МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «магістр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)
на тему:

«Пошук аномалій в роботі систем автоматизованого керування з використанням машинного навчання»

Виконав студент групи ІПЗм-22-1
ІВАНОВ Іван Іванович

Керівник роботи:
ВЛАСЕНКО Олег Васильович

Рецензент:
ТОЛСТОЙ Ігор Анатолійович

Житомир – 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»
Зав. кафедри інженерії
програмного забезпечення
Тетяна ВАКАЛЮК
«18» вересня 2023 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ІВАНОВ Іван Іванович**
Керівник роботи: **ВЛАСЕНКО Олег Васильович**
Тема роботи: **«Пошук аномалій в роботі систем автоматизованого керування з використанням машинного навчання»**, затверджена Наказом закладу вищої освіти від **«18» вересня 2023 р. № 479/с**
Вихідні дані для роботи: **аналіз та пошук аномальних значень у роботі систем автоматизованого керування та впровадження нейронних мереж в процес пошуку аномалій; способи автоматизації процесу пошуку аномальних значень.**
Керівники / консультанти з магістерської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	**Керівник/ консультант**	Завдання видав	Завдання прийняв
1	Власенко О.В.	22.09.2023р.	22.09.2023р.
2	Власенко О.В.	22.09.2023р.	22.09.2023р.
3	Власенко О.В.	22.09.2023р.	22.09.2023р.

РЕФЕРАТ

Кваліфікаційна робота освітнього ступеня «магістр» присвячена автоматизації процесу пошуку аномалій у роботі автоматизованих систем. Пояснювальна записка складається з вступу, трьох розділів та списку використаної літератури. Пояснювальна записка до кваліфікаційної магістерської роботи містить 54 сторінки, 52 ілюстрацій, 1 таблицю, 4 додатки та 26 посилань на зовнішні джерела.

Метою роботи є створення аналізатора даних роботи автоматизованих систем та розробка зручного графічного інтерфейсу для користування.

В роботі описано процес створення програми, яка аналізує дані, які повертаються з відповідних датчиків автоматизованих систем та розробку інтерфейу. Досліджено способи навчання нейронної мережі з довгою короткочасною пам’яттю. Проведено тестування програми.

КЛЮЧОВІ СЛОВА: ДОВГА КОРОТКОЧАСНА ПАМ’ЯТЬ, НЕЙРОННА МЕРЕЖА, ДАТАСЕТ, АНАЛІЗ ДАНИХ.

ABSTRACT

The master's thesis is devoted to the automation of the process of finding anomalies in the operation of automated systems. The explanatory note consists of an introduction, three chapters and a list of used literature. The thesis note contains 54 pages, 52 illustrations, 1 table, 4 appendix and 26 links to external sources.

The purpose of the work is to create a data analyzer of automated systems and develop a user-friendly graphical interface.

The work describes the process of creating a program that analyzes the data returned from the corresponding sensors of automated systems and the development of the interface. The methods of learning a neural network with long short-term memory have been studied. The program has been tested.

KEY WORDS: LONG SHORT-TERM MEMORY, NEURAL NETWORK, DATASET, DATA ANALYSIS.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

NN – (англ. neural network) – нейронна мережа

STD – (англ. standard deviation) – середнє квадратичне відхилення

LSTM – (англ. long short-term memory) – довга короткочасна пам’ять

ВСТУП

Актуальність теми. Проблема виявлення аномалій у складних системах автоматизованого керування є доволі розповсюдженою у різних сферах людської життєдіяльності. Дані проблеми можуть намагатися вирішувати по-різному, залежності від бюджету та можливостей обслуговування різноманітних систем.

Візьмемо до прикладу станції генераторів електроенергії у важкодоступних місцях нашої планети. Контролювання роботи систем є складним, наприклад, у зв'язку з погодними умовами та утриманням персоналу у важкодоступних місцях. У тому числі потрібно враховувати складність контролювання даних, які повертаються з різноманітних датчиків, які повертають зовсім різні значення у різних діапазонах. Цю проблему можна спробувати вирішити декількома способами.

Самим очевидним з них є цілодобове спостереження за різними показниками використовуючи персонал. Але вирішуючи проблему таким чином, збільшується кількість персоналу, ускладнюється графік робіт, та підвищується ризик людського фактору.

Іншим варіантом вирішення є фіксовані значення мінімуму та максимуму для датчиків. Але проблеми такого підходу полягають в ускладненні налаштування значень для кожного з датчиків окремо та поновлення значень для кожного окремо у разі зміни діапазону роботи датчиків, що сповільнює та ускладнює роботу подібних систем.

Хорошим варіантом є створення додатку для пошуку аномалій в роботі систем автоматизованого керування з використанням машинного навчання. Завдяки машинному навчанню система буде автоматично адаптуватися до всього різноманіття діапазонів та значень, за якими ведеться нагляд.

Мета роботи – розробка системи за допомогою якої з'являється можливість легко контролювати аномалії в роботі систем автоматизованого керування.

Для досягнення поставленої мети необхідно вирішити такі основні завдання:

1. Дослідити роботу систем автоматизованого керування.
2. Дослідити методи машинного навчання для виявлення аномалій.
3. Створити додаток з використанням машинного навчання, який виявляє аномалії у значеннях систем автоматизованого керування.

Об’єктом дослідження є робота операторів різноманітних станцій, які контролюють роботу систем автоматизованого керування.

Предметом дослідження є методи та алгоритми машинного навчання спрямовані на виявлення аномалій у системах автоматизованого керування.

Наукова новизна роботи полягає у використанні методів аналізу даних в архітектурі нейронної мережі LSTM для реалізації пошуку аномальних значень у роботі автоматизованих систем, які повертають значення різних діапазонів з різною періодичністю. Особливістю даного пошуку аномалій є адаптація програми до різних графіків, та пошук аномалій у різноманітних діапазонах, що розширює можливості програмного додатку для різних сфер.

Практичне значення полягає у…

Апробація. За темою кваліфікаційної магістерської роботи було опубліковано статтю у закордонному виданні [1] та 2 тез конференції [2, 3].

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Аналіз проблем і особливостей процесу виявлення аномалій

Виявлення аномалій (або виявлення викидів) – це ідентифікація рідкісних елементів, подій або спостережень, які викликають підозри через суттєві відмінності від більшості даних. Як правило, аномальні дані можуть бути пов’язані з якоюсь проблемою чи рідкісною подією, наприклад, банківське шахрайство, медичні проблеми, структурні дефекти, несправне обладнання тощо. Цей зв’язок робить дуже цікавим можливість вибрати, які точки даних можна вважати аномаліями, оскільки ідентифікація цих подій зазвичай дуже цікава з точки зору бізнесу.

Це підводить нас до однієї з ключових цілей: як визначити, чи є точки даних нормальними чи аномальними? У деяких простих випадках, як на прикладі рис. 1.1, візуалізація даних може надати нам важливу інформацію.

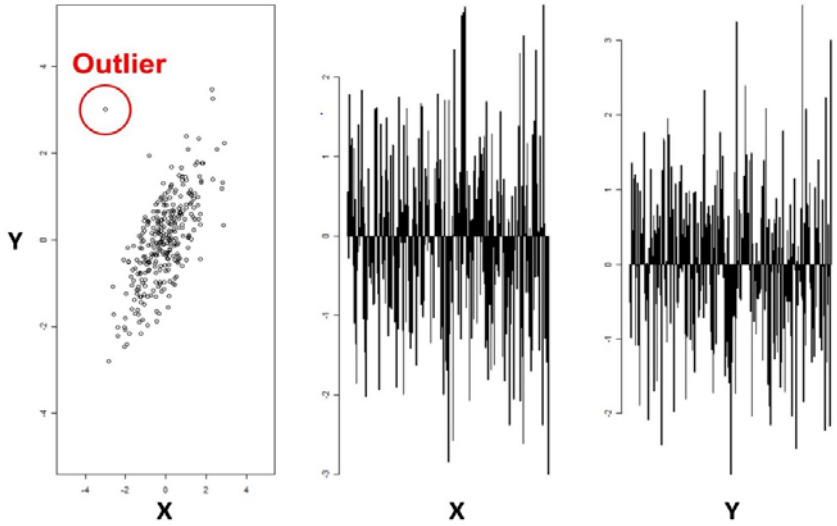

Рис. 1.1. Виявлення аномалії для двох змінних

У цьому випадку двовимірних даних (X і Y) стає досить легко візуально визначити аномалії через точки даних, розташовані за межами типового розподілу. Однак, дивлячись на малюнки праворуч, неможливо ідентифікувати викид безпосередньо, досліджуючи одну змінну за раз: саме комбінація змінних X і Y дозволяє нам легко ідентифікувати аномалію. Це

суттєво ускладнює справу, коли ми масштабуємо від двох змінних до 10-100 змінних, що часто буває в практичних застосуваннях виявлення аномалій.

Будь-яка машина, чи то обертова (насос, компресор, газова чи парова турбіна тощо), чи не обертова машина (теплообмінник, дистиляційна колона, клапан тощо), з часом досягне точки поганого стану. Цей момент може бути не фактичним збоєм або вимкненням, а моментом, коли обладнання більше не працює в оптимальному стані. Це сигналізує про те, що може знадобитися певне технічне обслуговування для відновлення повного робочого потенціалу. Простіше кажучи, визначення «стану працездатності» нашого обладнання є сферою моніторингу стану.

Найпоширеніший спосіб виконання моніторингу стану – перегляд кожного вимірювання датчика машини та встановлення для нього мінімального та максимального обмеження значення. Якщо поточне значення знаходиться в межах, то машина справна. Якщо поточне значення виходить за межі, це означає, що машина несправна, і надсилається сигнал тривоги.

Відомо, що ця процедура встановлення жорстко закодованих обмежень тривоги надсилає велику кількість хибних тривог, тобто тривог для ситуацій, які насправді є справними для машини. Є також відсутні сигнали тривоги, тобто ситуації, які є проблемними, але не викликають тривоги. Перша проблема призводить довитрати часу та сил. Друга проблема є більш важливою, оскільки вона призводить до реальних пошкоджень разом з витратами на ремонт і втратою продуктивності.

Обидві проблеми можуть бути результатом однієї причини: людська помилка. Навіть якщо посадити декількох операторів переглядати усі значення датчику, станція не досягне енергоефективного нагляду за обладнанням, бо людина може пропустити якесь значення, а також на перегляд об’ємних даних потрібно багато часу. Саме тому варто використовувати автоматизовані методи пошуку аномалій. Таким чином можна уникнути людської помилки, і зосередити увагу обслуговуючого

персоналу на вже знайдені додатком аномалії, замість перегляду усього об’єму даних.

Уявімо певну станцію, кожна з них, з великою вірогідністю буде мати певне обладнання, що має зовсім різні параметри, одиниці вимірювання, кількість датчиків та ін. Таким чином постає питання контролю усього обладнання, ведення звітності, та ускладнюється процес виявлення аномальних значень, якщо враховувати пропорційний ріст станцій до росту персоналу. У результаті зі збільшенням обладнання все складнішим стає процес виявлення неправильної поведінки певних автоматизованих систем.

Зазвичай у такому разі, наймається більше персоналу, більше операторів, які ведуть звітність та намагаються виявити візуально неправильність роботи тієї чи іншої автоматизованої системи. Або ж розроблюється різні додатки, які є налаштовані саме для певних автоматизованих систем. Тому треба зауважити що універсальність використання грає дуже важливу роль у подальших проектуваннях та подальшій розробці програмного продукту, так як зазначалося вище різне оладнання має зовсім різний набір датчиків, від простих датчиків температури до складних комплексних датчиків, які можуть вимірювати тиск, напругу, швидкість та ін.

1.2. Аналіз наявних методів автоматизації процесу виявлення аномалій

Автоматизація процесу виявлення аномалій спрямована в основному на вдосконалення моніторингу роботи різноманітних автоматизованих систем, а також підвищення ефективності догляду за обладнанням на різноманітних станціях. Наразі існує декілька додатків де можна вести моніторинг обладнання.

Одним із найпопулярніших і доволі зручних є додаток 60HertzEnergy. Його перевагами є доволі зручний інтерфейс, завдяки якому можна створювати та налаштовувати станції та обладнання під кожну з них окремо.

Програмне забезпечення 60HertzEnergy допомагає комунальним підприємствам і власникам активів, незалежно від розміру, реалізувати ретельний, легко виконуваний план, щоб гарантувати ефективне проведення перевірок і скоротити дорогі виїзди на станції. Завдяки даному програмному забезпеченню доволі легко встановлювати контроль за обладнанням різноманітних автоматизованих систем.

Одним з головних недоліків даного програмного продукту є те що аномальні значення обладнання вираховуються завдяки фіксованим значенням діапазонів. Це означає що якщо датчик повертає значення за межами певного, попередньо встановленого, діапазону, користувач буде бачити у звітності помилку. Такий підхід підходить далеко не для усіх типів значень, які повертаються з автоматизованих систем. Це буде призводити до великої кількості хибних повідомлень про помилку, або ж навпаки, коли значення було дійсно аномальним, існує вірогідність того що воно потрапить у діапазон допустимих значень, та не викличе помилки. Другий варіант є особливо небезпечним, тому що призведе до невчасно виявленої несправності автоматизованої системи, що у свою чергу призведе до некоректної роботи, у кращому випадку, та до поломки, у гіршому. Інтерфейс даного програмного додатку показано на рис. 1.2.

Іншим прикладом подібної системи є collectiveFleet. Інтерфейс зображено на рис 1.3. Основною позитивною особливістю є простий інтерфейс користувача. За допомогою collectiveFleet компанії можуть ідентифікувати надмірно та недостатньо використовувані транспортні засоби, скоротити витрати шляхом усунення непотрібних запасів, оптимізувати загальний робочий процес і точно звітувати про різні аспекти операцій автопарку.

Рішення містить повний набір інструментів для управління автопарком: відстеження транспортних засобів і активів, управління технічним обслуговуванням, управління запасами запчастин, ризики та гарантії, ліцензії

та сертифікати, а також звіти. Завдяки управлінню техобслуговуванням користувачі мають доступ до історії ремонту автомобіля, графіків технічного обслуговування та робочих нарядів, а також можуть отримувати сповіщення електронною поштою про час обслуговування.

