

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

«На правах рукопису»
УДК 004.896

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

« ___ » _____ 2022 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою «Інформаційні управляючі системи та технології»

зі спеціальності 126 «Інформаційні системи та технології»

на тему: «Інформаційна система виявлення захворювань худоби»

Виконав:

студент VI курсу, групи ІС-12мпв
Левицький Костянтин Костянтинович

Керівник:

доцент каф ІСТ, к.т.н., доцент,
Коган Алла Вікторівна

Рецензент:

доцент каф ОТ, к.т.н., доцент,
Роковий Олександр Петрович

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____

Київ – 2022 року

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки**

Кафедра інформаційних систем та технологій

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«___» _____ 2022 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Левицькому Костянтину Костянтиновичу

1. Тема дисертації «Інформаційна система виявлення захворювань худоби», науковий керівник дисертації Коган Алла Вікторівна, доцент, кандидат технічних наук, затверджені наказом по університету від «01» 12 2022 р. № 4351-с

2. Термін подання студентом дисертації _____

3. Об'єкт дослідження — бізнес-процеси сільськогосподарського підприємства, що займається тваринництвом, виявлення захворювань худоби на підприємстві.

4. Перелік завдань, які потрібно розробити:

- провести огляд існуючих систем моніторингу здоров'я худоби;
- провести аналіз методів виявлення захворювань;
- програмна реалізація алгоритмів та розробка програмного забезпечення;

5. Орієнтовний перелік графічного (ілюстративного) матеріалу:

- діаграма прецедентів;
- схема зв'язків сутностей;
- схема структурна компонентів;
- схема розгортання

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Дослідження літератури та документації	02.03.2022	
2	Огляд існуючих рішень	01.04.2022	
3	Аналіз теоретичних методів	06.05.2022	
4	Проектування системи	30.06.2022	
5	Програмна реалізація системи	01.09.2022	
6	Тестування та виправлення помилок	01.11.2022	
7	Оформлення документації	15.11.2022	
8	Попередній захист	29.11.2022	

Студент

Костянтин ЛЕВИЦЬКИЙ

Науковий керівник

Алла КОГАН

РЕФЕРАТ

Магістерська дисертація: с. 123, рис. 41, табл. 30, 32 джерела, 8 додатків.

Агропромисловий комплекс є значущою та невід'ємною складовою економік багатьох країн світу, зокрема України, забезпечуючи 10% ВВП.

Виявлення ознак хвороб у тварин за допомогою розпізнавання звуків є актуальною проблемою, що постає перед промисловістю сьогодні, так як найбільші втрати поголів'я промислового тваринництва припадає на втрати від респіраторних захворювань. В свою чергу вчасне інформування має зупинити подальше розповсюдження інфекції та пришвидшити початок лікування.

Тому актуальною є розробка програмного продукту для виявлення захворювань худоби та інформатизація процесу обміну звітами щодо виявлених захворювань між співробітниками підприємств.

Метою магістерської дисертації є зменшення часу виявлення респіраторних захворювань у худоби та покращення процесу інформування співробітників та керівництва підприємств.

Цілями розробки є:

- автоматичне інформування ветеринарів про можливі захворювання;
- зменшення витрат часу співробітників на створення звітів про стан тварин на підприємстві;
- створення зручної системи для автоматичного відслідковування поточного стану мікро та макроклімату та попередження про виявлення потенційних респіраторних захворювань.

Об'єктом досліджень є процес інформування співробітників сільськогосподарського підприємства про можливі випадки респіраторних захворювань худоби.

Предметом досліджень у роботі є методи і алгоритми розпізнавання звуків навколишнього середовища із застосуванням машинного навчання.

**МАШИННЕ НАВЧАННЯ, НЕЙРОННІ МЕРЕЖІ, КЛАСИФІКАЦІЯ ЗВУКІВ,
СІЛЬСЬКОГОСПОДАРСЬКІ ПІДПРИЄМСТВА**

ABSTRACT

Master's Thesis: pp. 123, fig. 41, tab. 30, sources 32, app. 8.

The agro-industrial complex is a significant and integral part of the economies of many countries, including Ukraine, providing 10% of GDP.

Detection of signs of animal diseases using sound recognition is an actual problem the industry is facing today, since the greatest losses of animals in industrial livestock are caused by respiratory diseases. Therefore, timely information should stop the further spread of infection and speed up the start of treatment.

Therefore, the development of a software product for detecting livestock diseases and computerization of the process of exchanging reports on detected diseases between employees of companies is relevant.

The purpose of the master's thesis is to reduce the time of detection of respiratory diseases in livestock and to improve the process of informing employees and management of enterprises.

The objectives of the development are:

- reducing the time spent by employees on creating reports on the condition of animals at the livestock;
- automatic informing of veterinarians about possible diseases;
- creation of a convenient system for automatic monitoring of the current state of micro and macro climate and warning of potential respiratory diseases.

The object of research is the process of informing employees of an agricultural enterprise about possible cases of respiratory diseases of livestock.

The subject of research in the work are methods and algorithms for recognizing environmental sounds using machine learning.

MACHINE LEARNING, NEURAL NETWORKS, SOUND CLASSIFICATION,
AGRICULTURAL ENTERPRISES

ЗМІСТ

Вступ.....	9
1 Проектні рішення з розробки системи виявлення захворювань худоби....	12
1.1 Опис бізнес-процесів	12
1.1.1 Опис процесу діяльності.....	12
1.1.2 Актори і функції	13
1.2 Опис постановки задачі.....	14
1.2.1 Цілі та задачі розробки.....	14
1.3 Наявні рішення	15
1.3.1 Наявні технічні рішення для розпізнавання звуків	15
1.3.2 Наявні аналоги систем моніторингу ферм	17
1.4 Сутності присутні у системі.....	20
2 Математичне забезпечення для розпізнавання звуків	22
2.1 Дискретизація звукової хвилі.....	24
2.2 Перетворення Фур'є для обробки звукових сигналів.....	25
2.3 Фреймінг сигналу та віконна функція	29
2.4 Виділення ознак звукових сигналів.....	32
2.5 Алгоритми класифікації	41
2.6 Принципи роботи штучних нейронних мереж.....	42
2.7 Згорткові нейронні мережі (CNN).....	50
2.8 Математична постановка задачі	56
3 Архітектурні рішення системи.....	58
3.1 Пристрій збору даних	59
3.2 Серверна частина	60

	7
3.3 Клієнтський веб-застосунок.....	61
4 ER-діаграма	62
4.1 Опис сутностей представлених у базі даних	62
5 Опис програмного та технічного забезпечення.....	70
5.1 Вимоги до програмного продукту	70
5.1.1 Вимоги до програмного модулю розпізнавання кашлю	70
5.1.2 Вимоги до серверної частини.....	70
5.1.3 Вимоги до клієнтського застосунку.....	71
5.2 Засоби розробки	71
5.2.1 Засоби розробки модулю розпізнавання кашлю	71
5.2.2 Засоби розробки серверної частини.....	73
5.2.3 Засоби розробки клієнтського застосунку	75
5.3 Розгортання програмного забезпечення	79
5.3.1 Розгортання серверної частини	79
5.3.2 Розгортання клієнтської частини	80
6 Інструкція користувача	82
6.1 Початок роботи	82
6.2 Авторизація у системі.....	82
6.3 Огляд доступних об'єктів підприємств	83
6.4 Управління і огляд об'єкту підприємства	83
6.5 Перегляд показників спостережень.....	84
6.6 Перегляд попереджень	85
6.7 Перегляд та відправка повідомлень	86
6.8 Створення нового об'єкту	87
6.9 Панель адміністратора.....	89

6.10	Виникнення помилок.....	90
7	Стартап-проект	91
7.1	Опис ідеї проекту	91
7.2	Технологічний аудит ідеї	94
7.3	Аналіз ринкових можливостей запуску стартап-проекту	95
7.4	Розроблення ринкової стратегії проекту.....	106
7.5	Розроблення маркетингової програми стартап-проекту	112
	Висновки	118
	Перелік використаних джерел.....	119

ВСТУП

Із розвитком та становленням людства, важливою складовою цивілізації стало тваринництво у різних його формах. Попри розвиток науки і техніки впродовж віків, однією із найбільших проблем сучасного промислового тваринництва залишаються втрати його основного ресурсу — тварин. Серед причин втрати поголів'я у США, перше місце за частотою випадків займають втрати від респіраторно-інфекційних захворювань [31].

Агропромисловий комплекс є значущою та невід'ємною складовою економік багатьох країн світу, зокрема України. За даними Державної служби статистики [1], сільське господарство, є однією із провідних галузей економіки, забезпечуючи 10% ВВП. В свою чергу продукція тваринництва займає 20,9%, що відповідає близько \$4.2 млрд на рік. Тож розвиток галузі та її оптимізація завдяки інформатизації мають значущий вплив на всесвітню економіку.

Промислове тваринництво, особливо у Європі, США та Китаї, вже сьогодні використовує різноманітні інформаційні системи для підтримки бізнес процесів та автоматизації виробництва, такого як: регулювання та підтримка оптимального мікроклімату у приміщеннях ферм, годування тварин за розкладом, підрахунок голів.

Завдяки розвитку новітніх технологій, збільшенню їх доступності та всесвітній глобалізації, підприємства нерідко вдаються до використання моделі віддаленого ведення бізнесу та (або) отримання послуг за контрактом. Часто, для оптимізації штату, працівники, чия спеціалізація не вимагає постійної фізичної присутності на сільськогосподарському підприємстві, працюють віддалено одразу із низкою об'єктів. Таким чином, є актуальною проблема швидкого та повноцінного інформування спеціалістів, щодо поточного статусу на відповідному підприємстві.

Інформаційні системи для аналізу стану здоров'я худоби тільки починають розвиватися та розповсюджуватися у світі. Такі системи, за допомогою зібраних із різних датчиків даних, дозволяють виявляти та фіксувати параметри навколишнього середовища та показників мікроклімату тощо.

Також, для надання додаткової інформації щодо анамнезу життя тварин, крім фіксації спостережень співробітників підприємства, важливими є умови утримання тварин за яких виникають захворювання. Серед них можна виділити наступні:

- температура та відносна вологість повітря;
- вміст у повітрі діоксиду вуглецю.

Гострі респіраторні захворювання напряму пов'язані із запальними процесами у верхніх дихальних шляхах та легенях, що призводять до відомого усім захисного рефлексу — кашлю. Саме тому, одним із основних маркерів інформування системи про виявлення захворювань може слугувати кашель. Так само як він використовується ветеринарами для виявлення хворих тварин серед поголів'я.

Виявлення і розпізнавання ознак респіраторних захворювань є дуже важливою проблемою що постає перед дослідниками сьогодні. Цю задачу вже певний час вирішують за допомогою нейронних мереж та машинного навчання. Серед задач що можна виокремити — це виявлення та подальша класифікація кашлю, що також набула популярності, особливо через спалах коронавірусної інфекції у 2019 році. Вчасне ж виявлення респіраторних захворювань за допомогою інформаційних технологій пришвидшує реагування для усунення причини захворювання і таким чином допомагає уникнути розповсюдження.

Суттєвих успіхів і популярності досягли системи розпізнавання звуків на основі згорткових нейронних мереж (CNN) та їх варіації, що спершу були створенні для розпізнавання зображень. Проте представлення звукових послідовностей у вигляді спектрограми дає можливість застосовувати такого роду нейронні мережі і для розпізнавання фраз, звуків, тощо.

Інформаційні системи, що впроваджуються на підприємствах, здебільшого покликані закрити лише якусь певну частину потреб сучасного агропромислового бізнесу. Будь-то підтримка комунікації між співробітниками, що представлено у одному з рішень [13] або покривають лише складову дистанційного відслідковування параметрів мікроклімату як у системі [5].

Проте жодна з оглянутих існуючих інформаційних систем не може одночасно слугувати інструментом моніторингу, інформування та комунікації співробітників підприємства, а також виявляти кашель.

Розробка інформаційної системи з виявлення захворювань худоби є актуальною, та має на меті забезпечити сільськогосподарські підприємства комплексним рішенням, що покликано:

- поліпшити взаємодію співробітників;
- спростить завчасне виявлення захворювань;
- зможе надавати інформацію про поточні показники.

В рамках даної роботи розглянуто використання штучних нейронних мереж для розв'язання задачі розпізнавання звуків кашлю тварин та створення інформаційної системи для виявлення захворювань із застосуванням цього рішення.

1 ПРОЕКТНІ РІШЕННЯ З РОЗРОБКИ СИСТЕМИ ВИЯВЛЕННЯ ЗАХВОРЮВАНЬ ХУДОБИ

1.1 Опис бізнес-процесів

Об'єктом автоматизації є процес відслідковування сільськогосподарським підприємством статусу та інформації щодо потенційних захворювань худоби на фермі, або декількох окремих фермах та двосторонньої взаємодії між працівниками керуючих підрозділів, співробітниками ферм та віддаленими консультантами.

Керівництво підприємства зацікавлене у вчасному виявленні захворювань у поголів'ї для пришвидшеного реагування і зупинки подальшого розповсюдження інфекцій на більшу кількість голів.

Завдяки інформуванню ветеринарних спеціалістів або керівних органів без використання рекурентних звітів від співробітників, електронних листів, тощо, зменшується час на прийняття рішень бізнесом. Швидші рішення про початок лікування чи ізоляцію хворих тварин потенційно покращать фінансові показники підприємства.

1.1.1 Опис процесу діяльності

Структурна схема основного процесу виявлення захворювань на фермі описана за допомогою BPMN діаграми наведеної у додатку А.

На діаграмі можна виділити основні відділи, що беруть участь у процесах:

- Співробітники підприємства;
- Ветеринари;
- Керівництво підприємства.

Співробітники підприємства, що фізично мають доступ до тварин, збирають та фіксують інформацію щодо змін умов утримання, наявності кашлю у тварин, змін поведінки, тощо. Ця інформація переноситься у звіт, що регулярно надсилається ветеринарам.

Ветеринари в, свою чергу, переглядають звіти на рахунок відхилень та вирішують чи має місце підозра на респіраторне захворювання у тварин за якими ведеться спостереження. Якщо ветеринар вирішує, що на основі отриманих об'єктивних даних можна поставити діагноз про виявлення респіраторного захворювання, він формує звіт для керівного відділу із рекомендаціями для усунення проблеми.

Керівництво підприємства, на основі рекомендацій від ветеринара, може відмовитися від подальших дій або надати запит до ветеринарного відділу на формування нових інструкцій для співробітників ферми на основі рекомендацій чи певного обраного сценарію.

Після отримання запиту, ветеринаром формуються інструкції та відправляються співробітникам підприємства для подальшого виконання.

1.1.2 Актори і функції

Для повноцінного функціонування системи можна виділити наступні ролі та акторів що беруть участь у системі.

Супер-Адміністратори — користувачі з найвищими правами доступу, вони можуть оперувати усіма доменами даних наявних у системі з правами на читання, створення та модифікацію. Зокрема створювати нові підприємства і надавати Адміністративні права користувачам прив'язаним до підприємства.

Адміністратори — роль користувачів, що належить лише до якогось певного підприємства. Користувачі із такою роллю можуть виконувати ведення даних що відносяться до певного підприємства.

Звичайні користувачі — співробітники конкретних підприємств. Мають обмежені права на запис, але майже усі права на читання в рамках підприємства до якого належать.

Докладна взаємодія акторів із основними функціями системи представлена у додатку Б.

