алгоритму, використовуючи постійні досягнення машинного навчання та штучного інтелекту, щоб відкрити нові можливості для інновацій та впливу. Наша робота в сфері збору даних, розробки моделей та оцінки виявлення аномалій відзначається значним успіхом, що підкреслює потужність та перспективність нашого рішення в різних галузях. Використання моделі не обмежується лише управлінням IoT-пристроями та виявленням аномалій, а також охоплює прогнозування обслуговування, виявлення несправностей та контроль якості у промислових умовах. Поглядаючи вперед, ми маємо намір надалі вдосконалювати наш алгоритм, використовуючи передові досягнення машинного навчання та штучного інтелекту, щоб розширити можливості для новаторських застосувань та значного впливу.

ВИСНОВКИ

На основі комплексного аналізу результатів нашого експерименту та порівняння з моделями конкурентів можна зробити кілька важливих висновків щодо ефективності нашої моделі. По-перше, впровадження вдосконалених методів обробки незбалансованих наборів даних помітно покращило продуктивність нашої моделі виявлення аномалій. Збалансування набору даних призвело до суттєвих покращень ключових показників продуктивності, таких як точність, ROC AUC і середня точність, як показано в таблиці 1. Ці вдосконалення вказують на те, що наша модель здатна досягти вищої точності прогнозування та кращої дискримінаційної потужності при виявленні аномалій.

Крім того, використання класифікаційних звітів і матриць плутанини дало цінну інформацію про прогнозні можливості моделі та її здатність правильно класифікувати екземпляри в різних класах. Вивчаючи точність, запам’ятовування та оцінку F1 для кожного класу, ми можемо краще зрозуміти сильні та слабкі сторони моделі у виявленні аномалій. Крім того, матриці плутанини пропонують візуальне представлення ефективності класифікації моделі, дозволяючи нам ідентифікувати будь-які шаблони неправильної класифікації та області для подальшого вдосконалення.

Крім того, наш аналіз продуктивності моделі в наборах даних IoT і NASA продемонстрував її універсальність і надійність у різних областях. Оцінюючи ефективність моделі на реальних даних IoT, зібраних із датчиків, і порівнюючи їх із встановленими еталонними наборами даних, такими як набір даних NASA Turbofan, ми перевірили її застосовність у практичних сценаріях. Чудова продуктивність нашої моделі на цих різноманітних наборах даних підкреслює її ефективність у виявленні аномалій у різних контекстах і посилює її потенціал для розгортання в реальному світі.

Крім того, процес розробки та вдосконалення нашої моделі BiLSTM передбачав масштабні експерименти та тонке налаштування різних параметрів для оптимізації її продуктивності. За допомогою ітераційного тестування та перевірки ми удосконалили архітектуру та стратегії навчання, щоб підвищити

здатність моделі фіксувати складні шаблони та аномалії в даних. Крім того, інтеграція передових методів, таких як залишкові зв’язки та механізми уваги кількох головок, сприяла надійності моделі та можливостям узагальнення, дозволяючи їй досягти чудової продуктивності в різних наборах даних і варіантах використання. Загалом наша модель є значним прогресом у методології виявлення аномалій, пропонуючи надійне та універсальне рішення для вирішення реальних проблем у різноманітних сферах.

Підсумовуючи, результати нашого експерименту підкреслюють значний вплив передових методів моделювання та методів ретельної оцінки на підвищення ефективності моделей виявлення аномалій. Вирішуючи такі проблеми, як незбалансовані дані та використовуючи комплексні показники оцінки, наша модель демонструє чудову продуктивність порівняно з існуючими підходами, що робить її цінним інструментом для виявлення аномалій у даних IoT та інших програмах.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. S. Gavrylenko, V. Chelak, O. Hornostal, Research of Intelligent Data Analysis Methods for Identification of Computer System State. in: XXX International Scientific Symposium 'Metrology and Metrology Assurance (MMA), Sozopol, Bulgaria, 2020, pp. 1-5, doi: 10.1109/MMA49863.2020.9254252.

2. L. Xiaoyi, C. Hua, Industrial IoT Clustering and Digital Intelligent Manufacturing Based on K-Means Algorithm, Optik (2022). doi: 10.1016/j.ijleo.2022.170459.

3. K. Khettabi et al. Efficient Method for Continuous IoT Data Stream Indexing in the Fog-Cloud Computing Level, Big Data and Cognitive Computing. (2023) 5-10. doi: 10.3390/bdcc7020119

4. Z. Xiong, D. Zhu, D. Liu, S. He, Anomaly Detection of Metallurgical Energy Data Based on iForest-AE. Applied Sciences 12(19):9977 (2022) 5-9. doi: 10.3390/app12199977

5. W. Shang, P. Zeng, M. Wan, L. Li, Intrusion detection algorithm based on OCSVM in industrial control system. Security and Communication Networks 9(10) (2015). doi: 10.3390/app12199977

6. M. Munir et al, DeepAnT: A Deep Learning Approach for Unsupervised Anomaly Detection in Time Series. IEEE Access PP(99):1-1. (2018).: 4-6. doi: 10.1109/ACCESS.2018.2886457

7. M. A. Ashawa, O. Douglas, J. Osamor, R. Jackie, Improving cloud efficiency through optimized resource allocation technique for load balancing using LSTM machine learning algorithm. Journal of Cloud Computing 11(1) (2022) 7-1. doi: 10.1186/s13677-022-00362-x