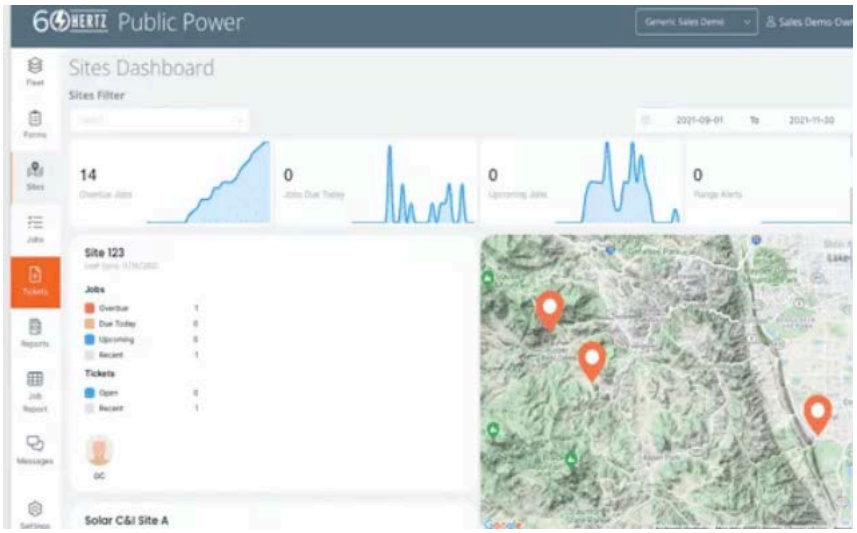

Рис. 1.2. Інтерфейс додатку 60HertzEnergy

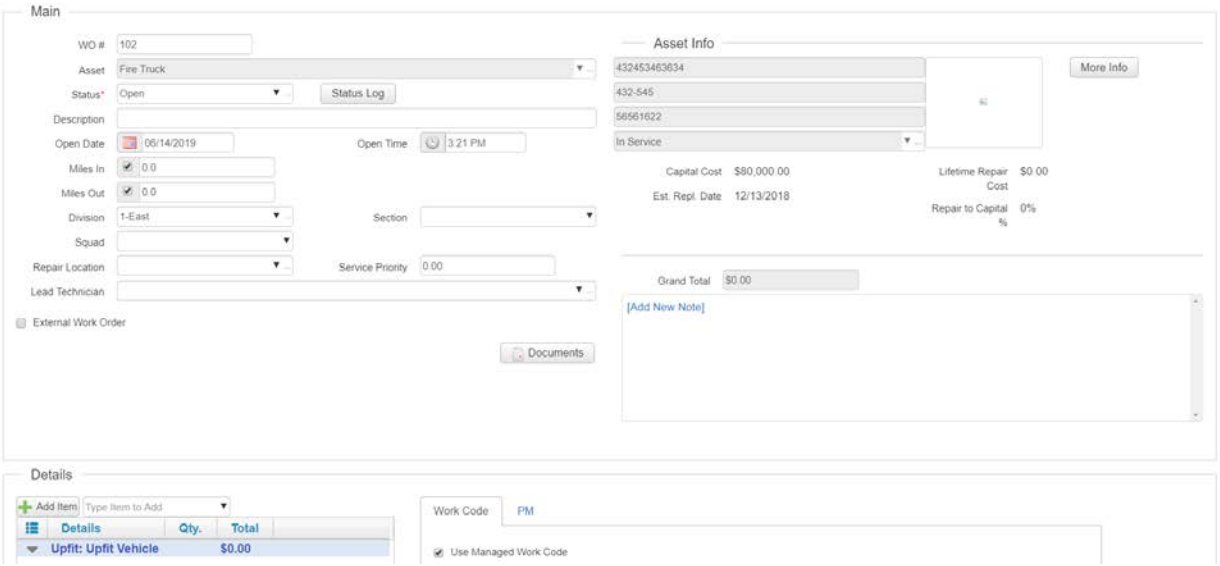

Рис. 1.3. Інтерфейс додатку collectiveFleet

Недоліками даного програмного забезпечення є вузька спеціальність програмного продукту, яка підходить тільки лдя обліку станцій, які стосуються логістики, що негативно впливає на універсальність даного програмного додатку.

Ще одним з прикладів є Web Work Azzier – це комп’ютеризована система керування технічним обслуговуванням (CMMS), розроблена для галузей із великими капітальними активами, яка пропонує коригувальний, профілактичний та аварійний контроль обслуговування в рамках пакету. Інтерфейс зображено на рис. 1.4. Рішення доступне як у хмарному, так і в локальному варіантах розгортання.

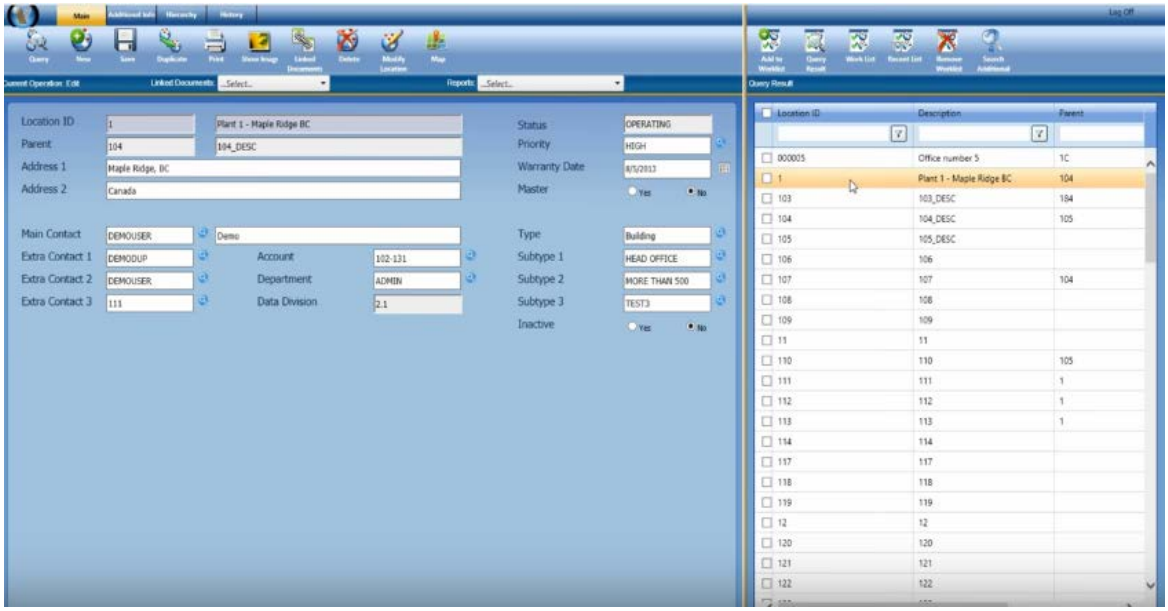

Рис. 1.4. Інтерфейс додатку Web Work Azzier

Azzier має функцію керування робочими замовленнями, де користувачі можуть налаштовувати свої форми робочих нарядів відповідно до галузевих вимог. Робочі замовлення можна призначати необхідним людям, а користувачі можуть стежити за ходом роботи на інформаційній панелі в режимі реального часу та виявляти несправності за допомогою чітко зазначених обмежень.

Недоліком цього додатку є незрозумілий інтерфейс, також проблема виявлення аномалій із строгим значенням діапазону, що у свою чергу призводить до вищезгаданих наслідків.

Таблиця 1.1

Порівняння інструментів для контролю та виявлення аномалій

	60HertzEnergy	collectiveFleet	Web Work Azzier
Доступ	Внутрішній додаток	Внутрішній додаток	Внутрішній додаток
Універсальність	Так	Має вузьку спеціалізацію	Так
Зручність користування	Середній	Середній	Поганий
Виявлення аномалій	Має встановлення чітких діапазонів	Має встановлення чітких діапазонів	Має встановлення чітких діапазонів
Мова програмування	JavaScript, Python	Невідомо	Невідомо

1.3. Постановка задачі дослідження

Проаналізувавши усі переваги та недоліки можна виявити основні аспекти, які мають бути виправлені та покращені у майбутньому додатку. Самим головним недоліком попередньо згаданих програмних додатків є відсутність автоматичного механізму виявлення аномалій у автоматизованих системах. Іншим, не менш важливим недоліком, є незручність або ж нелогічність деяких інтерфейсів програмних додатків, що у свою чергу сповільнює навчанню користування ними. Також основним недоліком є вузька направленість та обмеженість певних програмних додатків, що відображається на універсальності та охопленні ринку у різних сферах.

При постановці задачі слід враховувати, що багато з подібних додатків має облік значень, які повертаються та фіксуються з різноманітних автоматизованих систем, але вони мають доволі обмежений функціонал, який дозволяє виявляти аномалії лише за допомогою чітко встановленого мінімуму та максимуму відповідного діапазону.

Аналізуючи переваги та недоліки, можемо сформувати перше уявлення про те, як має функціонувати програмний додаток. Він повинен мати зручний інтерфейс, повинен бути універсальним для ринку, та має мати аналіз даних реалізований методом штучного інтелекту для спрощення виявлення аномалій.

На даному етапі варто розділити створення програмного додатку на етапи, для спрощення проектування та подальної реалізації. На першому етапі варто розробити програмний додаток, який буде отримувати дані з автоматизованих систем, структурувати та очищувати дані. На другому етапі варто продумати зручний інтерфейс для організації відповідної структури станцій, обліку автоматизованих систем, які вони мають. На третьому етапі варто розробити модель машинного навчання для виявлення аномальних значень.

1.4. Вибір засобів реалізації

Для аналізу даних використовується багато різноманітних методів, але доцільно обґрунтовано підійти до вибору, так як для кожної проблеми є більш підходяще рішення, яке вплине на подальшу роботу програмного додатку та як результат може повпливати на роботу самого бізнесу.

Для розпізнавання аномалій добре підходить нейронна мережа LSTM.

LSTM (від англ. Long short-term memory) – це штучна нейронна мережа, яка використовується в галузі штучного інтелекту та глибокого навчання. На відміну від стандартних нейронних мереж прямого зв’язку, LSTM має зворотні зв’язки. Така рекурентна нейронна мережа (RNN) може обробляти не лише окремі точки даних (наприклад, зображення), але й цілі послідовності даних (наприклад, мову чи відео). Завдяки тому що дана нейронна мережа може пам’ятати послідовності даних, це допомагає визначити поведінку певного датчика і надалі перевіряти чи збігаються наступні дані з такою поведінкою. Це означає що дану нейронну мережу можна використати для

багатьох зовсім різноманітних графіків, яку мають певну послідовну поведінку що допомагає уніфікувати додаток для використання у зовсім різному обладнанні.

На відміну від будь-яких інших методів аналізу даних LSTM підходить дуже добре підходить для виявлення аномальних значень на графіку. Але для того щоб провести аналіз даних для початку потрібно підготувати дані. Для початку потрібно виконати стандартизацію даних. Це потрібно для коректної роботи нейронної мережі.

Стандартна оцінка зразка x обчислюється за формулою z = (x - u) / c, де u – середнє значення навчальних вибірок, а s – стандартне відхилення навчальних вибірок. Центрування та масштабування відбуваються незалежно для кожної функції шляхом обчислення відповідних статистичних даних на зразках у навчальному наборі. Середнє значення та стандартне відхилення потім зберігаються для подальшого використання в даних за допомогою перетворення.

Стандартизація набору даних є загальною вимогою для багатьох оцінювачів машинного навчання: вони можуть мати погані результати, якщо окремі функції тим чи іншим не схожі на стандартні дані з нормальним розподілом (наприклад, гауссове значення з нульовим середнім і одиничною дисперсією).

Після стандартизації набору даних, їх варто розділити на тренувальні та тестові вибірки. Зазвичай вибірка для трунування становить – 70% від усієї кількості, а для тестування – 30%. Тренувальні дані використовуються для тренування моделі. Тестові для перевірки навченої моделі.

На рис 1.5 можемо бачити дані датчику, які неможливо проаналізувати на наявність аномалій програмно використовуючи фіксований мінімум та максимум.

Завдяки цій нейронній мережі можемо працювати з різними графіками навіть з подібними графіками, які зображено на рис. 1.5, де неможливо просто

задати мінімум та максимум, за який не може виходити значення датчику, так як є певна інтервальна послідовність, зменшення інтервалу, та його збільшення.

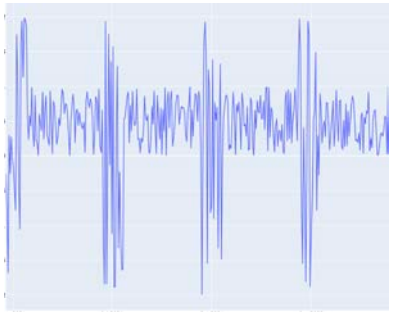

Рис. 1.5. Дані до стандартизації

Прикладом аномалії на подібній послідовності можуть бути приклади, які зображені на рис 1.6.

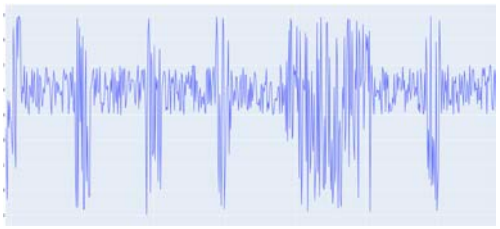

Рис. 1.6. Приклад аномалії

Висновки до розділу 1

В даному розділі були проаналізовані та детально розглянуті аналоги програмного забезпечення. Виявлені явні переваги та недоліки наявного функціоналу. Завдяки розглянутій інформації було сформовано орієнтовний план роботи програмного додатку.

Було складене технічне завдання зі шляхами реалізації та вимогами до функціонального наповнення програми на кожному з етапів. На основі розглянутих аналогів було підкреслено позитивні аспекти, які варто допрацювати та реалізувати у даному програмному додатку. Головним елементом даного програмного додатку буде нейронна мережа, яка автоматично знаходить та показує аномальні значення у різноманітних автоматизованих системах.

РОЗДІЛ 2. РОЗРОБКА ТА ПРОЕКТУВАННЯ АНАЛІЗАТОРА АНОМАЛІЙ ЗА ДОПОМОГОЮ НЕЙРОННОЇ МЕРЕЖІ

2.1. Розробка аналізатора аномалій за допомогою нейронної мережі

Для прогнозної аналітики додатку слід використати прогнозне моделювання за допомогою мови програмування Python.

Для реалізації пропонується використати наступні модулі:

- scikit-learn – це модуль Python для машинного навчання, створений на основі SciPy;
- pandas – програмна бібліотека, написана для мови програмування Python для маніпулювання даними та їхнього аналізу. Вона, зокрема, пропонує структури даних та операції для маніпулювання чисельними таблицями та часовими рядами;
- Numpy – розширення мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих математичних функцій для операцій з цими масивами;
- Plotly – для відображення графіків;
- TensorFlow – для реалізації моделі.