1.2 Опис постановки задачі

Призначенням розробки системи є автоматизація бізнес-процесів у великих підприємствах, що стикаються із задачею контролю та ведення багатьох сільськогосподарських об'єктів одночасно.

Дана розробка має спростити процес обміну інформацією між підрозділами в рамках однієї корпорації, інформування керівництва корпорації щодо поточного стану тварин на фермах, зокрема епідеміологічної ситуації.

Передбачається розробка технічного модуля збору акустичної інформації, показників про стан оточення: температури, відносної вологості та вмісту CO₂, а також програмного забезпечення для класифікації зафіксованих звуків з метою виявлення ознак респіраторних захворювань.

Для забезпечення швидшого реагування і прийняття рішень на основі отриманих даних про зміни у мікро та макрокліматі в приміщеннях, а також у стані здоров'я тварин, передбачається створення модуля чату із можливістю включення окремих користувачів та об'єднання їх у групи.

Розробка передбачає створення веб застосунку, що будуть доступні співробітникам підприємств.

1.2.1 Цілі та задачі розробки

Цілями розробки є:

- зменшення витрат часу співробітників підприємства на створення відповідних звітів про поточний стан тварин;
- автоматичне інформування ветеринарів підприємства про захворювання тварин на фермах;
- створити зручну та інтуїтивну систему для автоматичного відслідковування поточного стану мікро та макроклімату та попередження про виявлення потенційних респіраторних захворювань.

Для досягнення поставлених цілей необхідно вирішити наступні задачі:

- створення алгоритму для виявлення та класифікації звуків кашлю;
- визначення технічного забезпечення необхідного для фіксації та передачі даних на сервер;
- розробка програмного забезпечення для модулів фіксації подій та параметрів;
- розробка серверного та клієнтського програмного забезпечення веб-застосунку.

1.3 Наявні рішення

Основними задачами є створення алгоритму розпізнавання кашлю та інформаційної системи, що зберігатиме та надаватиме підприємствам доступ до даних в онлайн режимі. Тож аналіз існуючих рішень можна розділити на огляд технічних рішень з розпізнавання звуків та пошук інформаційних систем-конкурентів на ринку.

1.3.1 Наявні технічні рішення для розпізнавання звуків

Проблему розпізнавання звуків можна розділити на два основних напрямки: напрямок розпізнавання живого мовлення та напрямок розпізнавання звуків оточення.

Розпізнавання мовлення — найпоширеніша проблема у світі із самого початку існування інтерфейсу «людина-комп'ютер», тож і представлених технічних рішень іншими дослідниками доволі багато. Майже усі технологічні гіганти наразі вже мають чи розвивають системи персональних асистентів, що не можуть існувати без методів розпізнання саме мови.

На відміну від розпізнавання мови, розпізнавання окремих звуків оточення набуло популярності значно пізніше, а успіхів у напрацюваннях було досягнуто лише останні 10 років завдяки збільшенню використання машинного навчання у задачах класифікації.

Серед готових рішень, що займаються саме розпізнаванням різних звуків оточення, можна виділити наступні:

- SEE SOUND;
- Abilisense;
- Audio Analytic.

Ці рішення в основному покликані полегшити реагування на оточення саме людям із будь-якими вадами слуху.

SEE SOUND — це пристрій для розпізнавання звуків, створений Wavio, компанії-виробника програмного забезпечення, що належить глухим власникам, і рекламного гіганта AREA 23. На меті є створення продукту, що може інформувати людей із вадами щодо небезпечних чи корисних звуків що їх оточує. Компанія використовує нейронні мережі для розпізнавання звуків, проте продукт знаходиться у стадії розробки, тож даних щодо його ефективності замало.

Abilisense — це компанія, яка розробила технологію, яка пропонує складний спосіб аналізу звуків удома, на роботі, у місті та транспорті. Так само як і попередній представник, використовується для виявлення небезпечних звуків таких як: звуки повітряної тривоги, постріли, вибухи, розбиття скла, крики людей.

Audio Analytic — британська компанія зі штаб-квартирою в Кембриджі, яка розробила запатентовану програмну структуру розпізнавання звуку під назвою AI3, яка пропонує технологію для розпізнавання контексту оточення через зареєстрований звук. Цей фреймворк містить вбудовану програмну платформу, яка може реагувати на низку звуків, таких як димова сигналізація, та сигналізація чадного газу, розбиття вікна, дитячий плач і гавкіт собак.

Підсумовуючи, наявні аналоги здебільшого орієнтовані на сектор охорони здоров'я та інклюзивності, пропонуючи свої рішення малозахищеним верствам населення вільніше себе почувати у світі оточеному звуками. Проте аналіз технологій, які дані компанії використовують для розпізнавання звуків, доводить доцільність застосування нейронних мереж у вирішенні поставленої задачі.

1.3.2 Наявні аналоги систем моніторингу ферм

На даний момент системи моніторингу підприємств здебільшого надаються за SaaS-моделлю, що не передбачає передачі чи встановлення програмного забезпечення у власність замовнику (підприємству). Серед аналогів, що надають подібний до потрібного функціонал були обрані наступні:

- Farm Health Monitor від Farm Health Guardian;
- Nanoenvi FARM від ENVIRA IoT;
- Porphyrio;
- Fancom.

Розглянемо окремо кожен з наявних аналогів докладніше та виділимо переваги та недоліки кожного з них.

Farm Health Monitor — це система звітності про стан здоров'я тварин, яка покращує зв'язок між виробниками, ветеринарами та технічним персоналом для раннього виявлення захворювань.

За допомогою Farm Health Monitor, співробітникам підприємства надається можливість записувати свої об'єктивні спостереження за тваринами, групою тварин, або цілим комплексом.

Записи зроблені співробітниками централізовано надсилаються і зберігаються у загальній системі, тож записи про стан здоров'я стають доступні для усіх співробітників, що знаходяться у групі охорони здоров'я підприємства. Таким чином, повідомляючи про симптоми, стани та лікування в режимі реального часу.

Серед переваг Farm Health Monitor:

- розподіл відповідальності за прийняття рішень між користувачами;
- централізоване сховище даних;
- можливість дистанційного збору звітів від технічного персоналу та передачі ветеринарному підрозділу.

Недоліки системи:

- відсутність автоматизованого збору інформації;
- обмеженість системи у наданні попередньої оцінки записаним даним;

- відсутність розпізнавання захворювань без участі користувача.

Nanoenvi FARM — рішення для моніторингу здоров'я тварин на фермах, від компанії ENVIRA IoT, що сприяє підвищенню ефективності виробництва та комфорту тварин.

Дане рішення включає в себе декілька частин: систему моніторингу та систему оповіщення.

На стороні ферми встановлені модулі відслідковування у кожному з приміщень, а також центральна станція, куди надходять дані від модулів, що передає дані на віддалений сервер компанії. Де зібрані дані використовуються для створення попереджень і сповіщень про події, пов'язані з потенційним ризиком.

Це рішення покликано виявляти фактори, важливі для виробництва та здоров'я, і контролювати бізнес-процеси на фермі. Технічне устаткування дозволяє відстежувати такі показники добробуту тварин у режимі реального часу:

- відносна вологість приміщення;
- температура у приміщенні;
- концентрація аміаку в повітрі.

Рішення підтримує зв'язок із модулями моніторингу та зберігає дані локально з інструментами для аналізу даних і створення попереджень.

Кожен дистанційний модуль може бути оснащений до 8 датчиків. Його конструкція з низьким споживанням означає, що він також може працювати з дуже маленькою сонячною панеллю. Система програмується для збору даних відповідно до бажаного періоду інтеграції, далі дані надсилаються бездротовим способом до центрального модуля через заплановані проміжки часу.

Ці віддалені модулі можуть працювати автономно, без центрального модуля, оскільки вони можуть бути оснащені 3G-модемом для надсилання даних безпосередньо на сервер у хмарі, що належить ENVIRA IoT або третій стороні. Вони використовують стандартні протоколи зв'язку JSON, які можна налаштувати для надсилання даних третім особам.

Кількість модулів, які можна підключити до кожної системи, практично необмежена. Користувач може додавати модулі за потреби. Крім того, для конкретних потреб можна розробити спеціальні модулі.

Серед переваг рішення:

- наявність датчиків для збору інформації;
- створення попереджень про потенційні аномалії;
- можливість створення розподіленої системи окремих датчиків без залучення центрального хабу;
- можливість глибокої інтеграції у наявні системи автоматизації.

Недоліки системи:

- відсутність можливості безпосередньої комунікації у рішенні без додаткових витрат на розробку та інтеграцію;
- відсутність фіксації акустичної інформації та її розпізнавання.

Porphyrio — система розроблена спеціально для використання у птахівній промисловості та відслідковування життєдіяльності домашніх птахів. Система розроблена для роботи у хмарі та побудована таким чином, що буде доступною як для окремого фермера або загального вертикально інтегрованого виробника.

Вбудований у систему алгоритм сприяє оптимізації процесів виробництва підприємства, а також прибутку шляхом моніторингу, обчислення та аналізу всіх відповідних даних, отриманих під час виробничого процесу. Алгоритм також робить точні довгострокові прогнози на основі індивідуальної поведінки поголів'я та зібраних статистичних даних. Ці прогнози дозволяють підприємствам заздалегідь оптимізувати планування виробництва та продажів.

Також система включає ведення журналів слідкування за життєдіяльністю та здоров'ю тварин та створення на їх основі звітів, що можуть бути передані у інші системи через інтегрування.

Серед переваг рішення:

- можливість глибокої інтеграції у наявні системи автоматизації;
- ведення журналів подій на фермах та можливість створення звітів;
- наявність інтегрованих алгоритмів прогнозування;

- використання хмарного середовища.

Недоліки системи:

- відсутність наявного рішення від компанії по автоматичному збору даних;
- відсутність можливості швидкої комунікації з використанням месенджера
- відсутність функціоналу розпізнавання захворювань.

1.4 Сутності присутні у системі

На основі бізнес-процесів, що були описані у розділі 1.2, ми можемо сформуванати основний набір сутностей, представлення яких потребуватиме інформаційна система. У додатку В наведена діаграма зв'язків між сутностями за нотацією Пітера Чена. Це не кінцеве представлення зв'язків та наявних у системі сутностей, але вона покликана виділити основні складові системи.

Розглянемо кожну сутність окремо.

Користувач — сутність для збереження даних користувача, імені, електронної пошти, паролю, тощо. З важливих властивостей несе інформацію про ім'я та роль у системі.

Підприємство — сутність із базовою інформацією про підприємство. Ідентифікується назвою.

Об'єкт — сутність для репрезентації одного конкретного об'єкта підприємства. Несе в собі інформацію щодо назви, локації (географічних координат), країни місцезнаходження, видів худоби що утримується на підприємстві, приналежність до певного підприємства.

Простір для спостережень — це загальна репрезентація абстрактного простору у якому здійснюються спостереження за допомогою встановлених датчиків. Дана сутність передбачає ієрархічну структуру, тож було використано п «само на себе» для можливості об'єднання структур у групи. У тілі сутності зазначається один з попередньо створених у системі типів простору: будівля, поверх, секція, стійло.

Пристрій — сутність, що описує фізичний пристрій із датчиками, його серійний номер та положення у Airspace

Подія — сутність для опису події, що була зафіксована конкретним пристроєм.

Нотатка — сутність для ведення записів стосовно об'єктивних спостережень працівниками у рамках одного об'єкта підприємства.

Чат — сутність для опису процесу обміну повідомленнями між користувачами.

Висновок до розділу

В рамках розділу було розглянуто актуальні бізнес-процеси присутні у сільськогосподарських підприємствах промислового тваринництва, та які мають бути перенесені у інформаційну систему. Зроблено огляд наявних альтернатив для покриття наявних вимог підприємств і була підтверджена актуальність створення інформаційної системи для виявлення захворювань худоби. Був проведений аналіз задач і викликів з якими потрібно впоратися для успішного втілення системи.

2 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ РОЗПІЗНАВАННЯ ЗВУКІВ

Розпізнавання звуків навколишнього середовища (ESC) є основною проблемою обробки аудіосигналів. І хоча дослідження в галузі розпізнавання звуку традиційно зосереджені на мовленні [8] та музичних сигналах [11], проблема розпізнавання звуку навколишнього середовища отримала більше уваги в останні роки.

Деякі області немовного розпізнавання звуку, які були вивчені різною мірою, стосуються розпізнавання конкретних подій за допомогою порівняння із аудіо ретельно створених спеціально для визначення конкретних подій. Такими, часто стають дослідження з класифікації звуків на приналежність чи ні до мовлення, визначення музичних інструментів, музичних жанрів.

Більшість попередніх досліджень акустики зосереджувались насамперед на мові та музиці. ESC є складним у порівнянні з класифікацією мови та музики [18]. Основна причина цього полягає в тому, що звуки навколишнього середовища є нестатичними і не мають певної структури. Розпізнавання мовлення моделює складні слова, розбиваючи їх на фонемі. Звуки навколишнього середовища не мають фонетичної структури. Крім того, звуки навколишнього середовища, на відміну від музики, не мають стаціонарних аспектів, таких як ритм і мелодія [9]. Звуки навколишнього середовища мають низьке відношення сигнал/шум, оскільки мікрофон, що вловлює звуки, розташований не так близько до джерела звуку як при цілеспрямованому мовленні. Оточуюча акустична сцена складається з численних звуків, що накладаються один на одного, що створює проблему для ESC.

У випадку розпізнавання звуків кашлю, буде доцільне використання підходів, заснованих на порівнянні еталонних звуків хворобливого кашлю, відхилення та інших звуків оточення в якості контрольних.

Будь-яке розпізнавання оточуючих звуків, передбачає наступні етапи (рис. 2.1):

1. збір акустичних даних;
2. попередню обробку;
3. виділення ознак;
4. нормалізація ознак;

5. класифікацію.

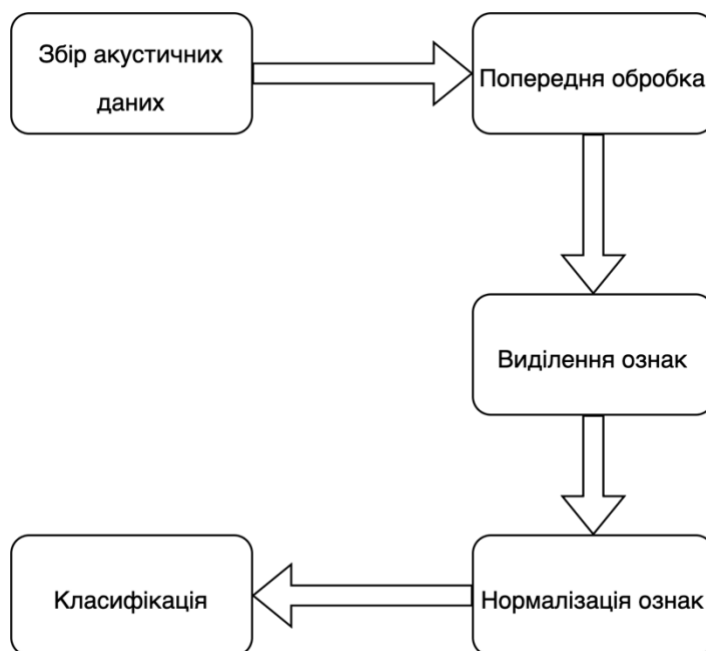


Рисунок 2.1 – Схема етапів розпізнавання звуків

Попередня обробка необхідна для видалення фонових шумів і обробки даних у формі, у якій їх можна використовувати для виділення ознак. Дослідники працювали з різними видами ознак, такими як часові, спектральні та динамічні трансформації шкали часу. Для зменшення кількості початкових ознак, зазвичай обираються тільки важливі, з використанням низки підходів.