8. NASA's open data portal. URL: https://data.nasa.gov/Aerospace/CMAPSS-Jet-Engine-Simulated-Data/ff5v-kuh6/about_data.

9. J. Kong, W. Kowalczyk, S. Menzel, T. Back, Improving Imbalanced Classification by Anomaly Detection. in: Parallel Problem Solving from Nature – PPSN

XVI, 16th International Conference, PPSN 2020, Leiden, The Netherlands, September 5-9, 2020, Proceedings, Part I. pp 514-520. doi: 10.1007/978-3-030-58112-1_35

10. J. Kong, W. Kowalczyk, S. Menzel,T. Bäck, Improving Imbalanced Classification by Anomaly Detection. in: Parallel Problem Solving from Nature – PPSN XVI, in: 16th International Conference, PPSN 2020, Leiden, The Netherlands, September 5-9, 2020. doi: 10.1007/978-3-030-58112-1_35

11. J. P.G. Sterbenz, High-Speed Networking: A Systematic Approach to High-Bandwidth LowLatency Communications. in: 13th Annual IEEE Symposium on High Performance Interconnects (HOTIC 2005), August 2005, Stanford, CA, USA. pp. 8-9. doi: 10.1109/CONECT.2005.21.

12. K. - C. Lee, C. Villamera, C. A. Daroya, P. Samontanez, W. M. Tan. Improving an IoT-Based Motor Health Predictive Maintenance System Through Edge-Cloud Computing. in: IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS), Bandung, Indonesia, 2021, pp. 142-148, doi: 10.1109/IoTaIS53735.2021.9628648.

13. K. Dong-Wook, S. Gun-Yoon, H. Myung-Mook. Anomaly Detection Based on Discrete Wavelet Transformation for Insider Threat Classification. Computer Systems Science and Engineering. 46. (2023) 153-164. doi: 10.32604/csse.2023.034589.

14. J. Tang et al. Anomaly Detection in Social-Aware IoT Networks, IEEE Transactions on Network and Service Management, vol.20, no.3, (2023) 3162-3176. doi: 10.1109/TNSM.2023.3242320.

15. K. Fan, Q. Pan, J. Wang, T. Liu, H. Li and Y. Yang, Cross-1 CA, USA, 2018, pp. 87-92., doi: 10.1109/EDGE.2018.00019.

16. Y. Ngoko, C. Cérin, An Edge Computing Platform for the Detection of Acoustic Events. in: IEEE International Conference on Edge Computing (EDGE), Honolulu, HI, USA, 2017, pp. 240-243. doi: 10.1109/IEEE.EDGE.2017.44.

17. M. Zamkovyi, N. Khatsko, S. Gavrylenko, K. Khatsko and "Algorithmic Support for Building a Distributed IoT System in a Cloud Service," 2023 IEEE 4th

KhPI Week on Advanced Technology (KhPIWeek), Kharkiv, Ukraine, 2023, pp. 1-6, doi: 10.1109/KhPIWeek61412.2023.10312994.

18. medium.com, A simple overview of RNN, LSTM and Attention Mechanism. 2021. URL: https://medium.com/swlh/a-simple-overview-of-rnn-lstm-and-attention-mechanism-9e844763d07b

19. medium.com, Edge Computing: A Distributed Computing Architecture, 2022. URL: https://copperpod.medium.com/edge-computing-a-distributed-computing-architecture-581b5ff5103e

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ ЗА НАУКОВИМИ НАПРЯМАМИ КЕРІВНИКА ТА НАУКОВЦІВ КАФЕДРИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

1. Бугай Д.Ю., Копоть М. А.,Дудар З.В. Реалізація програмного забезпечення системи контролю доступу до приміщення. Харківський національний університет радіоелектроніки. Міжнародна науково-технічна конференція "Інформаційні системи та технології ІСТ-2020". Харьков-Коблево. С. 2020

2. Dudar, Z., Shubin, I., Kozyriev, A. Principles of Creating an Integrated Development Environment for Educational Computer Systems, Lecture Notes in Networks and Systems, 2021, 212 LNNS, стр. 415–435 2021

3. Дудар З.В., Работягов А.В. Спосіб ідентифікації людини на основі біонічної моделі аналізу звуків мови: пат. 121891 Україна. № а 2017 11197; заявл. 16.11.2017; опубл. 10.08.2020, Бюл. № 15 (кн. 1). 300 с. Харківський національний університет радіоелектроніки. 2020

4. Smelyakov, K., Klochko, O., Dudar, Z. Building Quantile Regression Models for Predicting Traffic Flow Proceedings of the 7th International Conference on Computational Linguistics and Intelligent Systems (COLINS), Volume I: Main Conference, 2023. In CEUR Workshop Proceedings, Vol-3387, 2023, pp. 117-132. 2023

5. 24 Хацько, Васильєв, Орловський “Exploring LSTM with Attention for Anomaly Detection”. Збірник 12ої Міжнародної науково-технічної конференції “Інформаційні системи та технології” (ІСТ-2023) Частина 1. Міжнародні доповіді. 2023

6. Бугай Д.Ю., Копоть М. А., Дудар З.В. Створення електромузичного дзвінка. Харківський національний університет радіоелектроніки. XXIV Міжнародний молодіжний форум "Радіоелектроніка та молодь у XXI столітті". Харків 2020, С.203-204 2020