За основу візьмемо нейронну мережу LSTM (від англ. Long short-term memory) – це штучна нейронна мережа, перевагами якої є те що вона уникає проблем вибуху або затухання градієнту, так як вона не змінює ваги шляхів як при звичайному методі оберненого розповсюдження помилки.

Таким чином, ми можемо аналізувати доволі різні графіки даних, і підготувати достатньо універсальний програмний додаток, який буде корисний у багатьох сферах.

2.1.1. Побудова нейронної мережі

Для розуміння роботи нейронної мережі LSTM для початку ознайомимося з основними її компонентами, на рис 2.1 зображена основна структура LSTM.

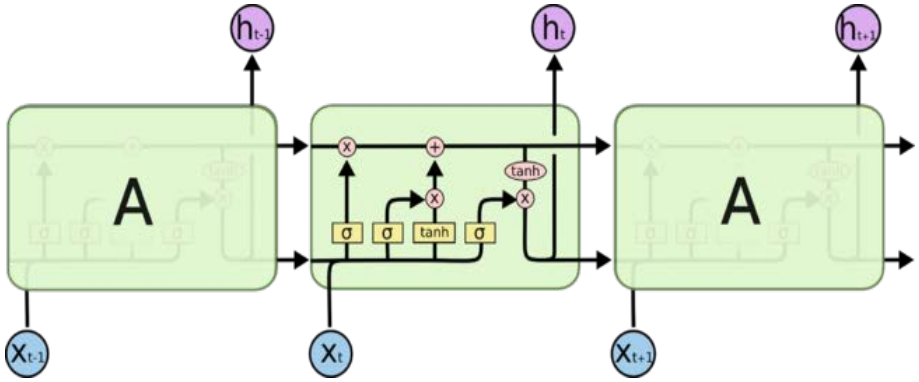

Рис. 2.1. Структура нейронної мережі типу LSTM

Розглянемо основні елементи структури, зображеній на рис. 2.1. під позначенням “σ” мається на увазі сигмоїда.

Сигмоїда – це неперервно диференційована монотонна нелінійна S-подібна функція, яка часто застосовується для «згладжування» значень деякої величини. Визначається за формулою зображеною на рис. 2.2.

$$S(x) = \frac{1}{1 + e^{-x}}$$

Рис. 2.2. Формула сигмоїди

За допомогою сигмоїд легко отримати число в діапазоні від 1 до 0 залежно від числа яке буде на вході як x. Графічний вигляд сигмоїди зображено на рис. 2.3.

Умовне позначення «tanh» означає функцію гіперболічний тангенс, який повертає результат у діапазоні від -1 до 1. Гіперболічна функція тангенса - є гіперболічним аналогом кругової функції тангенсу, яка

використовується в усій тригонометрії. Графічне представлення зображено на рис. 2.4.

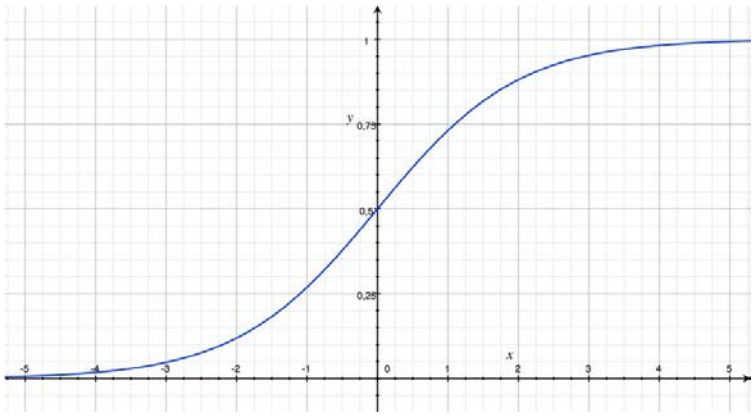

Рис. 2.3. Графічне представлення сигмоїди

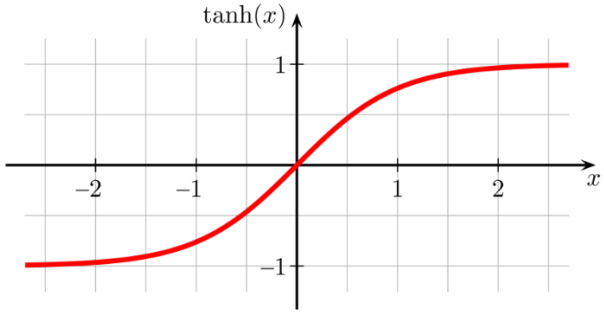

Рис. 2.4. Графічне представлення гіперболічного тангенсу

Визначається за формулою зображеною на рис 2.5.

$$f(x) = \frac{\left(e^{x} - e^{-x}\right)}{\left(e^{x} + e^{-x}\right)}$$

Рис. 2.5. Формула гіперболічного тангенсу

Умовне позначення «+» та «×» позначають дії додавання та множення відповідно.

Проаналізувавши символи на загальній схемі, перейдемо до детального огляду на роботу нейронної мережі LSTM.

На рис. 2.6 зображено перший етап роботи нейронної мережі. Також перший шар можна назвати шаром забування (англ. Forget gate layer). Саме в цьому шарі визначається, яку інформацію можна забути, а яку залишити.

Значення попереднього виходу h_{t-1} та поточного входу x_t пропускаються через сигмоїдальний шар. Отримані значення перебувають у діапазоні [0; 1]. Значення, які ближче до 0, будуть забуті, а до 1 залишені.

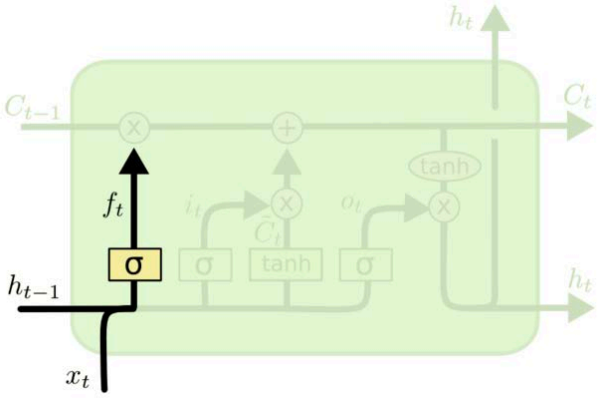

Рис. 2.6. Перший етап роботи LSTM

На рис. 2.7 зображено формулу даного етапу.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Рис. 2.7. Формула першого етапу роботи LSTM

Далі вирішується, яка нова інформація зберігатиметься у стані ячейки. Цей етап складається із двох частин. Спочатку сигмоїдальний шар під назвою «шар вхідного фільтра» (англ. input layer gate) визначає, які значення слід оновити. Потім шар гіперболічного тангенсу будує вектор нових значень-кандидатів, які можна додати до ячейки. На рис. 2.8 схематично зображено роботу другого етапу.

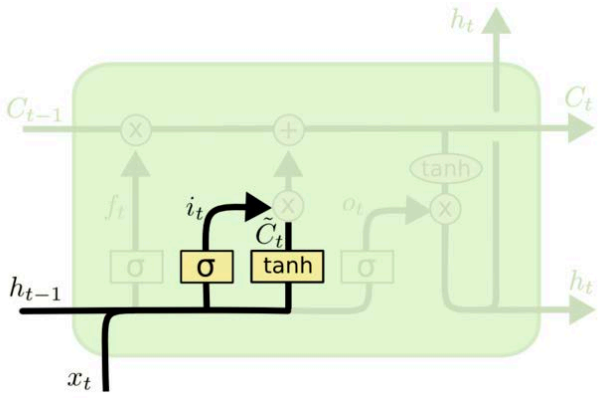

Рис. 2.8. Другий етап роботи LSTM

На рис. 2.8 зображено формулу даного етапу.

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Рис. 2.9. Формула другого етапу роботи LSTM

На третьому етапі для заміни старого стану ячейки C_{t-1} на новий стан C_t. Необхідно помножити старий стан на f_t, забуваючи про те, що вирішили забути раніше. Потім додаємо вираз $i_t * \tilde{C}_t$. Це нові значення-кандидати, помножені на t – на скільки оновити кожне із значень стану. На рис. 2.10 схематично зображено роботу третього етапу.

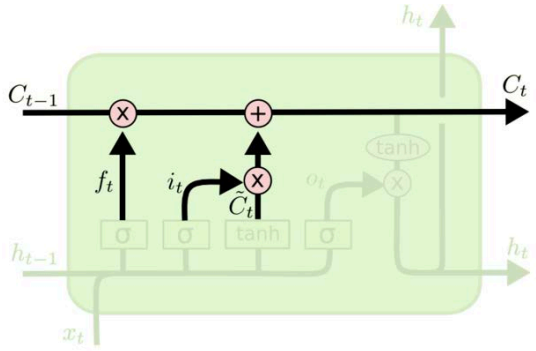

Рис. 2.10. Третій етап роботи LSTM

На рис. 2.11 зображено формулу даного етапу.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Рис. 2.11. Формула третього етапу роботи LSTM

На останньому етапі визначається те, яку інформацію буде отримано на виході. Вихідні дані будуть засновані на нашому стані ячейки, до них будуть використані деякі фільтри.

Спочатку значення попереднього виходу h_{t-1} і поточного входу x_t пропускаються через сигмоїдальний шар, який вирішує, яка інформація стану ячейки буде виведена. Потім значення стану ячейки проходять через шар гіперболічного тангенсу, щоб отримати на виході значення з діапазону від -1 до 1, і перемножуються з вихідними значеннями сигмоїдального шару,

що дозволяє виводити тільки необхідну інформацію. На рис. 2.12 схематично зображено роботу останнього четвертого етапу.

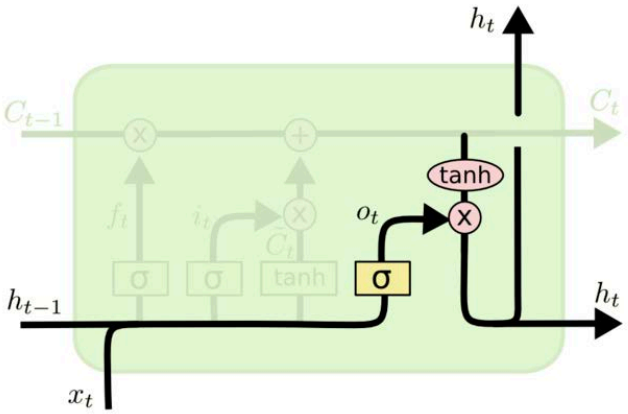

Рис. 2.12. Четвертий етап роботи LSTM

На рис. 2.13 зображено формулу даного етапу.

$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Рис. 2.13. Формула четвертого етапу роботи LSTM

Отримані таким чином h_t і C_t передаються далі ланцюжком. Для мінімізації загальної помилки можна було б використати ітеративний градієнтний спуск із метод зворотного поширення помилки але головною проблемою градієнтного спуску для стандартних рекурентних нейронних мереж є те, що градієнти помилок зменшуються з експоненційною швидкістю зі збільшенням тимчасової затримки між важливими подіями, що було виявлено в 1991 [11,12]. З LSTM-блоками, тим не менш, коли величини помилки поширюються у зворотному напрямку від вихідного шару, помилка виявляється замкнена в пам'яті блоку. Таким чином, регулярне зворотне поширення помилки є ефективним для тренування LSTM-блоку для запам'ятовування значень на дуже тривалі часові проміжки.

2.1.2. Створення датасету

Для реальної перевірки роботи алгоритму бажано мати реальне обладнання, яке повертало б певні значення, але так як обладнання може бути різним, було вирішено проаналізувати інтернет ресурси, та виявити які типи графіків можуть повертатися з автоматизованих систем.

Проаналізувавши інтернет ресурси було вирішено для прикладів генерувати дані у певному діапазоні, періодичні дані з повторенням коливань та дані синусоїди. Дані у певному діапазоні зображені на рис 2.14.

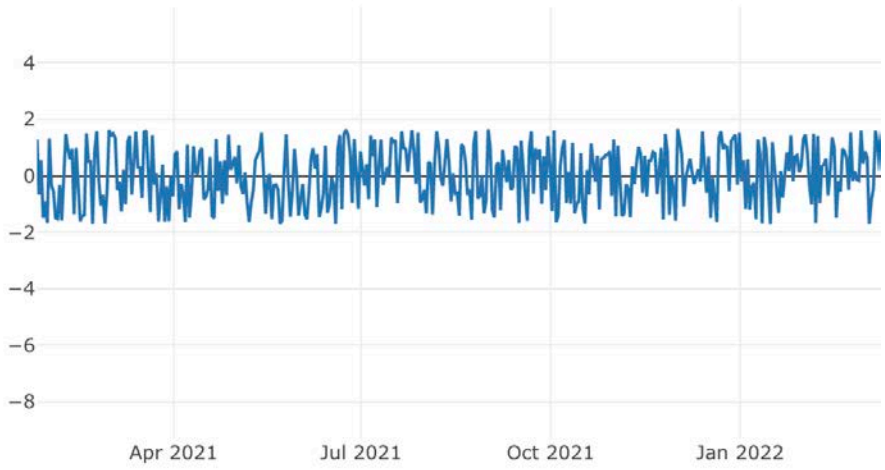

Рис. 2.14. Приклад датасету в певному діапазоні

Приклад періодичних даних зображені на рис 2.15.

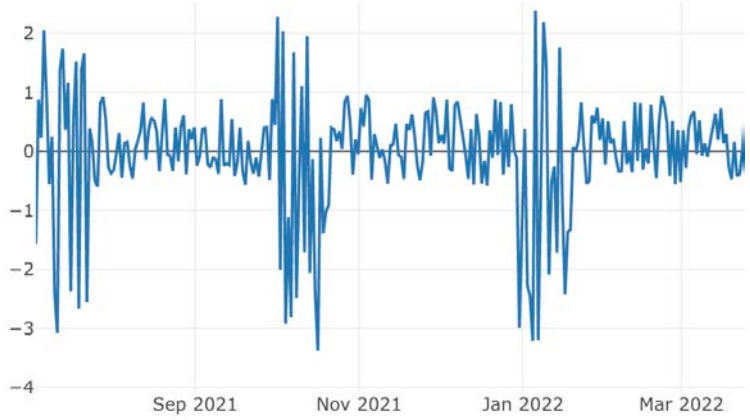

Рис. 2.15. Приклад періодичного датасету

Приклад даних синусоїди зображені на рис 2.16.