Нарешті, різні дослідження показують, що значних успіхів досягають класифікатори на основі машинного навчання, такі як:

- метод опорних векторів (SVM);
- метод k-найближчих сусідів (K-NN) ;
- приховані моделі Маркова (HMM);
- використання різних нейронних мереж.

2.1 Дискретизація звукової хвилі

Звукова хвиля, що фіксується мікрофоном представлена в якості аналогового сигналу. У цифрових системах, звукові хвилі, визначаються як набір значень змін тиску повітря із плином часу. У поточному контексті ми в першу чергу зацікавлені в аналізі та обробці таких сигналів у цифрових системах.

Тоді звуковий сигнал може бути представлений як скінченна послідовність чисел x_n , які представляють відносний тиск повітря в момент часу $n \in \mathbb{N}$. Це подання відоме як імпульсно-кодова модуляція (PCM). Тоді точність цього представлення визначається двома факторами:

1. частота дискретизації (крок у часі між n і та $n + 1$);
2. точність і розподіл амплітуд x_n .

Дискретизація сигналу є класичною задачею обробки та перетворення сигналів (рис 2.2). Найголовнішим аспектом у дискретизації звукових даних є частота Найквіста, яка становить половину частоти дискретизації F_s і визначає верхню межу найбільшої смуги пропускання $\left[0, \frac{F_s}{2}\right]$, якою можна однозначно представити хвилю сигналу що дискретизується. Іншими словами, якщо частота дискретизації становитиме 44100Гц, то сигнали в діапазоні частот від 0 до 22050Гц можна однозначно представити цією частотою дискретизації без втрати даних.

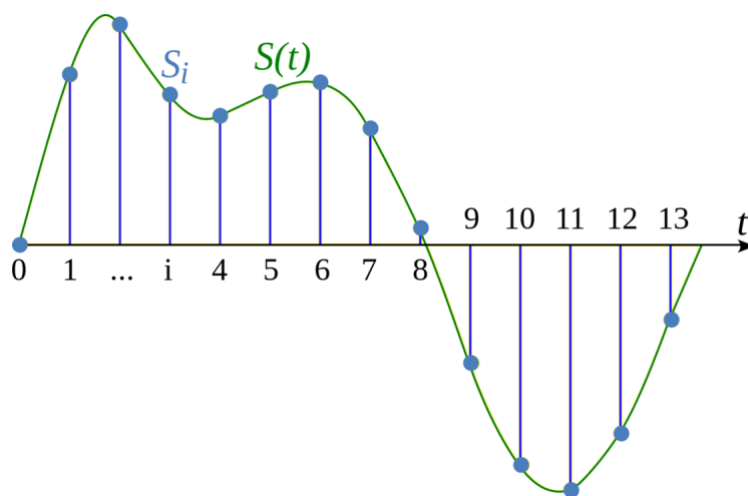


Рисунок 2.2 – Дискретизація неперервної функції $S(t)$ на набір дискретних величин

Вплив частоти дискретизації на якість оберненого представлення сигналу як неперервної функції наочно продемонстровано на рисунку 2.3. Порівняння початкової функції та функцій отриманих вже з дискретних даних із різною частотою дискретизації.

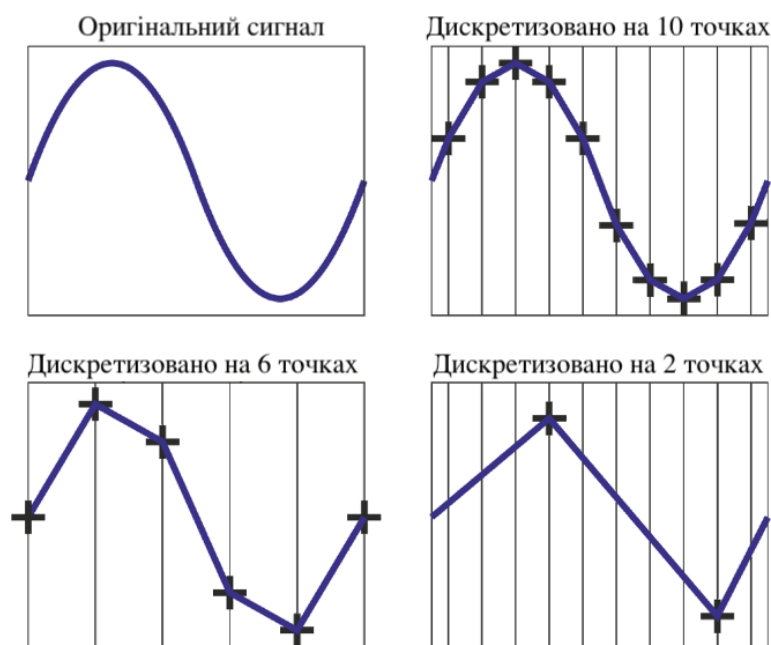


Рисунок 2.3 – Приклад дискретизації неперервної функції за різної частоти

2.2 Перетворення Фур'є для обробки звукових сигналів

Серед характеристик звукового сигналу, одною із найбільш інформативних характеристик, є частотний спектр цього сигналу. Проте інформації щодо частот з яких складається вхідний сигнал ми не отримуємо безпосередньо із самого сигналу. Для цього ми маємо використовувати спеціальні трансформації, що відкриють дорогу до аналізу складніших характеристик звуку.

Ми знаємо, що цифровий звуковий сигнал є набором дискретних величин на певному проміжку часу, тож амплітудна характеристика може бути отримана із часового домену звукового сигналу, в свою чергу опис частотної характеристики звуку може бути отриманий за допомогою перетворення Фур'є. Це перетворення розкладає дану функцію на осциляторні функції, для представлення у вигляді суми

гармонічних коливань і використовується для того, щоб розрахувати спектр частот для сигналів змінних у часі [2].

З точки зору задачі розпізнавання звуків, інформація окремі про складові звукового сигналу є суттєвою ознакою для його класифікації. Для базового пояснення перетворень Фур'є, існує класичний приклад із ідентифікації музичного акорду.

Уявімо, що на вхід подається звуковий сигнал, записаний мікрофоном, що слухав музичний інструмент. Цей сигнал несе у собі інформацію про одразу декілька нот, які є звуковими коливаннями різної частоти. Кожна окрема нота формує свій унікальний звуковий сигнал, тому фінальний сигнал акорду може бути представлений як добуток усіх цих окремих функцій. На рисунку 2.4 представлено два спрощених графіки перетворення сигналу, червоним — вхідний сигнал у часовій осі, а синім — піки присутні у цьому сигналі частоти на осі частот.

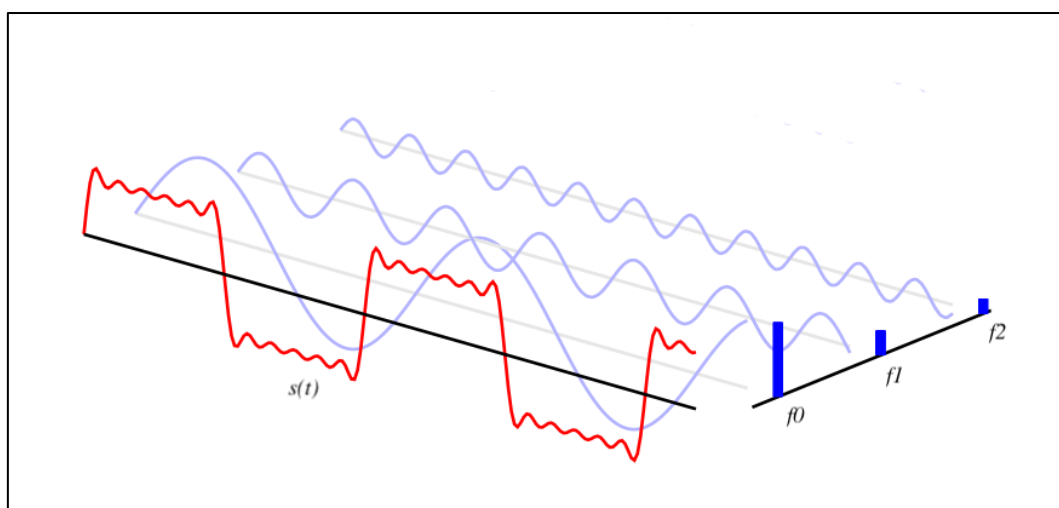


Рисунок 2.4 – Перехід від часового представлення сигналу до частотного

Розглянемо далі основні математичні поняття, потрібні для кращого розуміння перетворення Фур'є .

Рядом Фур'є називають функціональний ряд наступного виду

$$\frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos nx + b_n \sin nx) \quad (2.1)$$

Якщо ряд збігається, то його сума дорівнює періодичній функції $f(x)$ з періодом 2π , оскільки $\sin nx$ та $\cos nx$ є періодичними з періодом 2π .

Сталі числа a_0, a_n, b_n ($n \in \mathbb{N}$) називаються коефіцієнтами тригонометричного ряду. Для визначення коефіцієнтів ряду Фур'є користуються наступними формулами

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos nx \, dx, \quad (2.2)$$

$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin nx \, dx \quad (2.3)$$

Достатньою умовою для представлення періодичної функції у вигляді суми ряду Фур'є є наступні умови: $f(x)$ — періодична функція з періодом 2π , та функція є неперервною або має скінченну кількість точок розриву першого роду на відріжку $[-\pi, \pi]$.

Розглянемо також випадки, коли функція що розкладається є парною чи, навпаки — непарною. У випадку, коли функція є парною, тобто $f(x) = f(-x)$, ряд Фур'є складатиметься лише з косинусів, через те, що всі коефіцієнти при синусах дорівнюватимуть нулю. У іншому випадку, коли функція є непарною, тобто $f(-x) = -f(x)$, ряд Фур'є, навпаки, складатиметься лише з синусів, через нульові коефіцієнти при косинусах.

У випадку із неперіодичними функціями, їх неможливо розкласти у ряд Фур'є, оскільки сума ряду Фур'є є періодичною. Проте таку функцію можна розкласти на довільному проміжку $[a, b]$, на якому вона задовольнятиме умови Діріхле, розглянувши періодичне продовження її на всю числову вісь.

Перетворення Фур'є є аналогічним розкладу у ряд Фур'є неперіодичної функції і математично представлено наступною формулою перетворення функції $f(t)$ у комплексну функцію $F(\omega)$.

$$F(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-i\omega t} dt \quad (2.4)$$

Також є можливість зробити обернене перетворення, що представляється наступним виразом

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(\omega)e^{i\omega t} d\omega \quad (2.5)$$

У випадку роботи з цифровим звуковим сигналом, що подається як набір дискретних значень на певному проміжку часу, нас найбільше цікавить саме дискретна версія перетворень Фур'є (DFT).

Для переходу від неперервної до дискретної форми перетворення Фур'є, визначимо, що наша неперервна вхідна функція $f(x)$ перетворена на дискретний сигнал $s(t)$. Тож, якщо взяти до уваги, що процес інтегрування використовується для знаходження площі під графіком функції, ми можемо перенести цей принцип і до дискретного представлення сигналу в якості суми окремих значень на всьому проміжку, на якому визначено функцію. Тож для перетворення дискретного сигналу $s(t)$ із кількістю значень N , ми можемо використати наведений далі вираз, у якому $k = [0..N - 1]$ відобразить частотний діапазон на який буде розкладено наш сигнал

$$DFT \left[\frac{k}{N} \right] = \sum_{t=1}^{N-1} s(t)e^{-i2\pi \frac{k}{N}} \quad (2.6)$$

Також, можемо представити перетворення Фур'є у векторній формі, для вектору \vec{s} , розмірністю N , наступним чином

$$\vec{X} = \hat{A}\vec{s} \quad (2.7)$$

Де елементи a_{jk} матриці \hat{A} мають вигляд

$$a_{jk} = e^{-2\pi i \frac{jk}{N}} \quad (2.8)$$

І хоч таким чином ми вже маємо необхідний нам для подальших дій механізм переходу від часового домену сигналу до його частотних складових, все ж таки дослідники у сфері роботи із сигналами, зокрема у проблемах розпізнавання звуків, зіткнулися із проблемою надто вимогливих ресурсів для виконання дискретних перетворень Фур'є за допомогою обчислювальних систем. Причиною цьому є те, що існуючі алгоритми для обчислення дискретних перетворень Фур'є складаються із мінімум двох вкладених циклів розмірністю n , що призводить до визначення складності цих алгоритмів як $\Theta(N^2)$. За такої стрімко зростаючої квадратичної функції, виконання операції перетворення над великими об'ємами даних, такими як набори звукових сигналів, займає занадто багато часу.

Для того, щоб зменшити час, потрібний на виконання перетворення Фур'є для дискретних даних, і наблизити функцію складності алгоритму до прийнятних значень, було запропоновано варіацію перетворення Фур'є під назвою Fast Fourier Transform (FFT), або швидке перетворення Фур'є. Запропонований алгоритм, дозволяє зменшити складність і зробити її майже еталонною — $\Theta(N \log_2 N)$.

2.3 Фреймінг сигналу та віконна функція

Одною із проблем при розпізнаванні звуків є те, що перетворення Фур'є для великого обсягу інформації є доволі ресурсною обчислювальною задачею. Іншою, є відсутність можливості відслідковування змін спектру в динаміці.

Для вирішення цих викликів використовується розбиття вхідного звукового сигналу на окремі фрейми (блоки) рівного розміру. Такі фрейми нестимуть у собі менше загальної інформації і спростить їх перетворення.

Проте із розділенням сигналу на фрейми постає інша проблема пов'язана із спектральним аналізом за допомогою перетворень Фур'є — спектральний витік.

Якщо сигнал у фреймі, виділеному із основного сигналу, має нецілу кількість циклів. Тобто, якщо на проміжку представленому фреймом вкладається неціле число періодів сигналу, відбувається спектральний витік, що спотворює результат. Такий витік призводить до появи додаткових частотних спектрів, що не існували у оригінальному сигналі.

Для нівелювання цього ефекту, використовується метод накладання віконної функції, покликаної створювати затухання сигнал на кінцях певного проміжку. Ці функції зазвичай мають форму схожу до дзвону і мають нульове значення за межами якогось певного інтервалу, а також, найчастіше, симетричні навколо середини такого інтервалу.

У обробці звукових сигналів для подальшого використання у машинному навчанні частіше за все використовуються віконні функції Ханна.

Для заданої дискретної довжини $L \in \mathbb{N}$ зі значенням амплітуди $\frac{L}{2}$, функція Ханна може бути представлена наступною формулою:

$$w_0(x) = \frac{1}{L} \left(\frac{1}{2} + \frac{1}{2} \cos \frac{\pi x}{L} \right) \quad (2.9)$$

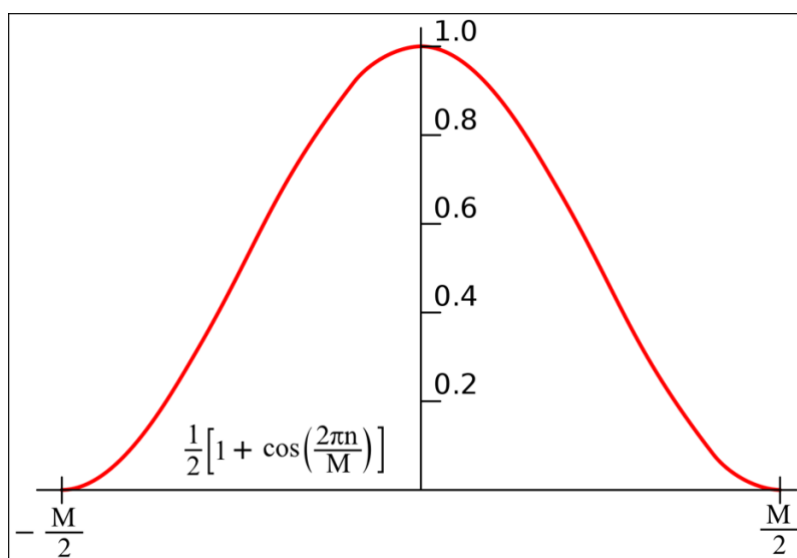


Рисунок 2.5 – Графік функції вікна Ханна

Для згладжування сигналу у фреймі за допомогою віконної функції, виконують множення функції на усій області визначення, таким чином отримуючи новий сигнал для подальшого використання. Накладання віконної функції наочно проілюстровано на рисунку 2.6.

Математично, віконне перетворення дискретної послідовності $s(k)$, де $k = 1 \dots N$, за допомогою віконної функції $w(k)$ виглядає так

$$s_w(k) = s(k)w(k) \quad (2.10)$$

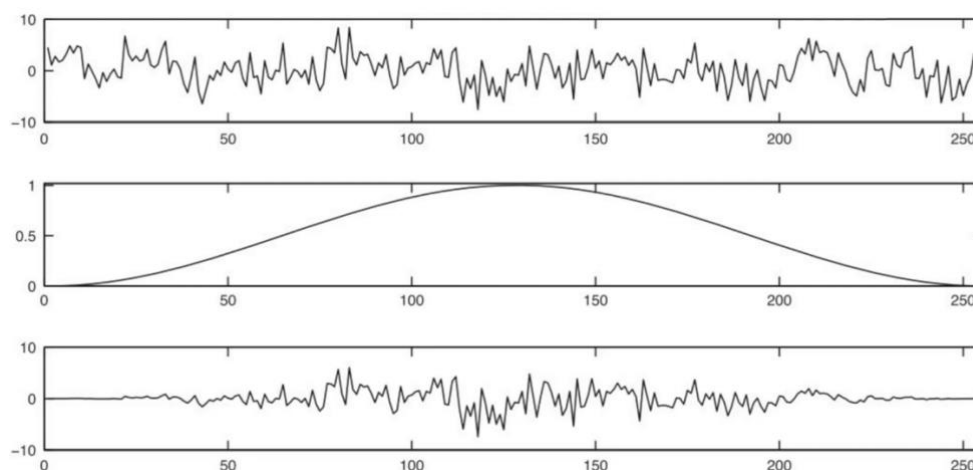


Рисунок 2.6 – Процес накладання віконної функції на оригінальний сигнал

Важливо зазначити, що при використанні віконної функції на окремих фреймах вхідного звукового сигналу ми втрачаємо ті частини цього сигналу, що знаходяться на межах кожного із фреймів (рис. 2.7). Тому, ми не можемо просто ділити вхідний сигнал на рівні за розміром дотичні частини, а маємо використовувати поняття фреймінгу з перекриттям. За такого підходу, ділянки що втрачаються, мають бути нівельовані зсувом кожного наступного фрейму «назад у часі». Пояснення принципу перекриття зображено на рисунку 2.8.

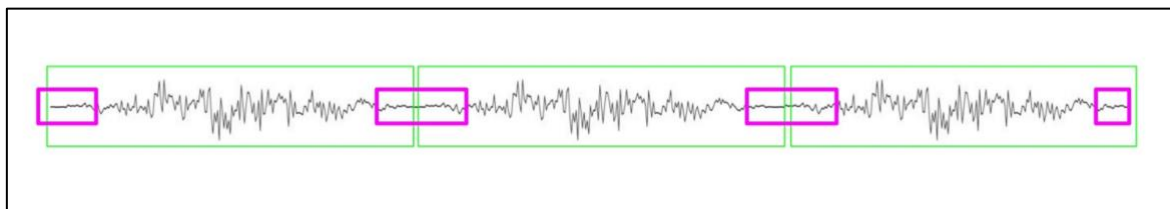


Рисунок 2.7 – Втрачені частини вихідного сигналу

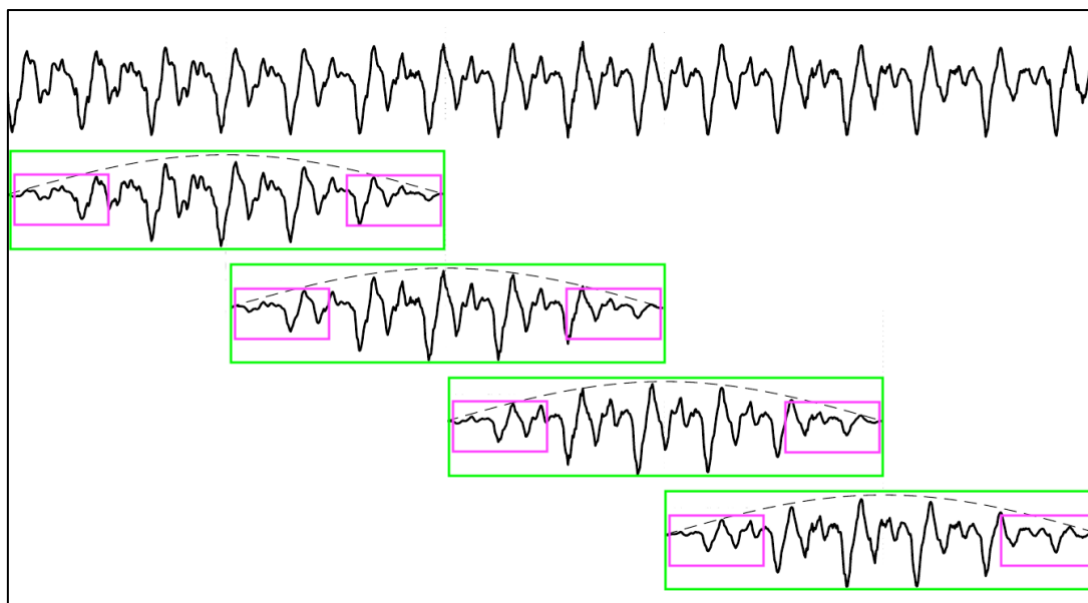


Рисунок 2.8 – Фреймінг з перекриттям

2.4 Виділення ознак звукових сигналів

Ознаки звукового сигналу — це відмінні характеристики звуків, які виділяються та передаються у класифікатори машинного навчання для подальшої обробки. Це один із найважливіших кроків у розпізнаванні звуків, через те, що майже всі методи класифікації вимагають подання на вхід саме попередньо отриманих числових векторів, що визначають якісь певні ознаки звукового сигналу. Для машинного навчання використовуються різноманітні методи виділення ознак, тож далі детально розглянемо різні функції та методи виділення цих ознак зі звукових сигналів.

Ознаки для класифікації можна умовно поділити на такі, що виділяються із темпорального (часового) домену, та частотного домену.

Ознаки темпорального домену

Темпоральні ознаки, що часто також називають ознаками часової області, витягуються безпосередньо зі звуків. Вони часто не потребують складних перетворень та є здебільшого статистичними. Серед таких ознак виділяють: швидкість переходу через нуль (Zero-crossing Rate), середньоквадратичне значення (Root mean square) магнітуди сигналу, амплітудну обвідну сигналу (Amplitude envelope)

Швидкість переходу через нуль — це швидкість, з якою сигнал змінюється з позитивного на нульовий і на навпаки. Для обчислення швидкості переходу використовується наступна формула

$$ZCR = \frac{1}{T-1} \sum_{t=1}^{T-1} 1_{\mathbb{R}<0}(s(t), s(t-1)), \quad (2.11)$$

де $s(t)$ — це сигнал довжиною T ;

$1_{\mathbb{R}<0}$ — індикаторна функція, що набуває значення 1, 0, -1 відповідно до знаку, якого набуває функція

$$1_{\mathbb{R}<0} = \begin{cases} s(t) > 0, & 1 \\ s(t) = 0, & 0 \\ s(t) < 0, & -1 \end{cases} \quad (2.12)$$

Автокореляція — залежність взаємозв'язку між сигналом та його зсунутою копією від величини часового зсуву. Дискретна автокореляція R за відставання l для дискретного сигналу $s(t)$ визначається так

$$R_{yy}(l) = \sum_{t \in \mathbb{Z}} s(t) \overline{s(t-l)} \quad (2.13)$$

Середньоквадратичне значення енергії — це обчислення середньоквадратичного значення амплітуд сигналу $s(t)$ довжиною T на всьому його проміжку.

$$RMS = \sqrt{\frac{1}{T-1} \sum_{t=1}^T s(t)^2} \quad (2.14)$$

Амплітудна обвідна сигналу — по суті, є функцією, що представлена максимальними значенням амплітуди сигналу кожного із дискретних значень, що описують сигнал у фреймі. Ця характеристика дає приблизне уявлення про гучність. Однак вона чутлива до шумів, що виходять за межі діапазону. Ця функція широко використовується для виявлення початку звуку і класифікації музичних жанрів.

$$AE = \max_{t \cdot K \leq k \leq (t+1) \cdot K - 1} s(k) \quad (2.15)$$

Ознаки спектрального домену

Ознаки які ми можемо виділити із спектрального домену звукового сигналу для використання у подальшому аналізі, є здебільшого такими, що були винайдені з огляду на людський механізм сприйняття звуку. Так як найбільш вдалим інструментом розпізнавання звуку допоки все ще є механізм роботи людського слуху, очевидно, що дослідники намагаються досягти результатів що будуть співставні із результатами розпізнавання звуків людиною. Саме тому важливим є розуміння як саме ми розпізнаємо звуки.

Людське вухо може розрізнити звук у діапазоні частот в межах від 20-30 Гц до 20000 Гц в залежності від вроджених здібностей, віку та інших фізіологічних відмінностей. Варто зазначити, що людина здатна реєструвати також і деякі звуки у інфразвуковому та ультразвуковому діапазонах, проте завдяки іншим сенсорним можливостям. В свою чергу, основну інформацію від зовнішнього середовища людина отримує від звуків, чия частота знаходиться у діапазоні до 4кГц, що

еволюційно впливає з того, що основна потрібна для життєдіяльності інформація, така як голоси людей, тварин та важливі оточуючі звуки належать саме до цього спектру частот від 20Гц до 4кГц. Вищий частотний діапазон є надлишковим і здебільшого несе лише додаткову інформацію.

Для розпізнавання звуків, людина орієнтується висоту звуку, його силу та його тембр.

Висота звуку, або ж тон — це слухове відчуття, при якому слухач розподіляє музичні тони за відносними позиціями на музичній шкалі, ґрунтуючись насамперед на своєму сприйнятті частоти коливань. Висота звуку тісно пов'язана з частотою, але ці два поняття не є еквівалентними. Частота — це об'єктивний, науковий атрибут, який можна виміряти. Висота — це суб'єктивне сприйняття звукової хвилі кожною людиною, яке не може бути безпосередньо виміряне. Однак це не обов'язково означає, що більшість людей не погодиться з тим, які ноти є вищими, а які нижчими. Оскільки висота звуку є настільки близькою до частоти, вона майже повністю визначається тим, наскільки швидко звукова хвиля змушує повітря вібрувати, і майже не має нічого спільного з інтенсивністю, або амплітудою, хвилі. Тобто, "висока" частота означає дуже швидкі коливання, а "низька" — більш повільні.

Тембр, також відомий як забарвлення тону або якість — це сприйнята якість звучання музичної ноти, звуку або тону. Тембр розрізняє різні типи звуковидобування, такі як хорові голоси та музичні інструменти. Простіше кажучи, тембр — це те, що робить певний музичний інструмент або людський голос відмінним від іншого, навіть коли вони грають або співають одну і ту ж ноту. Наприклад, це різниця у звучанні між гітарою та фортепіано, які грають одну і ту ж ноту з однаковою гучністю. Обидва інструменти можуть звучати однаково налаштованими по відношенню один до одного, граючи одну і ту ж ноту, і, граючи на одному і тому ж рівні амплітуди, кожен інструмент все одно буде звучати по-своєму, зі своїм унікальним забарвленням тону. Фізичні характеристики звуку, що визначають сприйняття тембру, включають частотний спектр і амплітудну обвідну.

Сила звуку, або його інтенсивність — це кількість енергії звуку, що проходить через якусь певну площину за одиницю часу. Проте, для дослідження відчуття звуку

людиною, окремо сила звуку використовуватися не може, через те, що відчуття гучності людиною є нелінійним, а логарифмічним. Тобто за різної частоти звуку, його сила відчувається людиною по-різному. Тож в акустиці і обробці сигналів поширено використання не абсолютного значення сили звуку, а відносного за допомогою одиниці вимірювання — бел, або зазвичай його десятої частини — децибела.

Децибел — логарифмічна величина, що позначається дБ, і є відношенням сили (інтенсивності) певного сигналу I до певної нульової сили $I_0 = 10^{-12} \text{Вт/м}^2$. Має наступну формулу для обчислення:

$$L = 10 \lg \frac{I}{I_0} \quad (2.16)$$

У слухача також є певні обмеження не тільки до сприйняття частотної характеристики звуку, а також і обмеження у виявленні цих звуків в залежності від частоти таких звуків. Така особливість називається порогом чутності людини (рис 2.9), що її залежить віку, фізіологічних особливостей, тощо.

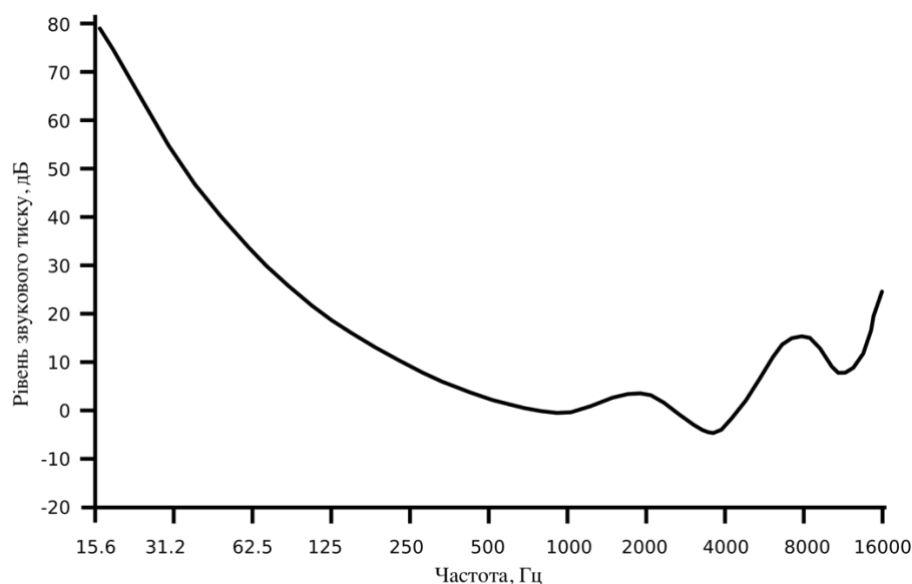


Рисунок 2.9 – Графік-відображення порогу чутності у людини

Спектрограма звукового сигналу — важлива і корисна ознака для класифікації звуків за допомогою машинного навчання. Вона наочно показує яких змін у

частотному домені зазнавав сигнал із плином часу. Для позначення амплітудної складової частоти, її переносять у третій вимір на тривимірному графіку, або використовують кодування кольором.