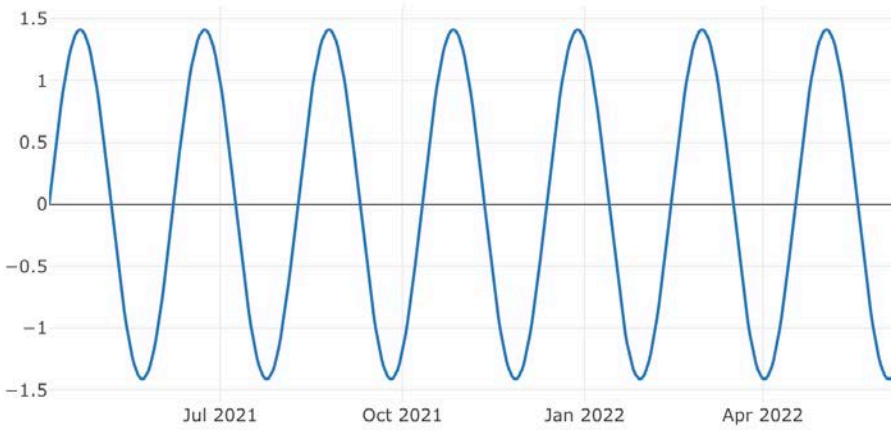

Рис. 2.16. Приклад синусоїдного датасету

На цих тестових даних також було згенеровано аномальні значення, прикладами подібних аномалій може бути ситуація, яка зображена на рис 2.17, де аномалії показані помаранчевими точками.

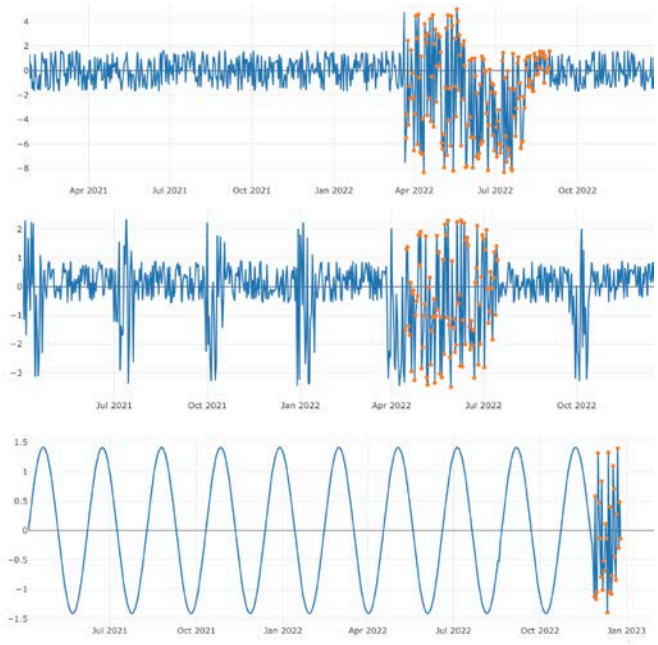

Рис. 2.17. Приклад аномальних значень

2.2. Підготовка до аналізу даних

Для роботи з алгоритмами штучного інтелекту потрібно підготувати дані у відповідному форматі. Після генерації датасету та збереження даних у csv файл варто вибрати тільки потрібні рядки та виконати стандартизацію даних (англ. standartisation).

Стандартизація даних – це процес перетворення даних у загальний формат, щоб аналізатори могли їх обробляти й аналізувати. Більшість організацій використовують дані з кількох джерел; це може включати локальні сховища даних, хмарні сховища та різні бази даних. Однак дані з різних джерел можуть бути проблематичними, якщо вони не є однорідними, що призводить до труднощів у подальшому (наприклад, коли ви використовуєте ці дані для створення інформаційних панелей, візуалізацій тощо).

Стандартизація даних має вирішальне значення з багатьох причин. Перш за все, це допомагає встановити чіткі, узгоджено визначені елементи та атрибути, забезпечуючи повний каталог ваших даних. Незалежно від того, яку статистику намагаємося отримати або проблеми, які намагаємося вирішити, правильне розуміння даних є важливою відправною точкою.

Для цього потрібно перетворити ці дані в єдиний формат із логічними та послідовними визначеннями. Ці визначення формуватимуть метадані – мітки, які ідентифікують різні аспекти даних. Це основа процесу стандартизації даних.

З точки зору точності, стандартизація способу позначення даних покращить доступ до найбільш актуальних частин інформації. Це спростить аналітику та звітність. Розраховується за формулою зображеною на рис. 2.18

$$Z = \frac{x - \mu}{\sigma}$$

Рис. 2.18. Формула обчислення стандартизації

У вище згаданій формулі x – точка з датасету, μ – є середнім арифметичним датасету, σ – є стандартним відхиленням датасету.

Формула розрахунку стандартного відхилення зображено на рис. 2.18, де $\bar{x}$ – є середнім арифметичним датасету, n – є кількістю значень датаету, x_i – є і-тим значенням з датасету.

$$\sqrt{\frac{\sum(x_i-\bar{x})^2}{n-1}}$$

Рис. 2.19. Формула обчислення стандартного відхилення

Після стандартизації набору даних, їх варто розділити на тренувальні та тестові вибірки. Зазвичай вибірка для трунування становить – 70% від усієї кількості, а для тестування – 30%. Тренувальні дані використовуються для тренування моделі. Тестові для перевірки навченої моделі.

2.3. Проектування додатку

Додаток виявлення аномалій у автоматизованих системах має бути зручним у використанні для користувачів.

Користувачі, залежно від ролі, мають доступ до наступних функцій:

1. Аутентифікація користувачів – вхід в систему і задання відповідних ролей користувача. Для системи важливо надати правильний набір функцій для певного користувача.
2. Авторизація користувачів – надання доступу до певного функціоналу додатку залежно від ролі.
3. Створення, зміна та видалення інформації станції.
4. Створення, зміна та видалення інформації обладнання.
5. Створення, зміна та видалення інформації датчику.
6. Створення, зміна та видалення типів обладнання.
7. Створення, зміна та видалення типів датчиків.

8. Перегляд даних про станціїї, обладнання, датчики, типи обладнання та типи датчиків.

9. Перегляд аномалій у значеннях з датчиків автоматизованих систем.

Виділимо дві ролі Менеджер та Оператор станції. Їх функціонал зображено на рис. 2.20.

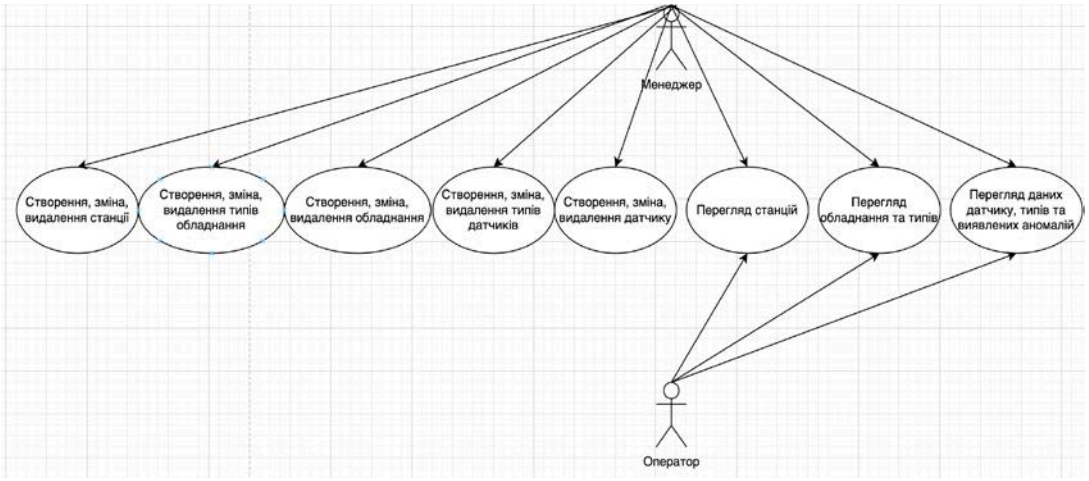

Рис. 2.20. Доступний функціонал залежно від ролі користувача

Функціональні вимоги:

1. Веб-додаток має функціонувати без збоїв та неочікуваних помилок.

2. Користувач потрібен перебувати онлайн, так як зміна будь-якої інформації є запитом до серверу.

Нефункціональні вимоги:

1. Сприйняття:

– анімації не повинні перевищувати 0.3 секунди;

– час відповіді системи для звичайних запитів не повен перевищувати 1 секунди, а для складних запитів – 2 секунди;

– інтерфейс має бути зрозумілим та відповідати вимогам UI/UX правил.

2. Можливість експлуатації:

– масштабування – після нової версії весь попередній функціонал має працювати коректно.

Проаналізувавши можливості користувачів створимо діаграму класів, зображено на рис. 2.21.

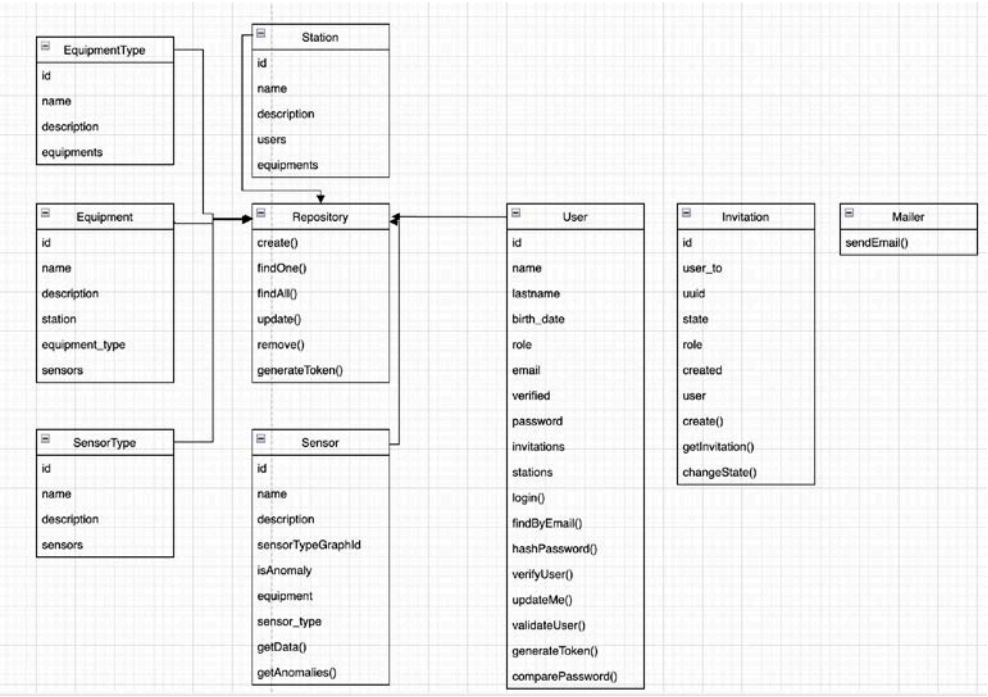

Рис. 2.21. Діаграма класів

Після діаграми класів розглянемо схему бази даних, зображену на рис. 2.22.

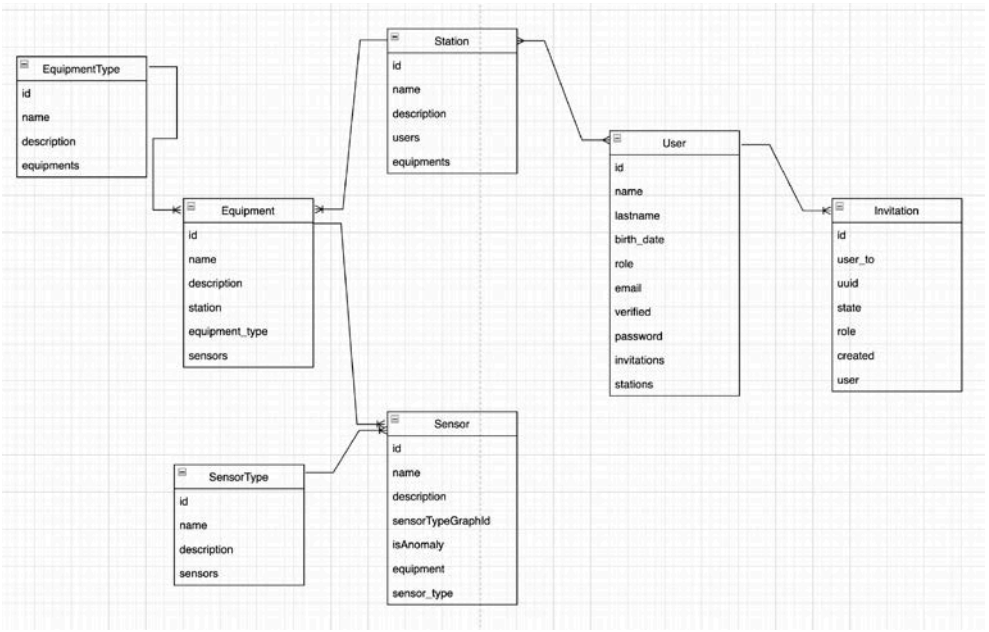

Рис. 2.22. Схема бази даних

Для простого опису роботи додатку створимо діаграму розгортання. На ній зобразимо браузер користувача, де буде розгорнута клієнтська частина, для сереверу створимо окремий елемент де буде розташований сам сервер та створена бізнес логіка, також для збереження інформації реалізуємо елемент бази даних де буде розгорнута postgreSQL, зображено на рисунку 2.22.

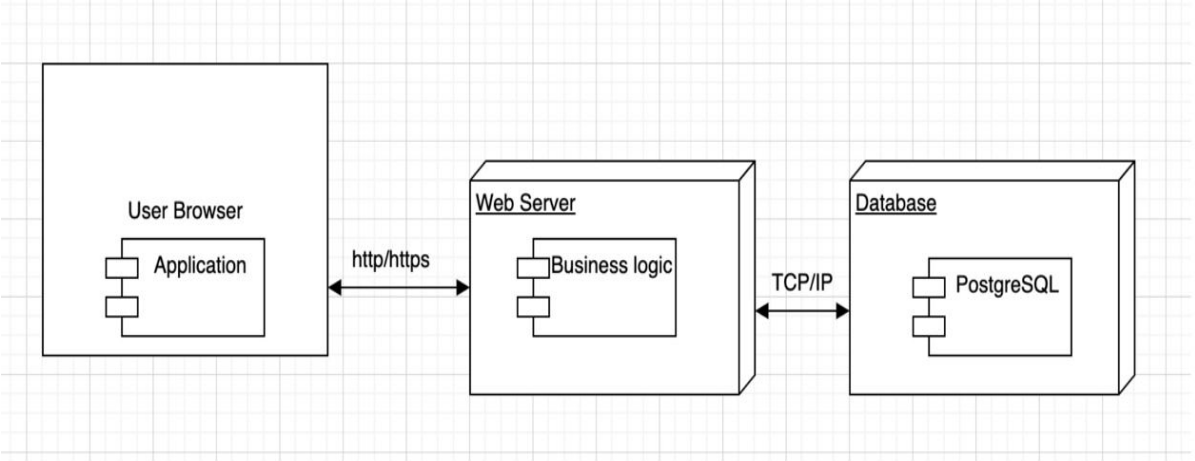

Рис. 2.22. Діаграма розгортання.

Висновки до розділу 2

В даному розділі була детально розглянута робота нейронної мережі LSTM. Було опрацьовано та досліджено методи чистки та оптимізації даних перед роботою нейронної мережі.

Спроектовано діаграму класів та основних компонентів, а також була спроектована база даних для подальшого зберігання інформації стосовно користувачів, станцій, обладнання, датчиків.

Реалізовано проектування програмного додатку для виявлення аномалій у автоматизованих системах. Спочатку система отримує дані, обробляє їх, чистить та відбирає тільки потрібні стовпці інформації. Після проходить первинну обробку шляхом стандартизації даних. Згодом проходить навчання та опрацьовується нейронною мережею для відображення результатів.

РОЗДІЛ 3. ОПИС ПРОГРАМНОГО ПРОДУКТУ

3.1. Графічний інтерфейс

Після завантаження ресурсу користувач бачить стартову сторінку з коротким описом продукту та кнопкою «Почати», що направить користувача на сторінку аутентифікації. Головна сторінка зображена на рис. 3.1.

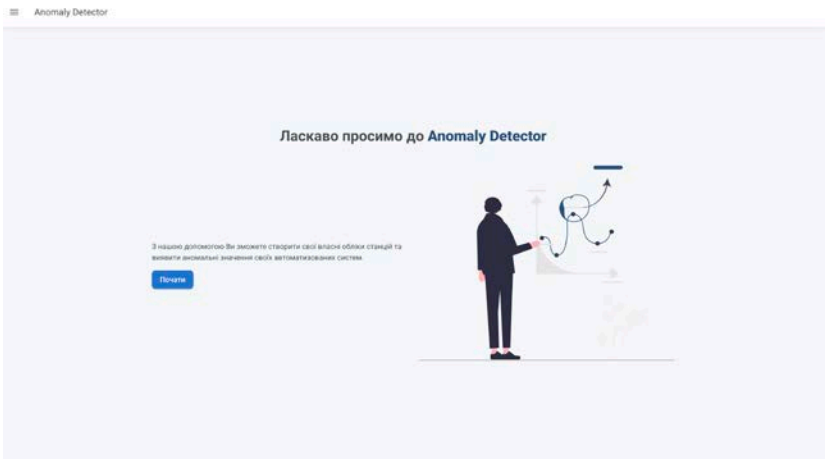

Рис. 3.1. Головна сторінка

Після перенаправлення на сторінку аутентифікації користувач бачить базове вікно для вибору реєстрації або ж логіну користувача. Зображено на рис. 3.2.

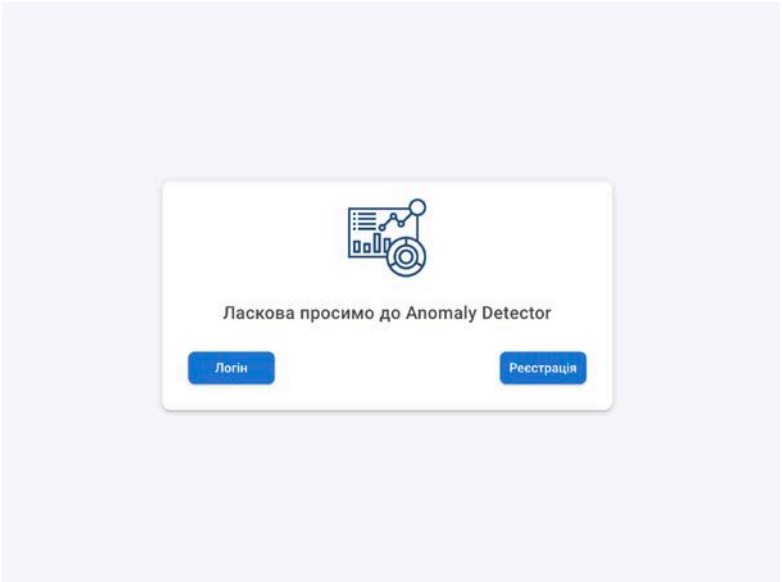

Рис. 3.2. Сторінка аутентифікації

Якщо користувач обирає реєстрацію, відкривається форма реєстрації користувача, де він може ввести особисті дані. Зображено на рис. 3.3.

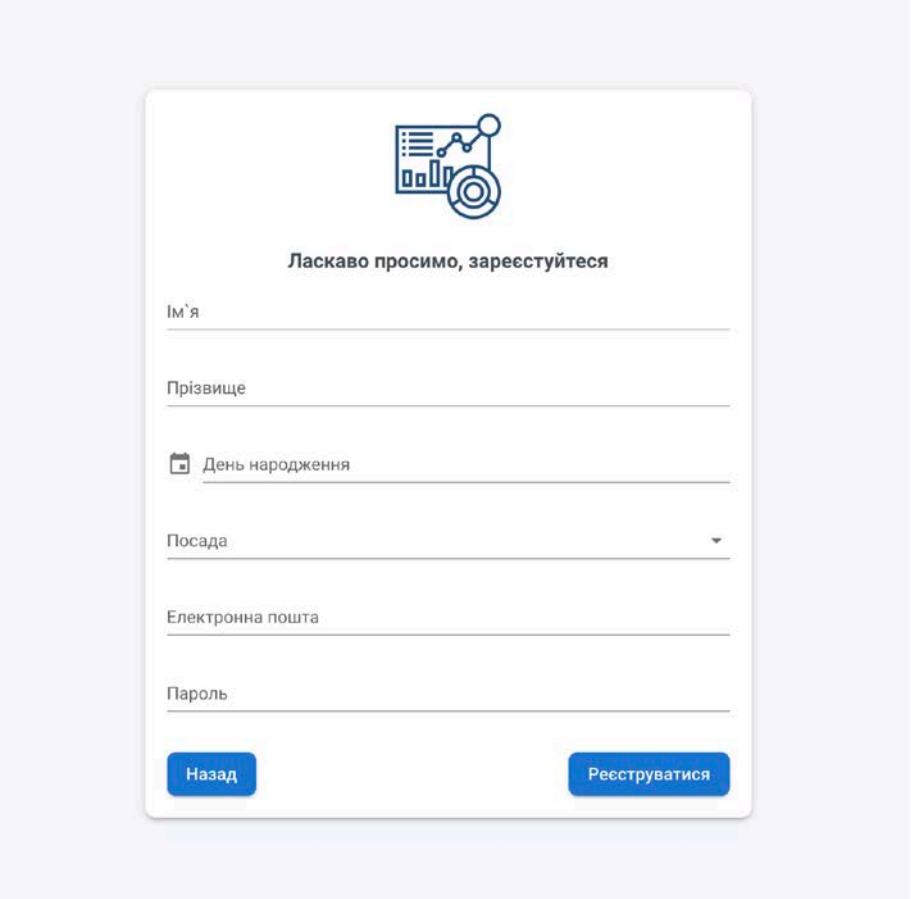

Рис. 3.3. Форма реєстрації

Форма логіну зображена на рис. 3.4. Також для кожної форми програмного додатку забезпечена валідація вводу, зображено на рис. 3.5.

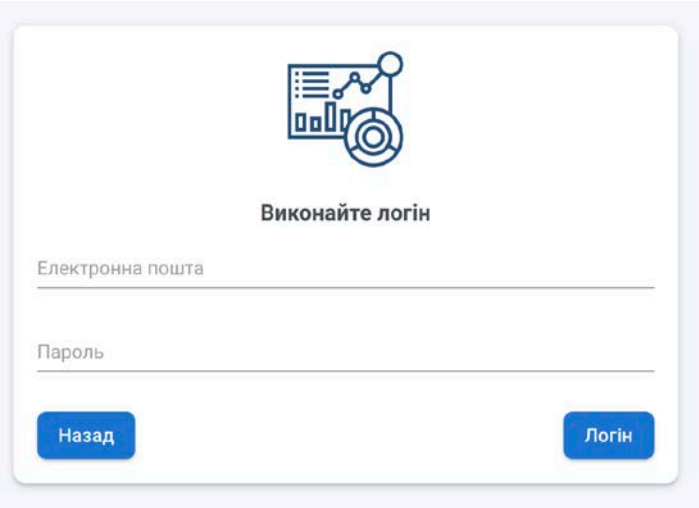

Рис. 3.4. Форма логіну

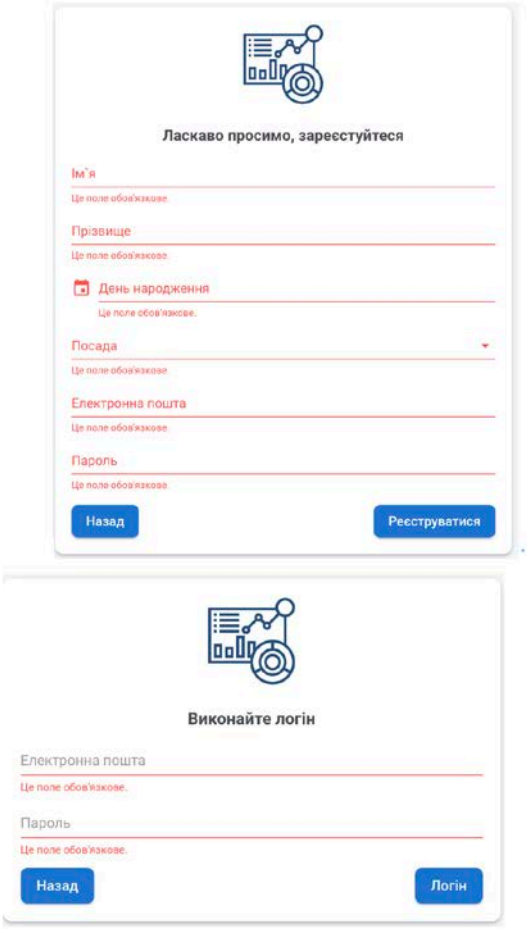

Рис. 3.5. Валідація форм аутентифікації

Після реєстрації користувач має підтвердити електронну пошту, за адресою, яку він вводив при реєстрації користувача у програмному додатку. Після підтвердження користувач отримає сповіщення про успішно підтверджену електронну пошту у додатку, зображено на рис.3.6.

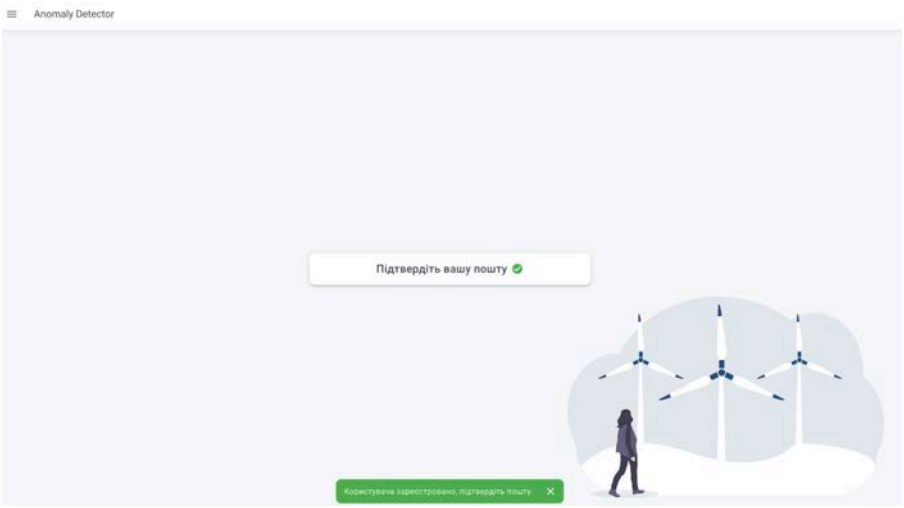

Рис. 3.6. Сторінка підтвердження користувача за поштою

Після успішної реєстрації користувач може ознайомитися з основним функціоналом програмного додатку. Перейшовши у свій профіль, користувач може переглянути особисту інформацію, ознайомитись з базовим функціоналом програми та змінити свою особисту інформацію, натиснувши на відповідну кнопку та ввівши нові дані для зміни. Зображено на рис.3.7.

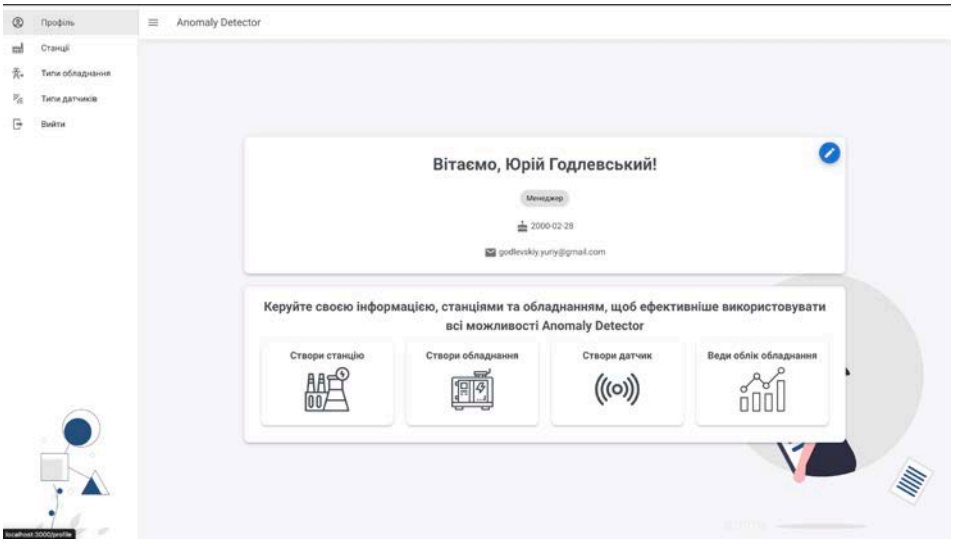

Рис. 3.7. Профіль користувача

Форма зміни особистої інформації відображена на рис. 3.8.