Інформація про те, яких змін набували частоти звукового сигналу за проміжок часу на якому він записаний, є однією із головних для ідентифікації різних звуків. Саме тому, у проблемах розпізнавання та класифікації звуків не застосовують просте перетворення Фур'є на всьому звуковому сигналі, бо ми отримаємо спектр, який є середнім значенням всіх частин звуку, а ділять його на під-фрейми використовуючи так зване віконне перетворення Фур'є. Розбиваючи сигнал на більш короткі сегменти, ми можемо зосередитися на властивостях сигналу в конкретний момент часу.

Віконне перетворення Фур'є (STFT) на практиці є звичайним дискретним перетворенням Фур'є, що застосовують до окремих фреймів, або віконних проміжків, і для перетворення сигналу для перетворення дискретного сигналу $s(t)$ із кількістю значень N , з використанням віконної функції $w(t)$, де використано зазор між вікнами розміру H , ми можемо застосовувати наведений далі вираз, у якому $k = [0..N - 1]$, а m відповідає за порядковий номер вікна у розбитті.

$$STFT(m, k) = \sum_{t=1}^{N-1} s(t)w(t)e^{-i2\pi\frac{k}{N}} \quad (2.17)$$

Спектрограму можна уявити собі як набір результатів STFT, накладених один на одного. Це спосіб візуального представлення гучності або амплітуди сигналу, як вона змінюється з часом на різних частотах. При обчисленні спектрограми за лаштунками відбуваються деякі додаткові деталі. Вертикальна вісь перетворюється в логарифмічну шкалу частоти, а розмірність відображена кольором — децибели, які так само є логарифмічною шкалою для амплітуди сигналу.

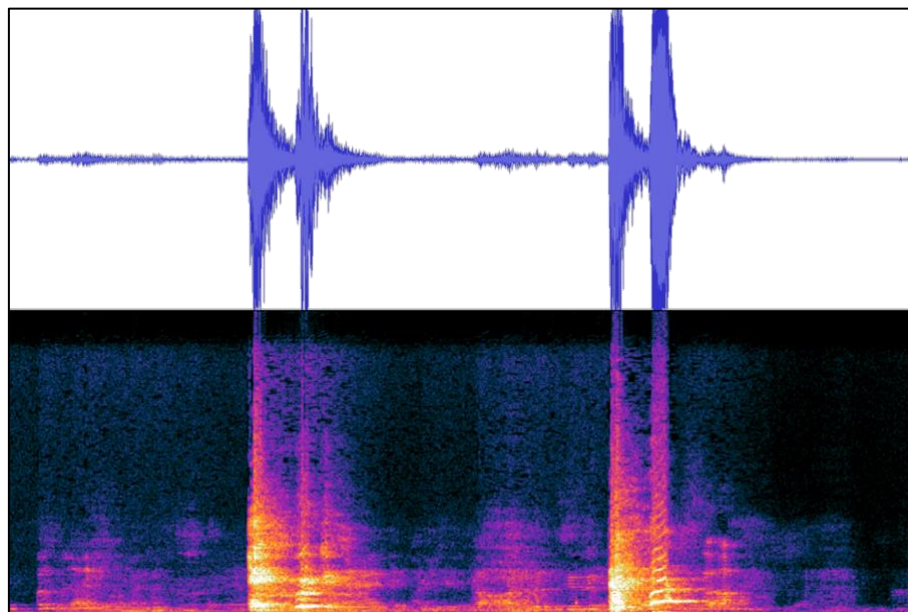


Рисунок 2.10 – Цифровий запис звуку кашлю та відповідна йому спектрограма

Мел-спектрограма звукового сигналу — аналогічна ознака, що може бути отримана шляхом додаткових маніпуляцій зі спектрограмою звуку. Для отримання мел-спектрограми, ми використовуємо накладання частотної складової на мел-шкалу.

Як було згадано раніше, сприйняття звуків людським слуховим апаратом є нелінійним, а скоріше логарифмічним. Наприклад, людина з легкістю відчує різницю між 500 і 1000 Гц, але навряд чи зможе відчути різницю між 10000 та 10500 Гц, навіть за у мови, що ширина цих двох частотних діапазонів рівна. Тому була введена шкала мел.

Мел-шкала — це перцептивна, тобто емпірично побудована на суб'єктивному відчутті висоти звуку людиною, шкала висот тонів, які слухачі вважають рівними за відстанню один від одного. Точка відліку між цією шкалою та звичайним вимірюванням частоти визначається шляхом присвоєння тону частотою 1000 Гц, що на 40 дБ вище порогу чутності слухача, значення висоти звуку, що сприймається, у 1000 мелів. На частоті приблизно вищій за 500 Гц слухачі оцінюють вже більші інтервали як такі, що дають однаково рівні прирости висоти тону.

Для того, щоб отримати значення мел із частоти у Гц, застосовують формулу:

$$m = 2595 \cdot \log\left(1 + \frac{f}{500}\right), \quad (2.18)$$

де f — значення частоти у Гц. Залежність представлено графіком на рисунку 2.11. А представлення спектрограми у мелах відображено на рисунку 2.12.

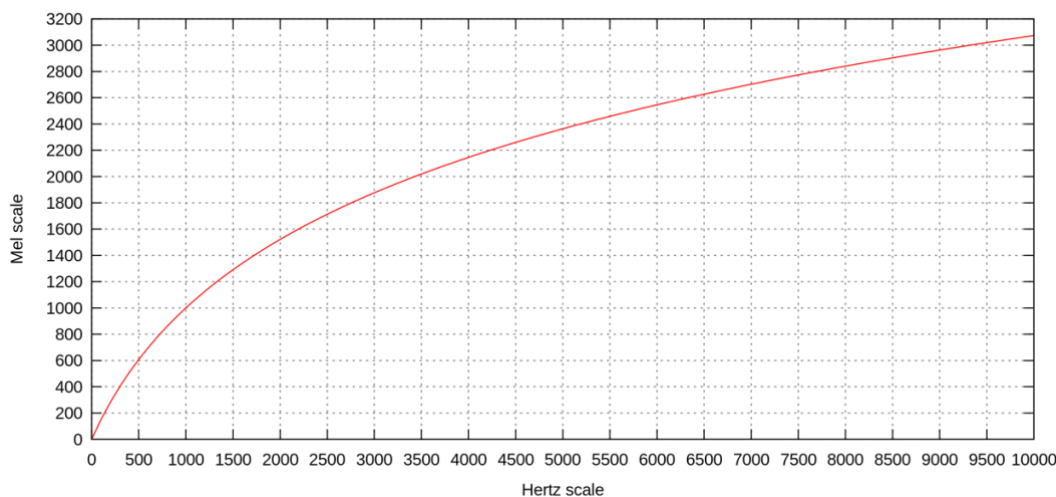


Рисунок 2.11 – Графік відношення значень частоти в Гц до одиниць у мелах

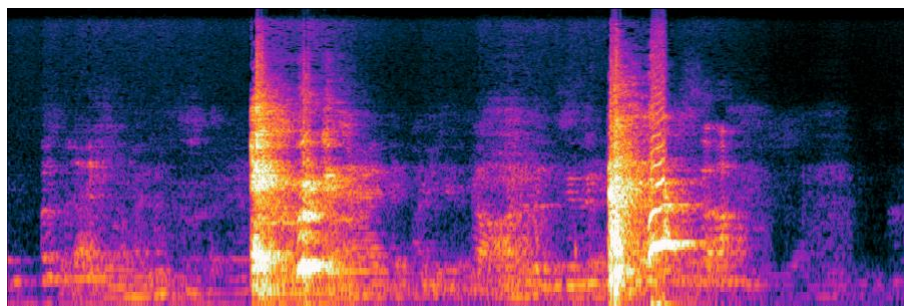


Рисунок 2.12 – Мел-спектрограма звуку кашлю

Мел-частотні кепстральні коефіцієнти (MFCC) — також одна із популярних ознак, що широко використовуються в класифікації звуку в сферах музики [17], мови [19] та звуків навколишнього середовища [26, 31]. Це такі коефіцієнти, що складають мел-частотний кепстр.

Кепстр — це, по суті, зворотне перетворення спектру логарифма спектру часового сигналу. Отриманий спектр не знаходиться ні в частотній, ні в часовій області і несе в собі інформацію про швидкість зміни спектральних смуг сигналу. Він обчислюється як зворотне перетворення Фур'є від результату застосування

логарифмічної функції на перетворенні Фур'є вхідного сигналу. Формула, що стоїть за цим обчисленням, виглядає наступним чином

$$C_s(q) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \ln|S(\omega)|^2 e^{i\omega q} d\omega, \quad (2.19)$$

де $S(\omega)$ — спектр вхідного сигналу, отриманий після перетворення Фур'є; аргумент q — особливий кепстральний час, через те, що $C_s(q)$, в будь-який момент часу q залежить від функції $s(t)$ вихідного сигналу зі спектром $S(\omega)$.

Кепстральний час іноді називається «сачтота» або «quifrency», як анограми до слів «частота» та «frequency».

Також у дискретній формі має альтернативу для перетворення

$$CEP[s(t)] = F^{-1}[\log(F[s(t)])], \quad (2.20)$$

де $s(t)$ — дискретний сигнал;

F — перетворення Фур'є;

F^{-1} — зворотне перетворення Фур'є.

Спектральний центроїд — є ознакою заснованою на визначенні центру мас отриманого спектру. І у деяких роботах використовується як показник «зabarвлення» звуку, і застосовується для визначення тембру. Його обчислюють як середньозважене значення частот, присутніх у сигналі, визначених за допомогою перетворення Фур'є, з їх амплітудами в якості вагових коефіцієнтів

$$SC_t = \frac{\sum_{n=0}^{N-1} f_t(n) m_t(n)}{\sum_{n=0}^{N-1} m_t(n)}, \quad (2.21)$$

де $f_t(n)$ — значення частоти для дискретного елементу n ;

$m_t(n)$ — значення амплітуди для цього елементу.

Ширина пропускання спектру — визначається як середньозважене відстаней між частотними спектрами спектрального центроїду. Застосовується у класифікації музикальних жанрів.

$$BW_t = \frac{\sum_{n=0}^{N-1} |n - SC_t| \cdot m_t(n)}{\sum_{n=0}^{N-1} m_t(n)} \quad (2.22)$$

де $f_t(n)$ — значення частоти для дискретного елементу n ;

$m_t(n)$ — значення амплітуди для цього елементу.

2.5 Алгоритми класифікації

Існує багато різних видів машинного навчання для вирішення задач класифікації, кожен із них має різноманітне застосування для прикладних задач. Розглянемо найбільш популярні із них далі.

Лінійна регресія — один із найпростіших видів машинного навчання. Лінійна регресія є поширеним статистичним інструментом для моделювання взаємозв'язку між деякими "визначальними" змінними та деяким реальним результатом, що оцінюється. Розглянемо задачу навчання, в якій предметна область X є підмножиною \mathbb{R}^d для деякого d , а множина ознак Y є множиною дійсних чисел. Потрібно визначити лінійну функцію $h: \mathbb{R}^d \rightarrow \mathbb{R}$, яка б якнайкраще відтворювала залежність між нашими змінними. Наприклад, за допомогою отриманої функції ми передбачити вагу дитини як функцію її віку та ваги при народженні [30, с. 123].

Алгоритм k -найближчих сусідів — є одним з найпростіших серед усіх алгоритмів машинного навчання. Ідея полягає в запам'ятовуванні навчальної множини і подальшому прогнозуванні мітки будь-якого нового екземпляра на основі міток його найближчих сусідів у навчальній множині. Обґрунтування такого методу базується на припущенні, що ознаки, які використовуються для опису точок домену, мають відношення до їх міток таким чином, що сусідні точки, ймовірно, матимуть

однакові мітки. На практиці алгоритм k -найближчих сусідів зазвичай застосовується після етапу попередньої обробки для зменшення розмірності [30, с. 258].

Метод опорних векторів — інший популярний алгоритм для машинного навчання. Його ідея базується на тому, що за допомогою алгоритму формується певний вектор або площина, яка буде уявно розділяти множину вхідних даних на певні класи. Цей алгоритм дозволяє розділити дані на два класи і підходить для простих проблем класифікації [30, с. 202].

Дерева рішень — класифікатор, ідея якого полягає у створенні дерева, здатного отримувати очікуваний від алгоритму результат. По своїй суті — це взаємопов'язаний набір предикатів виду $h: X \rightarrow Y$, які будуть слідувати до листка дерева доки не отримають шуканий результат. Їх найбільшим недоліком є складність навчання, що потребує застосування спеціальних евристичних алгоритмів для побудови предикатів [30, с. 250].

Нейронні мережі — це модель обчислень, натхненна структурою нейронних мереж у мозку. У спрощених моделях мозку він складається з великої кількості базових обчислювальних пристроїв (нейронів), які з'єднані один з одним між собою в складну комунікаційну мережу, завдяки якій мозок здатен здійснювати надскладні обчислення. Штучні нейронні мережі є формальними обчислювальними конструкціями, які моделюються за цією парадигмою [30, с. 268].

Базуючись на цьому огляді підходів до розпізнавання звуків оточуючого середовища, було обрано штучні нейронні мережі як алгоритм класифікації для подальшого використання у системі, через наявність попередніх успіхів у дослідників цієї тематики.

2.6 Принципи роботи штучних нейронних мереж

Штучні нейронні мережі (ШНМ) допомагають нам вирішувати задачі кластеризації і класифікації певних даних які вони отримують на вхід. За їх допомогою можливо групувати немарковані дані відповідно до схожості із прикладами наданими у вхідних даних. Також вони здатні класифікувати дані, коли

у них є вже створений маркований набір даних для навчання. Слід зазначити, що нейронні мережі також можуть витягувати ознаки, які передаються іншим алгоритмам для кластеризації та класифікації. Таким чином, нейронні мережі можуть слугувати як незалежними рішеннями для кластеризації і класифікації, так і окремими компонентами більших алгоритмів машинного навчання.

Принцип роботи нейронної мережі подібний до нейронної мережі мозку живих істот. І головною перевагою ШНМ є те, що за допомогою певних механізмів навчання, нейронні мережі можуть вирішувати задачі, наприклад розпізнавання котів на фотографії без жодного попереднього алгоритмічного розпізнавання котів.

Доволі довгий час вчені намагаються повторити структуру і принципи роботи людського мозку у цифровій формі, через те, що мозок досі є одним із найефективніших засобів для розпізнавання абстрактних образів, виділення із них певних ознак, та має змогу до навчання на основі попереднього досвіду.

ШНМ базується на наборі з'єднаних вершин, які називаються штучними нейронами, що певною мірою моделюють нейрони біологічного мозку. Кожне з'єднання, подібно до синапсів у біологічному мозку, може передавати сигнал іншим нейронам. Штучний нейрон отримує сигнали, потім обробляє їх і може сигналізувати підключеним до нього іншим нейронам у наступному прошарку. "Сигнал", що отримується на з'єднанні є дійсним числом, а вихід кожного нейрона обчислюється деякою нелінійною функцією від суми його входів.