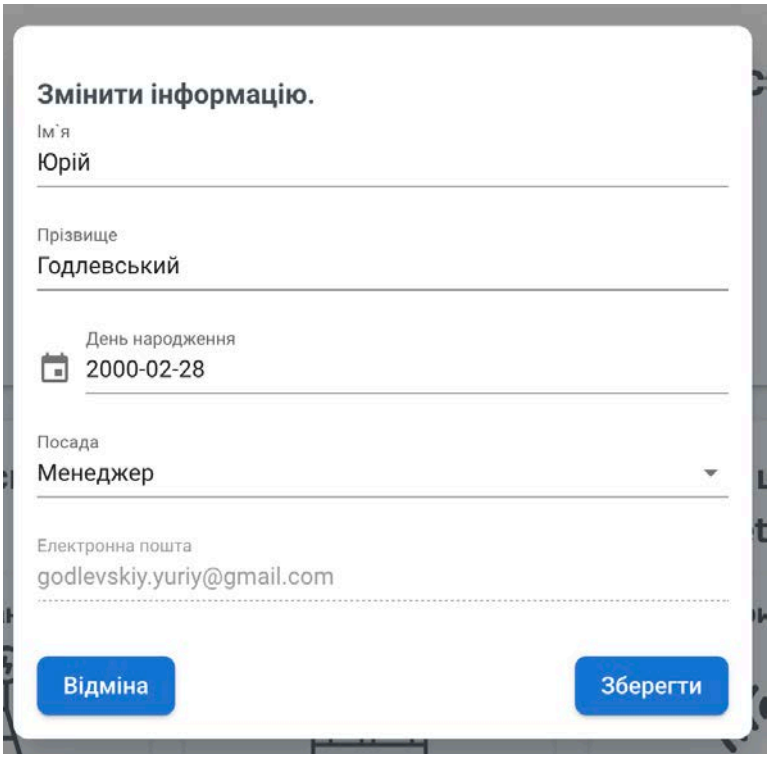

Рис. 3.8. Профіль користувача

На сторінці станцій користувач з правами менеджера може створити станції, змінити, або ж видалити якусь зі станцій. Сторінка станцій зображена на рис. 3.9.

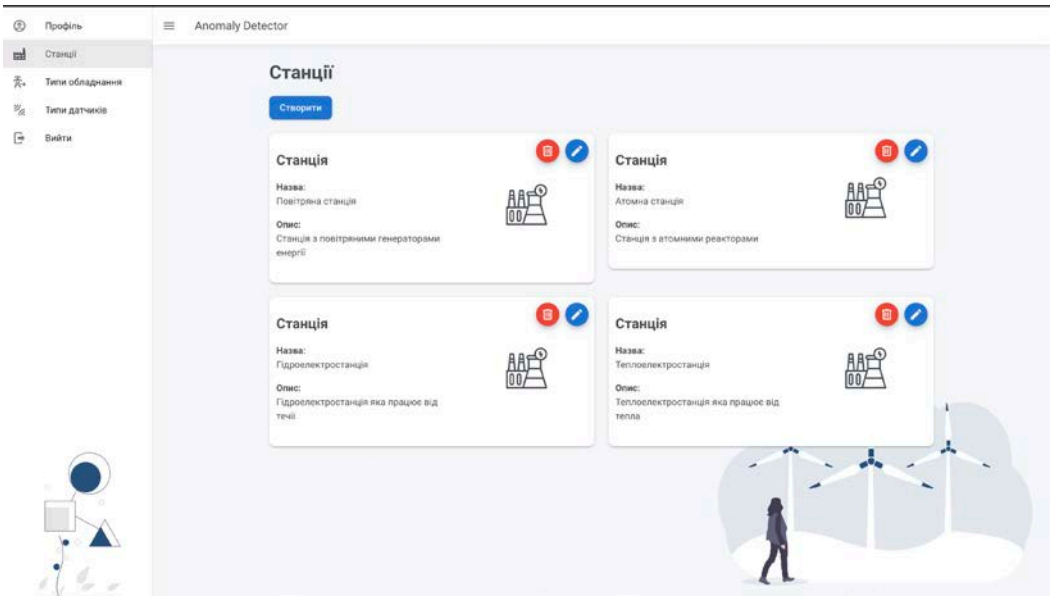

Рис. 3.9. Сторінка станцій

При видаленні якоїсь зі станцій з’являється діалогове вікно видалення, зображено на рис. 3.10.

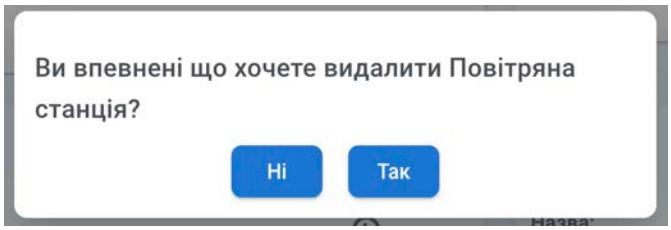

Рис. 3.10. Діалогове вікно видалення

Форма для створення та зміни інформації станції зображено на рис. 3.11.

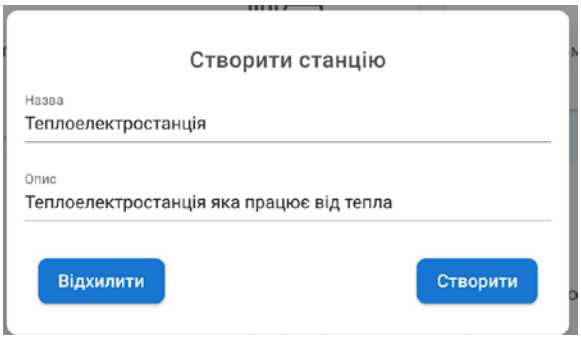

Рис. 3.11. Форма створення станції

Після створення типів обладнання, та типів датчиків, користувач може почати створювати обладнання та датчики, використовуючи заздалегідь підготовлені типи. Створення обладнання з підготовленим списком типів зображено на рис. 3.12.

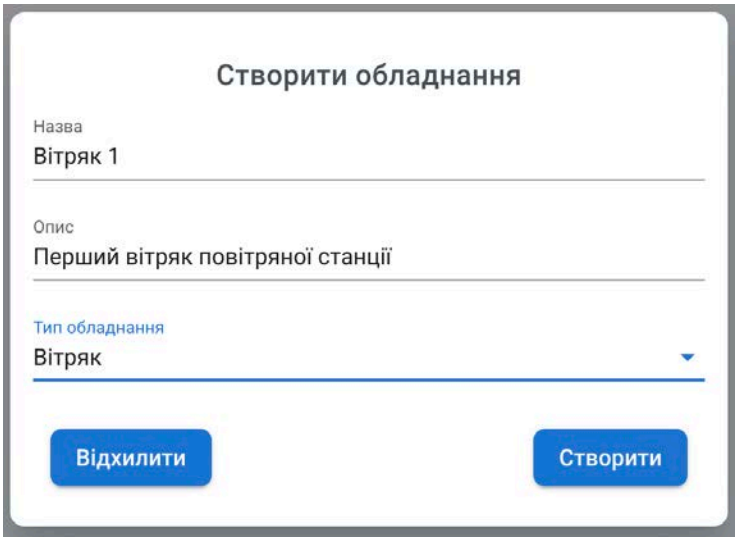

Рис. 3.12. Форма створення обладнання

Користувач може вибрати відповідний потрібний тип з випадаючого меню, зображено на рис. 3.13. Логіка роботи створення датчиків є аналогічною за вийнятком вибору типу графіку та присутність чи відсутність аномалій, так як датасет генерується, а не береться з реальної автоматизованої системи, так як доступу до подібної на стадії розробки немає.

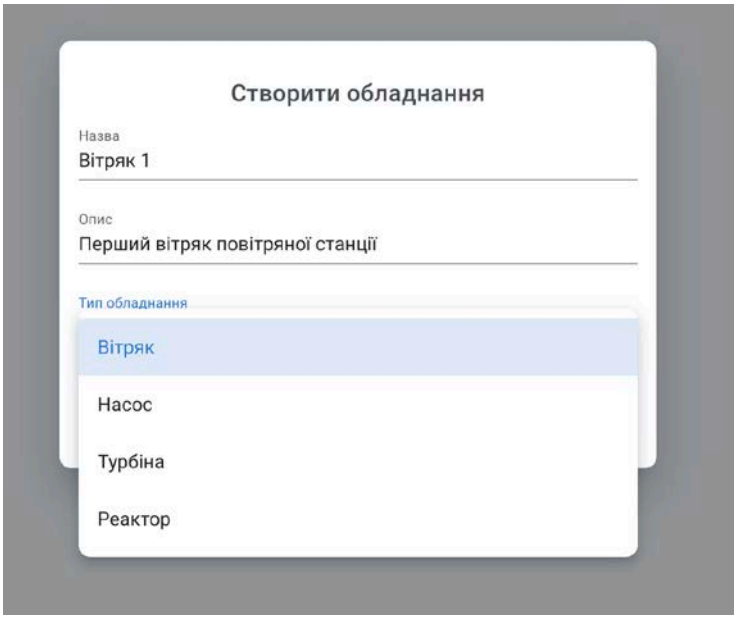

Рис. 3.13. Випадаюче меню типів обладнання

Форма створення датчику зображена на рис. 3.14

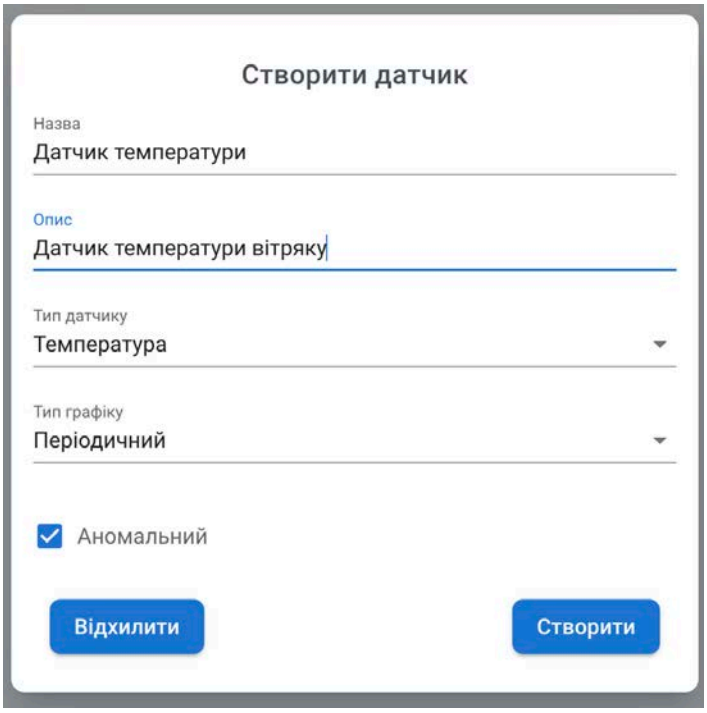

Рис. 3.14. Форма створення датчику

Серед типів графіку можна обрати періодичний, з певним періодом повторення діапазонів даних, звичайний, де є певний діапазон і значення, які не виходять за нього, та синусоїда. Вибрати тип можна також у випадаючому меню, яке зображено на рис. 3.15.

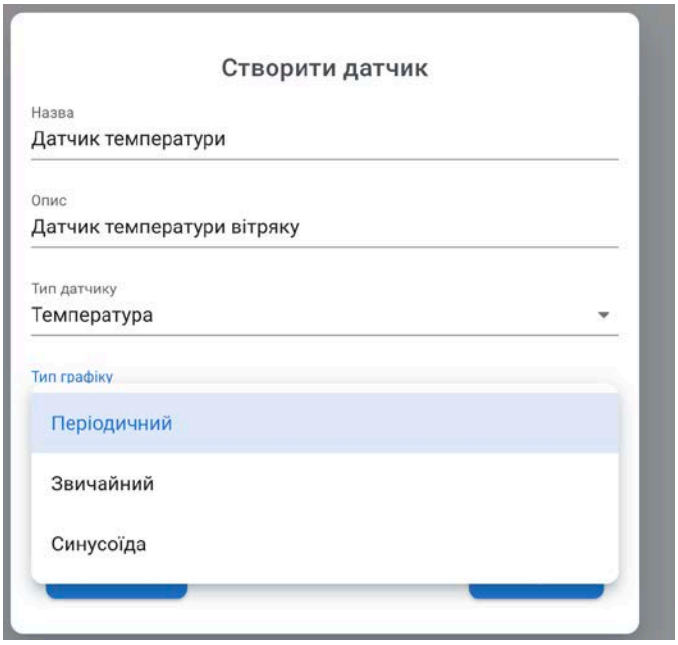

Рис. 3.15. Випадаюче меню типу графіку

На сторінці обладнання певної станції можна побачити створене обладнання та тип кожного з них на відповідній карті. Видалення та редагування інформації працює за таким самим алгоритмом, як було згадано вище у описі роботи користувача зі станціями. Зображено на рис. 3.16

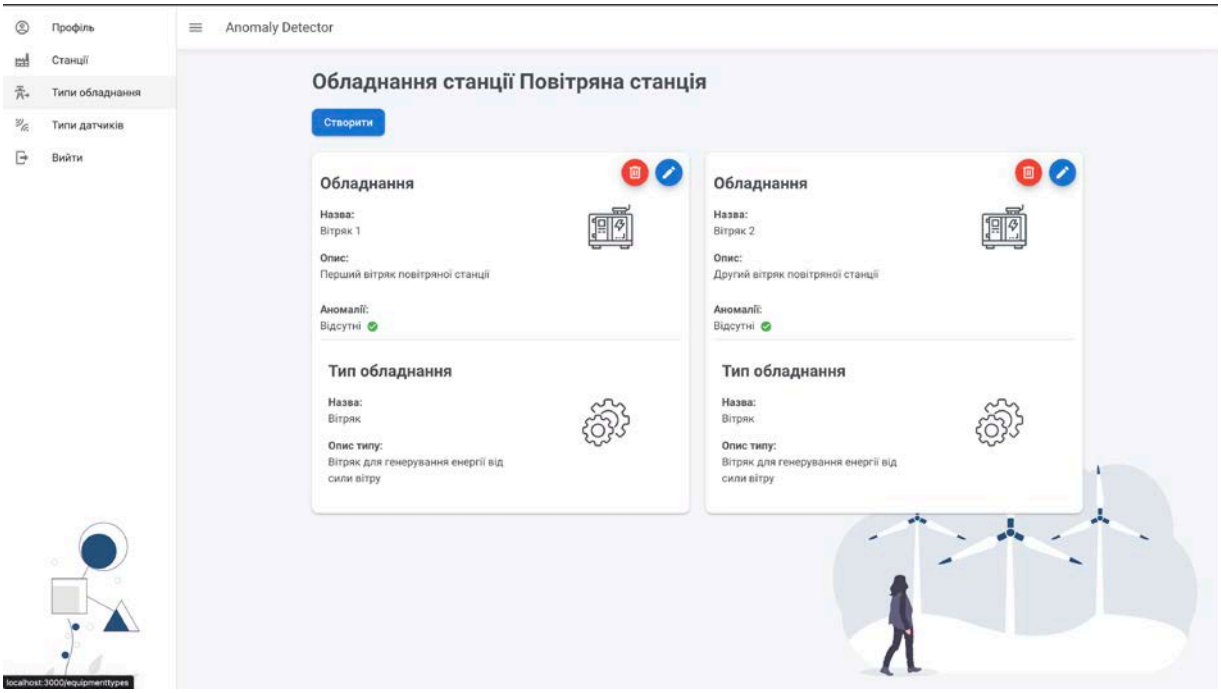

Рис. 3.16. Сторінка обладнання

Для зміни обладнання варто натиснути на кнопку редагування і з’явиться вікно редагування обладнання, зображено на рис. 3.17.

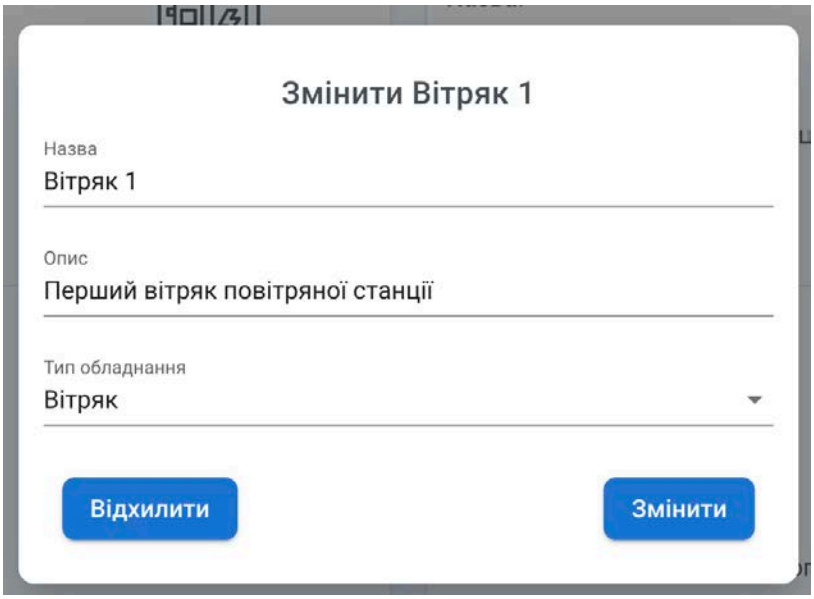

Рис. 3.17. Вікно редагування обладнання

3.2. Налаштування параметрів створення автоматизованих систем

Перед тим як користувач створить обладнання станцій та датчики обладнання, потрібно створити типи обладнання, що фактично є певною автоматизованою системою даної станції.

Так як різні обладнання можуть мати схожі властивості у різних станцій. Саме ці спільні особливості варто винести у тип обладнання. Такою самою логікою потрібно користуватися при створені типів для сенсорів, так як, наприклад, датчики температури можуть бути встановлені на різному обладнанні і не має сенсу створювати кожного разу одну і ту саму інформацію для різних типів обладнань. Сторінка типів обладнання зображена на рис. 3.18.

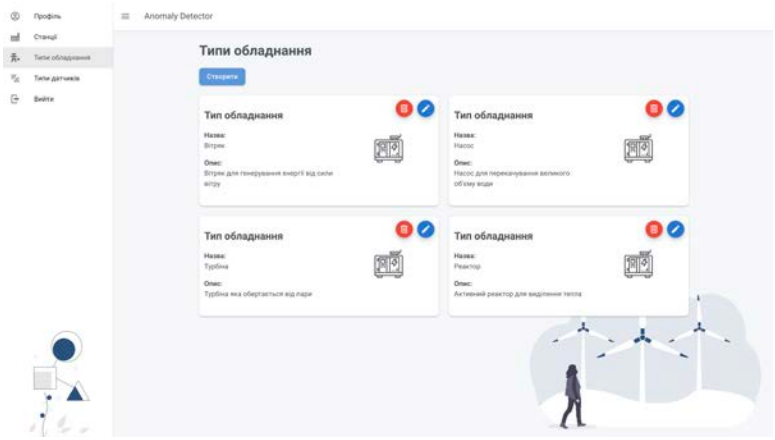

Рис. 3.18. Форма створення станції

Подібною до попередньої є і сторінка типів датчиків, зображено на рис 3.19.

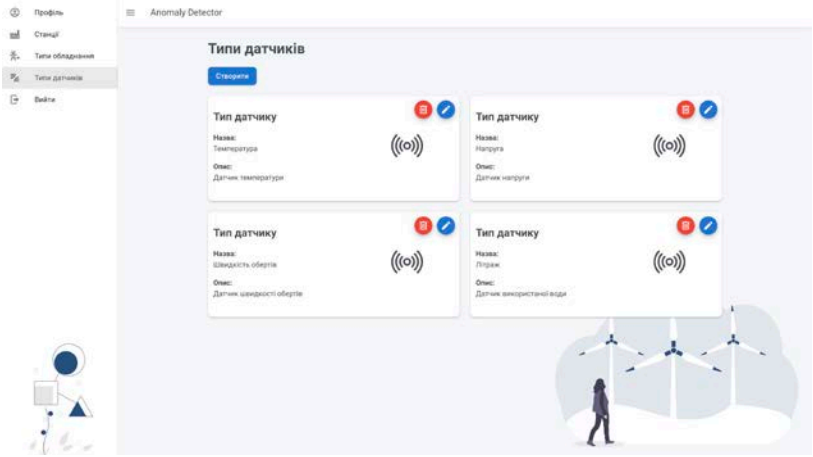

Рис. 3.19. Форма створення станції

3.3. Відображення результатів аналізу

Якщо система не виявила аномалій, то користувач буде проінформований написом що аномалії в конкретному обладнанні відсутні по усім датчикам, зображено на рис. 3.20.

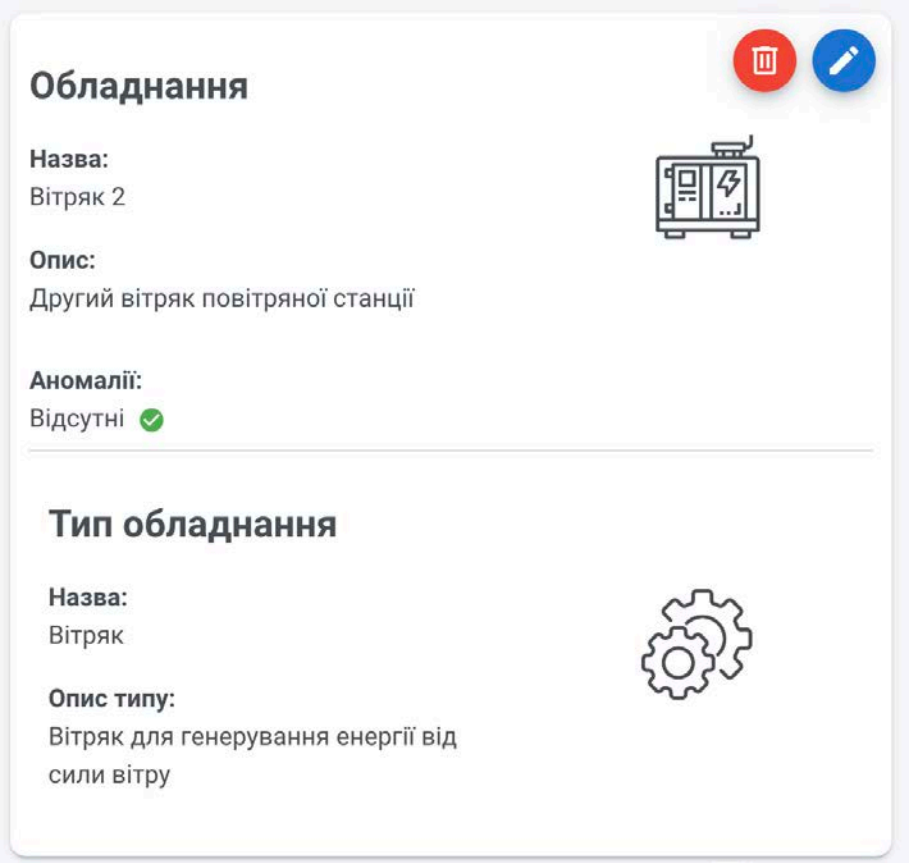

Рис. 3.20. Обладнання з відсутніми аномаліями

Якщо ж аномалії були виявлені, то статус зміниться на «Присутні», зображено на рис. 3.21.

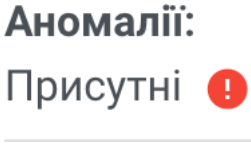

Рис. 3.21. Статус обладнання з присутніми аномаліями

Для детального вивчення де саме відбулась аномалія, користувачу потрібно клікнути на обладнання, яке йому потрібно дослідити, та перейшовши на сторінку усіх датчиків даного обладнання проаналізувати датчики, на наявність статусу «Присутні». Зображено на рис. 3.22.

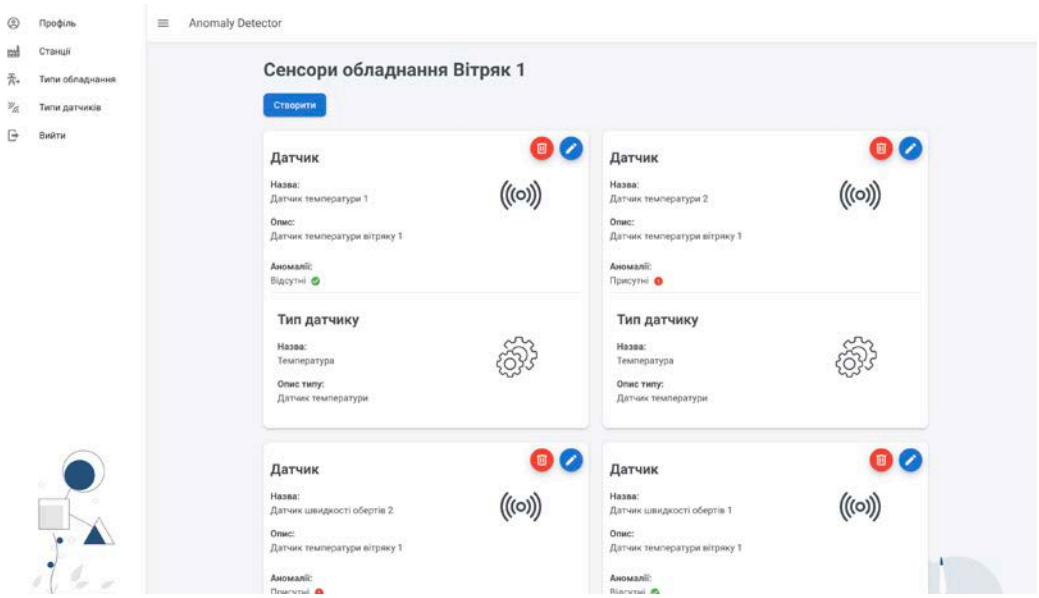

Рис. 3.22. Сторінка датчиків даного обладнання з присутніми аномаліями

Після цього користувач може перейти на датчик, який йому потрібно дослідити, та проаналізувати результати виявлення аномалій у даній автоматизованій системі. Графік подібних результатів зображено на рис. 3.23.

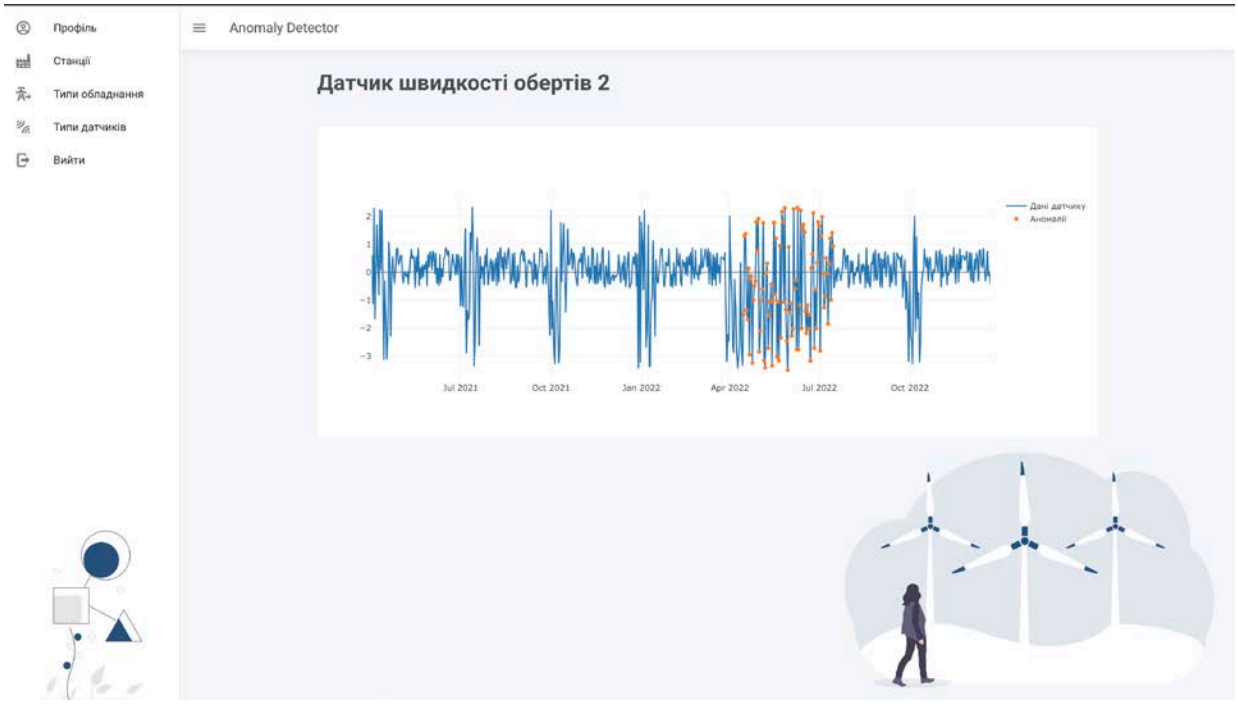

Рис. 3.23. Сторінка датчику даного обладнання з присутніми аномаліями

Якщо аномалії були виявлені, то вони позначаються помаранчевими точками, як зображено на рис. 3.23. Якщо ж аномалії відсутні, то графік буде складатися тільки з показників поточного датчику даного обладнання, зображено на рис. 3.24.