З'єднання між нейронами називаються ребрами. Нейрони і ребра, як правило, мають свою вагу, яка змінюється в процесі навчання. Вага збільшує або зменшує силу сигналу на з'єднанні. Одним із параметрів нейрону також може бути поріг спрацьовування, так що сигнал надсилається тільки в тому випадку, якщо сукупний сигнал отриманий нейроном перетинає цей поріг.

Зазвичай нейрони об'єднуються в шари. Різні шари можуть виконувати різні перетворення на своїх входах. Сигнали проходять від першого шару (вхідний шар) до останнього (вихідний шар), можливо, після проходження шарів кілька разів.

Для представлення нейронних мереж зазвичай використовуються зважені графи. Де нейрони — вершини такого графу, які можуть набувати певних значень, а ребра — зв'язки між нейронами.

Для початку розглянемо найпростішу форму штучного нейрону — перцептрон.

Перцептрони — винайдені Френком Розенблатом у 1958 році, є найпростішою нейронною мережею, яка складається з n кількості входів, лише одного нейрона та одного виходу, де n — кількість ознак нашого набору даних. Процес проходження даних через нейронну мережу відомий як пряме розповсюдження.

Один перцептрон отримує набір n вхідних значень від попереднього шару нейронної мережі. Він обчислює середньозважений добуток значень вектора вхідних значень $\vec{x} = (x_1, x_2, \dots, x_n)$, виходячи з вагового вектора $\vec{w} = (w_1, w_2, \dots, w_n)$ і додає до результатів зсув. Результат цього обчислення, можна позначити як \vec{z} передається у нелінійну функцію активації $\sigma(z)$, яка формує вихід із блоку обчислень. Наступний рисунок 2.13 надає уявлення щодо підсумовування значень отриманих із n -входів, після перемноження з відповідними вагами. Також, перед тим, як отримати вихідне значення \hat{y} , зважена сума пропускається через функцію активації.

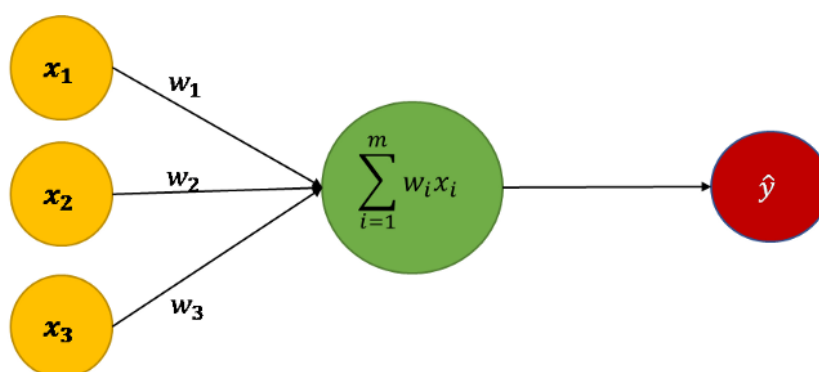


Рисунок 2.13 – Найпростіший перцептрон

Тепер давайте розглянемо сценарій від одного перцептрона до щільного стека з декількох перцептронів, які разом складають шари нейронної мережі. Мережа, що складається з чотирьох шарів, показана на рисунку 2.14.

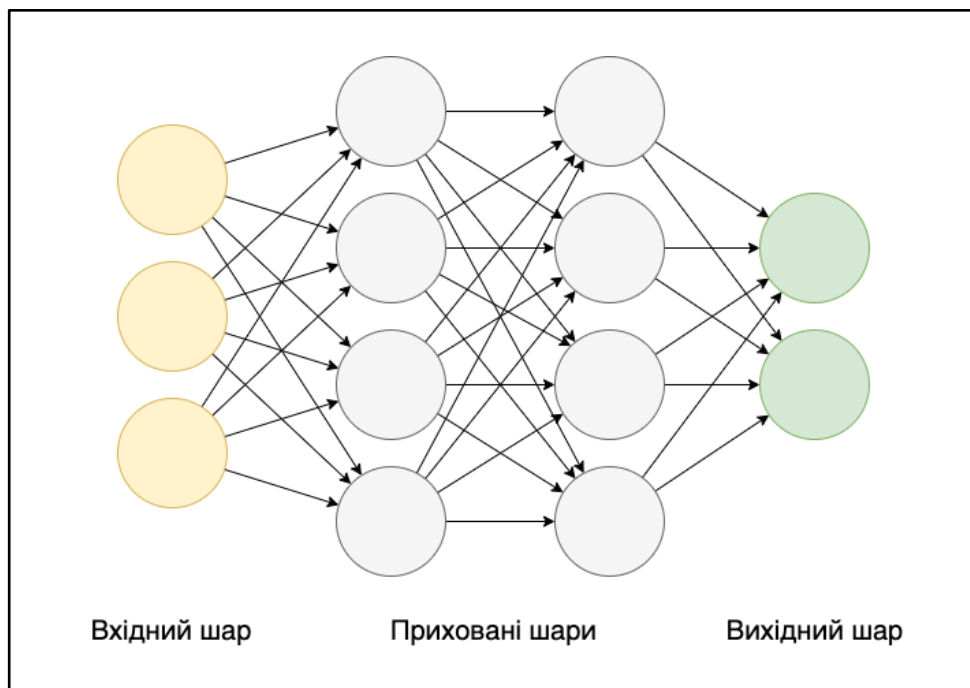


Рисунок 2.14 – Графічне представлення шарів нейронної мережі

На цьому рисунку мережа представляє собою каскад з чотирьох шарів, де кожен шар є стовпчиком перцептронів, накладених один над одним. Для математичного моделювання ми представляємо всі значення, включаючи вхідні, вихідні та проміжні ваги в якості відповідних векторів. Ці вектори будуть схожими до використаних вище для одного окремого перцептрона, але із урахуванням роботи вже не одного, а групи перцептронів об'єднаних у шари. Таким чином, для опису одного шару нейронної мережі ми можемо використовувати вектор вхідних даних та матрицю ваг.

Вектор вхідних значень виглядатиме наступним чином:

$$\vec{X} = \begin{bmatrix} x_1, \\ x_2, \\ \dots \\ x_n \end{bmatrix} \quad (2.23)$$

Переходячи до ваг, слід зазначити, що вага — це параметр, який трансформує вхідні дані в мережі. Кожен шар характеризується унікальною ваговою матрицею W . Тут проміжний шар має вагову матрицю \vec{W} і кожен елемент матриці представлений як вага між відповідними перцептронами.

$$\vec{W} = \begin{bmatrix} w_{11} & \cdots & w_{1m} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nm} \end{bmatrix} \quad (2.24)$$

Компоненти \vec{W} є вагами, що зв'язують вхідний елемент з відповідними перцептронами в даному шарі. Перший індекс n позначає елемент вхідного вектора \vec{X} , який буде надходити на шар, що описується. Другий індекс, m , позначає перцептрон у шарі, на який надходить вхідний сигнал.

Середньозважене значення вхідного вектора з відповідними вагами може бути математично описане як скалярний добуток відповідних векторів. Використовуючи скалярний добуток, ми множимо вхідний вектор \vec{X} на вагову матрицю \vec{W} .

Також, до отриманого скалярного добутку, зазвичай додається вектор зсуву $\vec{b} = (b_1, b_2, \dots, b_n)$ необхідний в більшості випадків, щоб перемістити всю функцію активації ліворуч або праворуч для генерації необхідних вихідних значень. Отримане значення можемо позначити як \vec{z} . Ці кроки в сукупності називаються прямим розповсюдженням.

$$\vec{z} = \vec{X} \times \vec{W} + \vec{b} \quad (2.25)$$

Останнім етапом для опису процесу прямого поширення є нелінійні перетворення за допомогою активаційної функції. Розглянемо докладніше поняття нелінійного перетворення та його роль у процесі прямого поширення.

Традиційні алгоритми машинного навчання базуються виключно на припущенні, що зв'язок між вхідними та вихідними ознаками є лінійним. Вони роблять акцент на створенні лінійності при виведенні рівняння. Але насправді це виявляється лише спотворенням дійсності. Всі оточуючі нас явища насправді нелінійні за своєю природою. За допомогою одного лише лінійного перетворення неможливо відобразити складні закономірності. Таким чином, ми вводимо в систему

новий компонент, який вносить додаткову нелінійність в дані. Це нове доповнення до архітектури нейронної мережі називається активаційною функцією.

Функції активації визначають точність моделі глибокого навчання, а також обчислювальну ефективність навчання моделі. Функції активації також мають великий вплив на здатність нейронної мережі до збіжності, тобто знаходження оптимальних векторів ваг і зсувів. Без них, нейронна мережа перетворилася б лише на комбінацію набору лінійних функцій. В якості елемента активації використовують різні функції, серед яких широко застосовуються такі як:

- ReLU;
- сигмоїда;
- гіперболічний тангенс.

Порівняння графіків різних функцій активації зі звичайною лінійною залежністю наведено на рисунку 2.15.

Проте найуживанішою на даному етапі розвитку нейронних мереж є ReLU функція, а класична сигмоїда, що була застосована як відображення реальних біологічних процесів у нейроні на ранніх етапах створення моделей нейронних мереж, вже не використовується, бо вважається малоефективною [27].

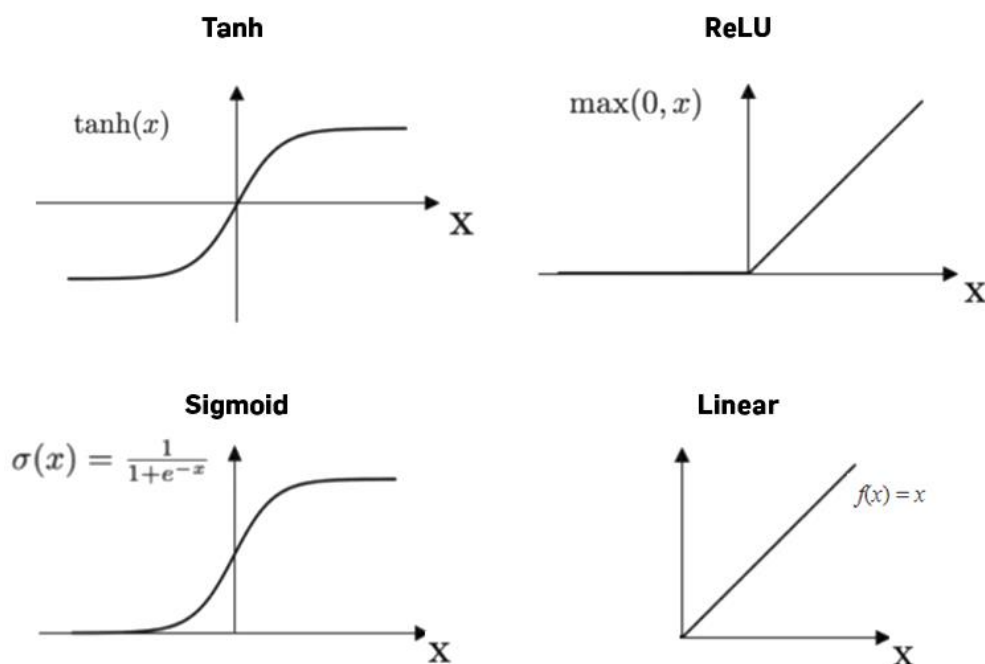


Рисунок 2.15 – Графіки різних функцій активації

Після прямого поширення кожен нейронний шар видає вихідний вектор \hat{y} , який передається наступному шару в якості вхідного вектора, і цей процес продовжується доки мережа не досягне свого останнього шару. Таким чином, на останньому шарі мережі ми отримуємо вихідний вектор всієї мережі. Під час первинного процесу прямого поширення, тобто першого проходження через мережу, ми випадковим чином визначаємо ваги з'єднань між нейронами та зсуви. Ці значення розглядаються як параметри алгоритму нейронної мережі. Після цього ми повинні послідовно виконати підбір цих параметрів відповідно до набору даних, або в загальному випадку це можна назвати "постановкою задачі". Це налаштування ваг і зміщень здійснюється за допомогою іншого виду алгоритму, який називається зворотним поширенням.

Зворотне поширення — це процес «навчання» зміни значень активації для кожного попереднього нейрона у кожному шарі мережі шляхом коригування значень ваг та зсувів.

На виході мережі ми маємо вектор \hat{y} , який, складається із значень, кожне з яких відповідатиме одному з класів, за якими ми класифікуємо. Для кожного отриманого результату, отриманому на одиниці вхідних даних, ми розраховуємо похибку результату мережі по відношенню до очікуваного результату.

Основною метою навчання є зведення сумарної похибки до мінімуму шляхом налаштування вагових коефіцієнтів та зсувів у побудованій мережі. Для того, щоб побудувати математичну модель, потрібно визначити вартісну функцію. Така функція відображає набір параметрів на скалярну величину, що означає, наскільки добре ці значення відповідають очікуваним результатам. Якщо отриманий результат від нейронної мережі відповідає бажаному, або є наближеним до нього, така функція поверне менше число. Якщо ж ми отримали результат, що не є наближеним до очікуваного, то функція поверне, відповідно, більше число. Математичним представленням функції похибки, найчастіше є середньоквадратичне відхилення результуючих значень \vec{y} від очікуваних значень \hat{y} . Формула для отримання функції похибки одного значення виглядає наступним чином:

$$C_n = (y_n - \hat{y}_n)^2, \quad (2.26)$$

де y_n — значення, отримане мережею,

\hat{y}_n — очікуване значення.

З наведеного вище, можна сформулювати розуміння того, що вартісна функція однозначно пов'язує загальну величину похибки, яку несе модель, з набором її параметрів, якими є ваги та зсуви.

Функція втрат розраховується для всього навчального набору даних задіяних у навчальній ітерації, а її середнє значення називається вартісною функцією, яка розраховується як середнє значення похибок отриманих на поточному наборі даних довжиною K .

$$C = \frac{1}{K} \sum_{n=0}^K C_n \quad (2.27)$$

Процес навчання полягає у зміні значень параметрів ваг, представлених \vec{W} та b (зміщення) таким чином, щоб мінімізувати функцію втрат. З набору даних береться кожна точка, поширюється вперед і отримується втрата. Інформація про помилку поширюється назад, у зворотному напрямку через мережу, так що вона може змінити параметри.

Першою задачею є розуміння, яким чином значення вартісної функції залежить від значень ваги та зсуву. Для цього, за допомогою правила ланцюга ми представимо залежність значення вартісної функції від ваги.

$$\frac{\partial C}{\partial W} = \frac{\partial \hat{y}}{\partial Z} \cdot \frac{\partial Z}{\partial W} \cdot \frac{\partial C}{\partial \hat{y}} \quad (2.28)$$

Аналогічно для зсуву матимемо:

$$\frac{\partial C}{\partial b} = \frac{\partial \hat{y}}{\partial Z} \cdot \frac{\partial C}{\partial \hat{y}} \quad (2.29)$$

Перед нами все ще постає завдання знайти оптимальні параметри, за яких функція втрат набуватиме найменшого скалярного значення. Вирішення цього завдання найчастіше покладається на алгоритм оптимізації, що називається градієнтним спуском.

Ідея полягає в тому, що ми змінюємо значення відповідних ваг та зсувів, засновуючись на пропорційному від'ємному значенню градієнта вартісної функції по відношенню до відповідної ваги або зсуву. Зворотне поширення та градієнтний спуск повторюються до тих пір, поки не буде досягнута збіжність.

$$W = W - \left(\alpha \times \frac{\partial C}{\partial W} \right), \quad (2.30)$$

$$b = b - \left(\alpha \times \frac{\partial C}{\partial b} \right) \quad (2.31)$$

де α — швидкість навчання, яка підбирається експериментально.

Нажаль, обчислювальна складність градієнтного спуску є завеликою для його використання на великих обсягах даних, тож існує інших підхід для його використання.

Стохастичний градієнтний спуск передбачає розділення вхідних тренувальних даних для нейронної мережі на менші порції даних, перед цим випадковим чином перемішуючи увесь набір тренувальних даних. Далі, для кожного окремого набору даних вираховується градієнтний спуск вартісної функції, що значно пришвидшує пошук локального мінімуму.

2.7 Згорткові нейронні мережі (CNN)

Як було розглянуто раніше, одними з найбільш інформативних ознак для класифікації звуків є спектрограми, що є графічними представленнями різних

звукових сигналів. Базуючись на цих знаннях, ми можемо розглядати дану задачу як задачу класифікації «зображень» різних звукових файлів. Через це, буде доцільно використовувати спеціальні нейронні мережі які орієнтовані на розпізнавання зображень. Одним із популярних видів таких мереж, є згорткові нейронні мережі.

Згорткові нейронні мережі, також відомі як CNN — це клас нейронних мереж, який спеціалізується на обробці даних, що мають сіткоподібну топологію, таких як зображення.

Цифрове зображення — це двійкове представлення візуальних даних. Воно містить серію пікселів, розташованих у вигляді сітки, яка містить значення пікселів для позначення того, наскільки яскравим повинен бути кожен піксель. На рисунку 2.16 ми бачимо RGB-зображення, яке розділене трьома кольоровими площинами — червоною, зеленою та синьою. Існує ціла низка таких кольорних просторів, в яких існують зображення — чорно-біле, RGB, HSV, CMYK тощо. Тож можна уявити, наскільки складними стануть обчислення, коли зображення досягнуть роздільної здатності, скажімо, 8K (7680×4320px). Завдання згорткової нейронної мережі полягає в тому, щоб зменшити зображення до такого вигляду, який буде легше опрацювати, не втрачаючи при цьому ознак, які є критично важливими для отримання якісних моделей прогнозування. Особливо важливим це стає, коли ми маємо побудувати систему класифікації, яка не лише добре розпізнає певні ознаки, але й може бути масштабована до великих масивів даних.

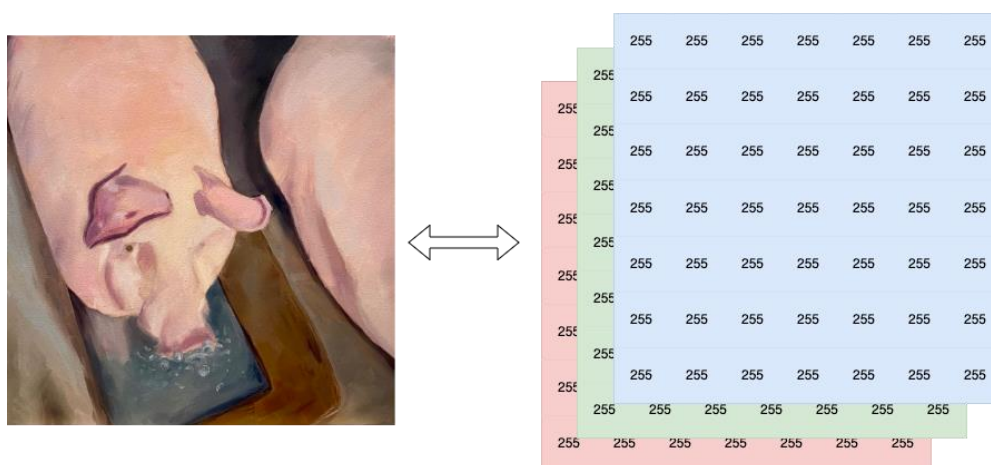


Рисунок 2.16 – Представлення зображення набором матриць

Людський мозок обробляє величезну кількість інформації в ту секунду, коли ми бачимо зображення. Кожен нейрон працює у своєму власному полі сприйняття і пов'язаний з іншими нейронами таким чином, що вони охоплюють все поле зору. Подібно до того, як кожен нейрон реагує на стимули тільки в обмеженій області зорового поля, яка називається рецептивним полем в біологічній системі зору, кожен нейрон в CNN також обробляє дані тільки в своєму власному рецептивному полі. Шари розташовані таким чином, що спершу вони виявляють простіші образи, такі як прямі чи криві лінії, а в наступних шарах — все більш складні поєднання попередніх примітивів, як наприклад, геометричні фігури складені з них. На малюнку 2.17 представлено згорткову нейронну мережу із двома шарами згортки.

Як правило, згорткові нейронні мережі, мають три основних складники:

- шари згортки;
- шари об;
- повноз'єднані шари.

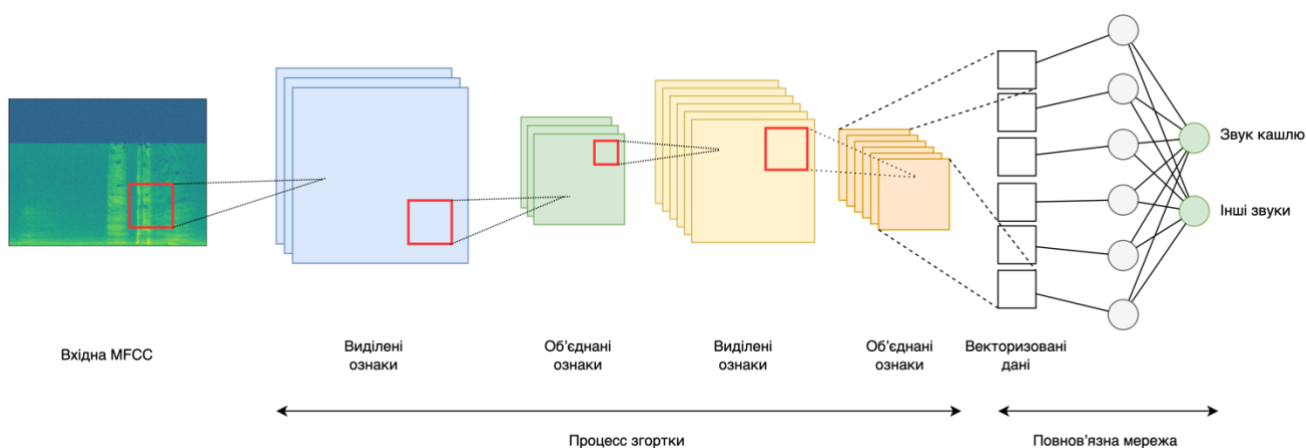


Рисунок 2.17 – Структура згорткової нейронної мережі

Шар згортки є основним структурним елементом CNN. На нього припадає основна частина обчислювального навантаження всієї мережі. Параметри шару складаються з набору згорткових ядер, які мають невелике рецептивне поле, але простягаються на всю глибину вхідного сигналу. Під час прямого поширення кожен фільтр згортається по ширині і висоті вхідного об'єму, обчислюючи скалярний

добуток між фільтрами і вхідним даними, створюючи двовимірну карту активації цього фільтра. В результаті мережа вивчає фільтри, які активуються, коли вона виявляє певний тип ознаки в певному просторовому положенні на вхідних даних.

Згорткове ядро — матриця певних значень, необхідна для виконання операції скалярного добутку зі рецептивним полем, аналогічної розмірності, що належить вхідному зображенню.

Крок ядра — це параметр згорткового ядра нейронної мережі, який впливає на величину зсуву ядра по зображенню. Наприклад, за значення кроку 1, він буде виконувати зсув згорткового ядра по одному пікселю за раз.

На рисунку 2.18 зображено процес накладання згорткового ядра розмірністю 3x3 на зображення із заданим кроком 1. Для виконання операції, із зображення виділяється рецептивне поле, що є однаковим за розмірністю із згортковим ядром. Над отриманими двома матрицями значень виконується операція знаходження скалярного добутку і отримане значення переноситься на нову матрицю ознак.

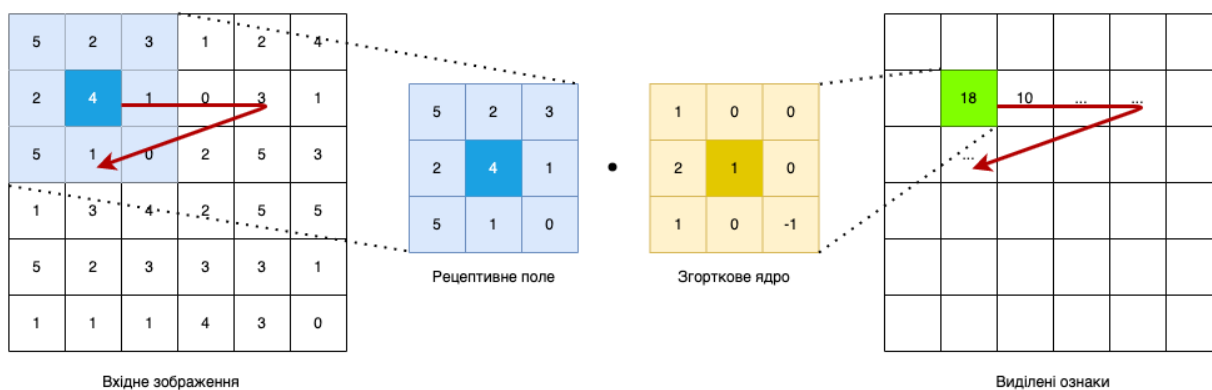


Рисунок 2.18 – Виконання етапу згортки

Але, не важко помітити, що при виконанні операції таким чином, ми отримаємо певну незаповнену ділянку по контуру нашої матриці. Це пов'язано із тим, що використання скалярного добутку неможливо при зміщенні ядра так, щоб його площа виходила за межі зображення. Звісно, можливо знехтувати цим недоліком і використовувати вихідну матрицю меншої розмірності, проте таким чином, ми по суті втрачаємо додаткову інформацію, що могла б поліпшити точність класифікації.

Для того, щоб вирішити цю проблему, до вхідного зображення, перед початком накладання згорткових ядер, додається спеціальний нульовий відступ.

Нульовий відступ — ніщо інше як набір нульових значень «по контуру» нашого зображення. Розміри нульового відступу можуть бути просто обчислені наступним чином:

$$X\text{Padding} = \frac{N - 1}{2}, \quad (2.32)$$

$$Y\text{Padding} = \frac{M - 1}{2}, \quad (2.33)$$

де N — горизонтальна розмірність згорткового ядра;

M — вертикальна розмірність ядра.

Варто зауважити, що розмірності ядра обов'язково мають бути непарними числами, для того, щоб можна було центрувати згорткове ядро відносно рецепторного поля.

Складання мап активації для всіх фільтрів вздовж глибинного виміру утворює повний вихідний набір даних шару згортки. Таким чином, кожен запис у вихідному наборі можна також інтерпретувати як вихід нейрона, який розглядає невелику область на вході і ділиться параметрами з нейронами на тій самій карті активації.

Шар об'єднання. Подібно до шару згортки, шар об'єднання відповідає за зменшення просторового розміру згорнутої функції. Це робиться для того, щоб зменшити складність обчислень, необхідних для обробки даних за рахунок зменшення їх розмірності. Крім того, він корисний для виділення домінуючих ознак, які є інваріантними щодо обертання та положення, таким чином підтримуючи процес ефективного навчання моделі.

Розрізняють два типи об'єднання: максимізуюче (Max Pooling) та усереднююче (Average Pooling) (рис. 2.19). Максимізація повертає максимальне значення з тієї частини зображення, яка покривається ядром. Натомість усереднення повертає середнє арифметичне всіх значень з частини зображення, що покривається ядром.

Max Pooling також виконує функцію зменшення шумів. Воно повністю відкидає усі шумні активації, між тим, виконуючи функцію зменшення шумів разом зі зменшенням розмірності. В свою чергу, усереднення просто виконує задачу зменшення розмірності. Таким чином, можна зробити висновок, що Max Pooling є більш універсальним, у порівнянні із Average Pooling. Різниця різних об'єднань проілюстрована на рисунку 2.20.

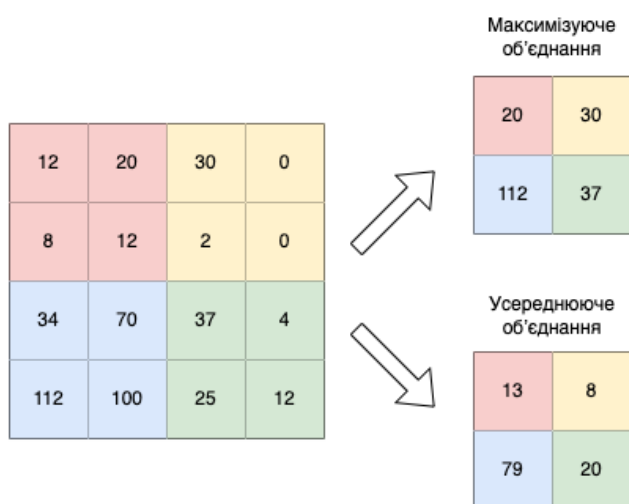


Рисунок 2.19 – Принцип застосування максимізуючого та усереднюючого об'єднань

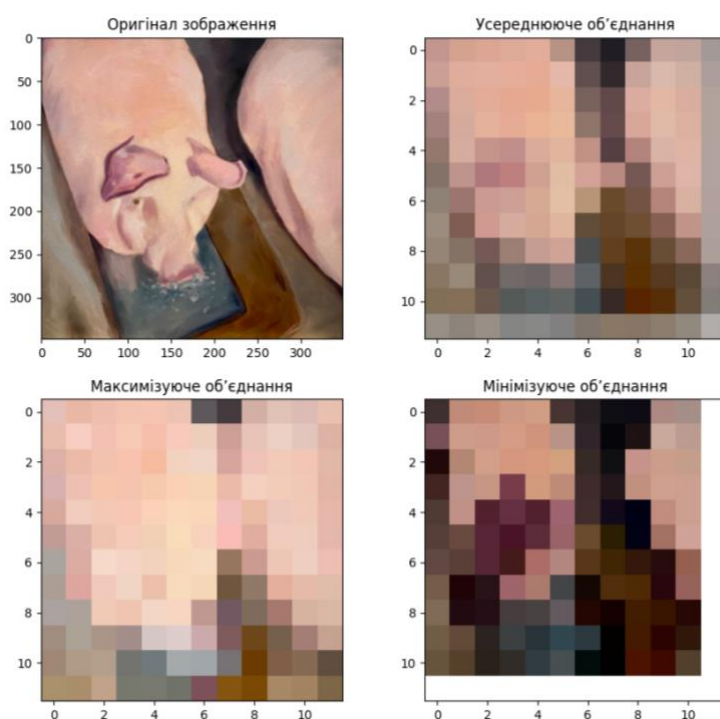


Рисунок 2.20 – Застосування різних об'єднань на зображенні

Згортковий шар і шар об'єднання разом утворюють i -тий шар згорткової нейронної мережі. Залежно від складності зображень, кількість таких шарів може бути збільшена для ще більшого захоплення низькорівневих ознак, проте ціною більших обчислювальних ресурсів.