Рис. 3.24. Сторінка датчику даного обладнання з відсутніми аномаліями

Таким чином, програма маже аналізувати досить різні графіки, приклад зображено на рис. 3.25.

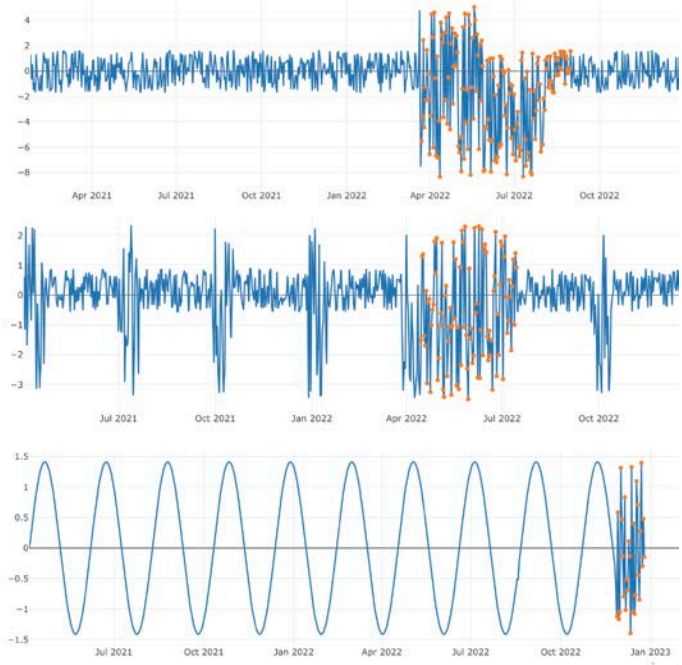

Рис. 3.25. Приклади проаналізованих аномальних графіків

Висновки до розділу 3

В даному розділі наведений опис способів взаємодії користувача з програмою. Описані способи налаштування автоматизованих систем. Розглянуто графічний інтерфейс програми, який реалізований з дотриманням UI/UX практик. Наведено перелік графіків з якими програма може працювати, та показано її ефективність. Також проведене тестування програмного забезпечення.

ВИСНОВКИ

В даній роботі було реалізовано програмний додаток, який є зручним у використанні для користувача. Що є комфортним для ведення обліку різних автоматизованих систем станцій та реалізовано механізм автоматичного виявлення аномалій в них. Було підготовлено технічне завдання та обрано коректні інструменти реалізації програми.

Головну роль у виявленні аномаліє відіграє нейронна мережа LSTM. В даній роботі наводиться опис процесу підготовки даних для її використання, її створення та навчання. Для тренування нейронної мережі був створений власний генератор датасету після аналізу різноманітних інтернет ресурсів. Описано процес вибору конкретної архітектури нейронної мережі. Також було реалізовано ряд алгоритмів для попередньої підготовки даних.

Після створення програми було проведене тестування. Воно полягало в перевірці якості програмного додатку та правильності його роботи. Отримані результати відповідають поставленим до програми вимогам.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Yurii O. Hodlevskyi, Tetiana A. Vakaliuk, Oleksii V. Chyzhmotria, Olena H. Chyzhmotria and Oleh V. Vlasenko. Finding Anomalies in the Operation of Automated Control Systems Using Machine Learning. Proceedings of the 4th International Workshop on Intelligent Information Technologies & Systems of Information Security, Khmelnytskyi, Ukraine, March 22–24, 2023. Edited by Tetiana Hovorushchenko, Oleg Savenko, Peter T. Popov, Sergii Lysenko. CEUR Workshop Proceedings (CEUR-WS.org, ISSN 1613-0073). Vol. 3373. Pp. 681-698. https://ceur-ws.org/Vol-3373/paper47.pdf
2. Preeti Rajni Bala, Ram Pal Singh. A dual-stage advanced deep learning algorithm for long-term and long-sequence prediction for multivariate financial time series. Applied Soft Computing. Volume 126, 2022, 109317. https://doi.org/10.1016/j.asoc.2022.109317.
3. Sheng Xiang, Yi Qin, Caichao Zhu, Yangyang Wang, Haizhou Chen. LSTM networks based on attention ordered neurons for gear remaining life prediction. ISA Transactions. Volume 106, 2020. Pp. 343-354. https://doi.org/10.1016/j.isatra.2020.06.023.
4. Sheng Xiang, Yi Qin, Caichao Zhu, Yangyang Wang, Haizhou Chen. Long short-term memory neural network with weight amplification and its application into gear remaining useful life prediction. Engineering Applications of Artificial Intelligence.Volume 91, 2020, 103587, https://doi.org/10.1016/j.engappai.2020.103587.
5. Anuraganand Sharma. Guided Stochastic Gradient Descent Algorithm for inconsistent datasets. Applied Soft Computing. Volume 73, 2018, pp. 1068-1080. https://doi.org/10.1016/j.asoc.2018.09.038.
6. Farajtabar M., Azizan N., Mott A., Li A. Orthogonal gradient descent for continual learning. Proceedings of the 23rdInternational Conference on

Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108. https://core.ac.uk/download/pdf/345075797.pdf

7. Azizan, Navid & Hassibi, Babak. (2018). Stochastic Gradient/Mirror Descent: Minimax Optimality and Implicit Regularization. ICLR 2019. https://openreview.net/pdf?id=HJf9ZhC9FX
8. Simon Du, Jason Lee, Haochuan Li, Liwei Wang, Xiyu Zhai. Gradient Descent Finds Global Minima of Deep Neural Networks. Proceedings of the 36th International Conference on Machine Learning, PMLR 97:1675-1685, 2019. http://proceedings.mlr.press/v97/du19c.html
9. Lee, J.D., Simchowitz, M., Jordan, M.I. & Recht, B.. (2016). Gradient Descent Only Converges to Minimizers. 29th Annual Conference on Learning Theory, in Proceedings of Machine Learning Research. 49: 1246-1257. https://proceedings.mlr.press/v49/lee16.html.
10. Soukup D., Cejka T., Hynek K. (2020) Behavior Anomaly Detection in IoT Networks. Lecture Notes on Data Engineering and Communications Technologies, 49, pp. 465 - 473. DOI: 10.1007/978-3-030-43192-1_53
11. Grcić, M., Bevandić, P., Šegvić, S. (2022). DenseHybrid: Hybrid Anomaly Detection for Dense Open-Set Recognition. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds) Computer Vision – ECCV 2022. ECCV 2022. Lecture Notes in Computer Science, vol 13685. Springer, Cham. https://doi.org/10.1007/978-3-031-19806-9_29
12. Supervised Machine Learning: Regression and Classification [Електронний ресурс] – Режим доступу до ресурсу: https://www.coursera.org/learn/machine-learning.
13. Statistics [Електронний ресурс] – Режим доступу до ресурсу: https://www.britannica.com/science/statistics/Numerical-measures.
14. Yurii O. Hodlevskyi, Tetiana A. Vakaliuk, Oleksii V. Chyzhmotria, Olena H. Chyzhmotria and Oleh V. Vlasenko. Finding Anomalies in the Operation of Automated Control Systems Using Machine Learning. Proceedings of the 4th

International Workshop on Intelligent Information Technologies & Systems of Information Security, Khmelnytskyi, Ukraine, March 22–24, 2023. Edited by Tetiana Hovorushchenko, Oleg Savenko, Peter T. Popov, Sergii Lysenko. CEUR Workshop Proceedings (CEUR-WS.org, ISSN 1613-0073). Vol. 3373. Pp. 681-698. https://ceur-ws.org/Vol-3373/paper47.pdf

15. Preeti Rajni Bala, Ram Pal Singh. A dual-stage advanced deep learning algorithm for long-term and long-sequence prediction for multivariate financial time series. Applied Soft Computing. Volume 126, 2022, 109317. https://doi.org/10.1016/j.asoc.2022.109317.
16. Sheng Xiang, Yi Qin, Caichao Zhu, Yangyang Wang, Haizhou Chen. LSTM networks based on attention ordered neurons for gear remaining life prediction. ISA Transactions. Volume 106, 2020. Pp. 343-354. https://doi.org/10.1016/j.isatra.2020.06.023.
17. Sheng Xiang, Yi Qin, Caichao Zhu, Yangyang Wang, Haizhou Chen. Long short-term memory neural network with weight amplification and its application into gear remaining useful life prediction. Engineering Applications of Artificial Intelligence.Volume 91, 2020, 103587, https://doi.org/10.1016/j.engappai.2020.103587.
18. Anuraganand Sharma. Guided Stochastic Gradient Descent Algorithm for inconsistent datasets. Applied Soft Computing. Volume 73, 2018, pp. 1068-1080. https://doi.org/10.1016/j.asoc.2018.09.038.
19. Farajtabar M., Azizan N., Mott A., Li A. Orthogonal gradient descent for continual learning. Proceedings of the 23rdInternational Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108. https://core.ac.uk/download/pdf/345075797.pdf
20. Azizan, Navid & Hassibi, Babak. (2018). Stochastic Gradient/Mirror Descent: Minimax Optimality and Implicit Regularization. ICLR 2019. https://openreview.net/pdf?id=HJf9ZhC9FX

21. Simon Du, Jason Lee, Haochuan Li, Liwei Wang, Xiyu Zhai. Gradient Descent Finds Global Minima of Deep Neural Networks. Proceedings of the 36th International Conference on Machine Learning, PMLR 97:1675-1685, 2019. http://proceedings.mlr.press/v97/du19c.html
22. Lee, J.D., Simchowitz, M., Jordan, M.I. & Recht, B.. (2016). Gradient Descent Only Converges to Minimizers. 29th Annual Conference on Learning Theory, in Proceedings of Machine Learning Research. 49: 1246-1257. https://proceedings.mlr.press/v49/lee16.html.
23. Soukup D., Cejka T., Hynek K. (2020) Behavior Anomaly Detection in IoT Networks. Lecture Notes on Data Engineering and Communications Technologies, 49, pp. 465 - 473. DOI: 10.1007/978-3-030-43192-1_53
24. Grcić, M., Bevandić, P., Šegvić, S. (2022). DenseHybrid: Hybrid Anomaly Detection for Dense Open-Set Recognition. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds) Computer Vision – ECCV 2022. ECCV 2022. Lecture Notes in Computer Science, vol 13685. Springer, Cham. https://doi.org/10.1007/978-3-031-19806-9_29
25. Supervised Machine Learning: Regression and Classification [Електронний ресурс] – Режим доступу до ресурсу: https://www.coursera.org/learn/machine-learning.
26. Statistics [Електронний ресурс] – Режим доступу до ресурсу: https://www.britannica.com/science/statistics/Numerical-measures.

ДОДАТКИ

ДОДАТОК А

Код алгоритму для розділення та стандартизації даних

```
df = pd.read_csv('data/test.csv')
df = df[['Date', 'Value']]
df['Date'].min(), df['Date'].max()
train = df.loc[:500,:]
test = df.loc[500:,:]
scaler = StandardScaler()
scaler = scaler.fit(np.array(train['Value']).reshape(-1, 1))
train['Value'] = scaler.transform(np.array(train['Value']).reshape(-1, 1))
test['Value'] = scaler.transform(np.array(test['Value']).reshape(-1, 1))
plt.plot(train['Value'], label='scaled')
plt.legend()
plt.show()
TIME_STEPS = 5


def create_sequences(X, y, time_steps=TIME_STEPS):
    X_out, y_out = [], []
    for i in range(len(X) - time_steps):
        X_out.append(X.iloc[i:(i + time_steps)].values)
        y_out.append(y.iloc[i + time_steps])

    return np.array(X_out), np.array(y_out)


X_train, y_train = create_sequences(train[['Value']], train['Value'])
X_test, y_test = create_sequences(test[['Value']], test['Value'])
print("Training input shape: ", X_train.shape)
print("Testing input shape: ", X_test.shape)
```

ДОДАТОК Б

Код генерації нейронної мережі

```
model = Sequential()
model.add(LSTM(128, activation='tanh', input_shape=(X_train.shape[1], X_train.shape[2])))
model.add(Dropout(rate=0.2))
model.add(RepeatVector(X_train.shape[1]))
model.add(LSTM(128, activation='tanh', return_sequences=True))
model.add(Dropout(rate=0.2))
model.add(TimeDistributed(Dense(X_train.shape[2])))
model.compile(optimizer=keras.optimizers.Adam(learning_rate=0.001), loss="mse")
model.summary()
history = model.fit(X_train,
                    y_train,
                    epochs=100,
                    batch_size=32,
                    validation_split=0.1,
                    callbacks=[keras.callbacks.EarlyStopping(monitor='val_loss', patience=5, mode='min')],
                    shuffle=False)
X_train_pred = model.predict(X_train)
```

ДОДАТОК В

Код виявлення аномалії

```
X_train_pred = model.predict(X_train)
train_mae_loss = np.mean(np.abs(X_train_pred - X_train), axis=1)

plt.hist(train_mae_loss, bins=50)
plt.xlabel('Train MAE loss')
plt.ylabel('Number of Samples')
threshold = np.max(train_mae_loss)
print('Reconstruction error threshold:', threshold)
X_test_pred = model.predict(X_test, verbose=1)
test_mae_loss = np.mean(np.abs(X_test_pred - X_test), axis=1)
plt.hist(test_mae_loss, bins=50)
plt.xlabel('Test MAE loss')
plt.ylabel('Number of samples')
anomaly_df = pd.DataFrame(test[TIME_STEPS:])
anomaly_df['loss'] = test_mae_loss
anomaly_df['threshold'] = threshold
anomaly_df['anomaly'] = anomaly_df['loss'] > anomaly_df['threshold']
```

ДОДАТОК Г

```
anomalies = anomaly_df.loc[anomaly_df['anomaly'] == True]
print(anomalies)
fig = go.Figure()
fig.add_trace(go.Scatter(x=df['Date'], y=df['Value'], name='Value'))
fig.add_trace(go.Scatter(x=anomalies['Date'], y=anomalies['Value'], mode='markers', name='Anomaly'))
fig.update_layout(showlegend=True, title='Detected anomalies')
fig.show()
df.to_csv('data/clear_data.csv', sep=',', index=False)
anomalies.to_csv('data/anomalies.csv', sep=',', index=False)
```