Повнозв'язний шар є, зазвичай, досить швидким способом розпізнавання нелінійних комбінацій високорівневих ознак, представлених на виході згорткового шару. Після того, як наше вхідне зображення було перетворено у відповідну структуру для застосування у багаторівневому перцептроні, нам необхідно розкласти зображення у вектор-стовпчик. Цей вектор подається до нейронної мережі прямого поширення, і до кожної ітерації навчання застосовується зворотне поширення. За ряд епох модель здатна розрізняти домінуючі та певні низькорівневі ознаки на зображеннях та класифікувати їх за допомогою Softmax методу.

Softmax класифікатор — це такий вид класифікації, за якого вид вихідної вартісної функції відносно кожного класу змінюється із числового коефіцієнту відмінності отриманих даних від очікуваних на ймовірнісний, за якого ми отримуємо значення від 0 до 1. Таким чином людина може більш органічно інтерпретувати вихід нейронної мережі. Інакше кажучи на виході ми отримуємо ймовірнісний вектор відповідності вхідних даних до кожного із класів.

Класифікатор Softmax є узагальненням бінарної форми логістичної регресії. Так само, як і у випадку вартісної функції, наша функція відображення визначена таким чином, що вона приймає вхідний набір даних X і відображає їх на вихідні значення класів за допомогою лінійного скалярного добутку даних X і вагової матриці W .

2.8 Математична постановка задачі

На основі поставленої задачі сформулюємо цільову функцію. Для доволно відсортованої вибірки спектрограм звуків навколишнього середовища маємо сформулювати вартісну функцію, яку необхідно мінімізувати.

Цільова функція розв'язання задачі:

$$L = -\frac{1}{N} \sum_{n=0}^N \log p_n \rightarrow \min, \quad (2.34)$$

де c_k — клас класифікації (чи відповідає ознака кашлю чи ні);

X_n — MFCC ознака зафіксованого звуку;

N — кількість ознак;

p_n — ймовірність приналежності MFCC X_n до класу c_{X_n} .

Висновки до розділу

В рамках розділу, ми сфокусувалися на обробці звукових хвиль та методі дискретизації для перетворення їх у цифрові сигнали. Розглянуті основні параметри, що характеризують звукові. В рамках огляду обробки сигналів, виявлено, що для отримання інформації щодо частотного домену сигналу важливу роль відіграють перетворення Фур'є, що якраз дозволяють виділити частотні характеристики сигналу.

Розглянуто загальний алгоритм вирішення задачі класифікації звукових сигналів і сформовано уяву про його конкретні кроки.

Перед переходом до класифікації звукових сигналів, важливо було виявити які саме ознаки, що характеризують звукові сигнали певних класів існують та яку і скільки інформації вони можуть зберігати для подальшої передачі у класифікатор. Серед найчастіше вживаних у задачах класифікації звуків, до яких і відноситься звук кашлю, найбільш інформативними є такі, що надають інформацію про частотні зміни у часі — спектрограми, мел-спектрограми, мел-частотні кепстральні коефіцієнти. Було вирішено використовувати методи класифікації, що вже зарекомендували себе у розпізнаванні зображень.

Розглянуто основні принципи побудови штучних нейронних мереж, їх математичний опис, а також принципи побудови і роботи специфічних згорткових нейронних мереж, що направлені спеціально на розпізнавання зображень.

В результаті було сформовано математичну постановку задачі за умови використання згорткових нейронних мереж що на вході отримують мел-частотні кепстральні коефіцієнти і визначають їх приналежність до певного класу.

3 АРХІТЕКТУРНІ РІШЕННЯ СИСТЕМИ

Система що передбачається, має на меті надавати сільськогосподарським підприємствам актуальну інформацію щодо поточного стану і показників на об'єкті підприємства. З огляду на поставлені задачі, які необхідно вирішити для повноти реалізації системи.

Органічно буде поділити систему на декілька частини, що виконуватимуть кожен свою функцію. Такими частинами є веб-застосунок, як точка входу користувача в систему, та модуль збору даних, що знаходиться на підприємстві. Схематично зобразити це можна як показано на рисунку 3.1.



Рисунок 3.1 — Спрощене представлення модулів системи

На рисунку ми бачимо розділення системи на три ключові складові:

- пристрій збору даних;
- клієнтський веб-застосунок (front-end);
- серверну частину (back-end).

Пристрій збору даних — модуль, що складається із спеціального технічного забезпечення та відповідного програмного забезпечення, що дозволяє фіксувати дані, які надходять від датчиків та міститиме вбудований класифікатор на основі

нейронної мережі для ідентифікації кашлю записаного мікрофоном. Він комунікуватиме із віддаленим сервером.

Веб-застосунок доречно було розділити на окрему презентаційну клієнтську частину та шар із бізнес-логікою та обробкою даних. Такий підхід повністю відповідає клієнт-сервісній архітектурі систем. Завдяки такому розділенню, ми не будемо в майбутньому залежні від використання тільки клієнтського веб-застосунку, а матимемо можливість створити наприклад мобільні застосунки, або розробити інтеграції із сторонніми сервісами, системами автоматизації тощо.

Серверна частина відповідає за усю подальшу обробку даних користувачів, пристроїв та реалізовує усі необхідні бізнес-процеси покладені в основу інформаційної системи.

Веб-застосунок відіграє роль основного інтерфейсу із яким користувач буде взаємодіяти із інформаційною системою.

Детальна структурна схема наведена у додатку Г.

3.1 Пристрій збору даних

Передбачається, що система автоматично відслідковує параметрів мікроклімату приміщення де утримуються тварини та фіксуватиме виявлення звуків кашлю у тварин. Основні модулі, що мають у своїй сукупності забезпечити функціонування пристрою, це модулі збору та обробки даних та модуль комунікації із серверами інформаційної системи.

На рисунку 3.2 представлено схему взаємодії модулів пристрою із back-end частиною системи. Складовими є:

- сенсорні модулі аналізу повітря, температури та вологості, і модуль фіксації сигналів з мікрофону, що забезпечують надходження даних спостереження;
- класифікатор звуків кашлю, що передбачає обробку даних зі звукового модулю;
- модуль для обробки даних отриманих від попередніх двох;
- мережевий клієнт для спілкування із сервером.

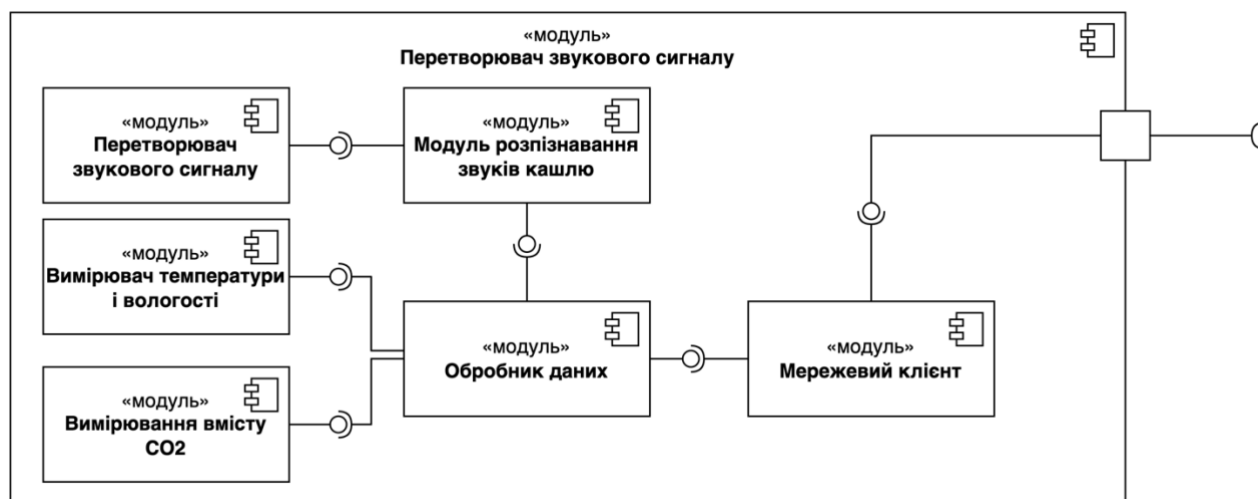


Рисунок 3.2 — Схема компонентів пристрою збору даних

3.2 Серверна частина

Серверна частина застосунку, побудована з огляду на архітектуру мікросервісів, що спілкуються між собою за шаблоном проєктування публікація-підписник (Pub/Sub), коли одні сервіси відправляють повідомлення, а інші очікують їх та обробляють отримуючи із черги повідомлень.

В структурі компонентів серверної частини, що зображено на рисунку 3.3, можна відзначити наступні підмодулі:

- модуль API частина, що відповідає за обробку запитів від користувача;
- сервіс обробки даних отриманих від пристроїв;
- сервіс авторизації, що окремо здійснюватиме перевірку прав доступу користувачів;
- сервіс роботи із повідомленнями;
- сервіс відправки сповіщень, який відповідає за відправку push-сповіщень на фізичні пристрої.

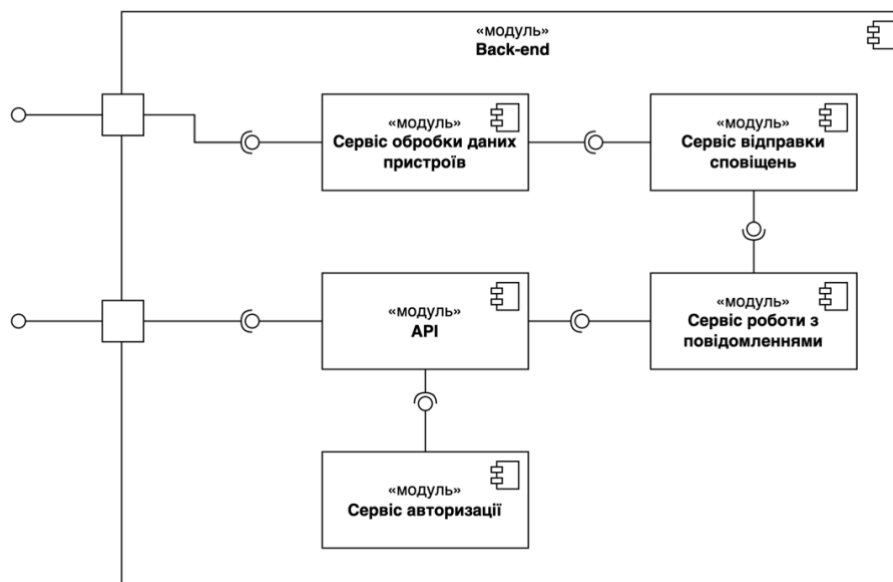


Рисунок 3.3 — Схема компонентів серверної частини

3.3 Клієнтський веб-застосунок

Основним інтерфейсом користувача виступає односторінковий веб-застосунок, що спілкується із серверною частиною через API запити. Користувацький інтерфейс можна розділити на дві окремі частини, що відповідатимуть інтерфейсам різних ролей, в нашому випадку це інтерфейс для звичайного користувача та інтерфейс для адміністраторів. Крім того, що таким чином ми розрізняємо сфери використання, даючи користувачу чітко зрозуміти у якій ролі він зараз користується системою, ми також оптимізуємо кількість даних, що матиме завантажувати користувач. Структурна схема усіх складових наведена на рисунку 3.4.

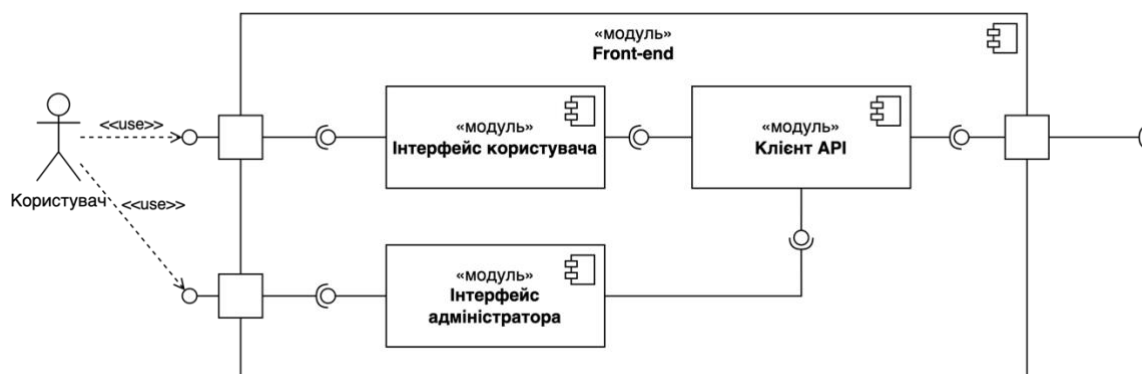


Рисунок 3.4 — Схема компонентів клієнтського застосунку

4 ER-ДІАГРАМА

Інформаційна система що розробляється вимагає постійного доступу до даних. Для цього, на основі попередньо окреслених моделей представлених у розділі 1.4, було розроблено структуру бази даних, яка зможе використовуватися для збереження даних у реляційній СУБД, а також зберігатиме зв'язки між суб'єктами автоматизації бізнес-процесів описаних у розділі 1.

Створена модель бази даних наведена у додатку Д.

Кожна сутність була відображена у таблицях бази даних максимально намагаючись дотримуватися принципу побудови реляційних баз даних із використанням першої нормальної форми.

Примітка — Відношення знаходиться в першій нормальній формі (ПНФ), якщо сфери використання всіх його атрибутів атомарні, а значенням будь-якого атрибуту в кортежі, є єдиними значення з його сфери використання. Еквівалентним визначенням є те, що кожен атрибут є неподільним і функціонально залежним від ключа. Значення, які є складеними, або декілька значень для атрибута, не допускаються. Ця вимога фактично є частиною визначення відношення. Відношення, яке порушує це правило, називається ненормованим.

4.1 Опис сутностей представлених у базі даних

Докладно розглянемо найважливіші із сутностей системи представлених у базі.

User — сутність для збереження даних користувача, імені, електронної пошти, паролю, тощо. Докладний опис полів, необхідних для опису сутності, наведено у таблиці 4.1. Для налаштувань користувача було використано додаткову сутність UserSettings що покликана зменшити кількість полів при використанні операції об'єднання із іншими зв'язаними із користувачем сутностями, її поля представлені таблицею 4.2.

Таблиця 4.1 — Опис полів таблиці User

Назва поля	Опис поля	Тип даних	Властивості	
			PK	FK
id	Унікальний ідентифікатор користувача	UUID	+	
first_name	Ім'я користувача	VARCHAR(100)		
last_name	Прізвище користувача	VARCHAR(100)		
email	Електронна пошта користувача, що також буде використана в якості логіну	VARCHAR(255)		
password	Пароль користувача у хешованому за допомогою SHA256 виді з додаванням солі	VARCHAR(255)		
country	Країна користувача у форматі ISO 3166-1 для ідентифікації найближчих серверів	VARCHAR(2)		

Таблиця 4.2 — Опис полів таблиці UserSettings

Назва поля	Опис поля	Тип даних	Властивості	
			PK	FK
id	Унікальний ідентифікатор налаштувань	UUID	+	
user_id	Ідентифікатор користувача	UUID		+

Назва поля	Опис поля	Тип даних	Властивості	
			PK	FK
language	Мова встановлена у користувача, зберігається значення за стандартом ISO 639	VARCHAR(2)		
timezone	Часовий пояс користувача у форматі Europe/Kyiv	VARCHAR(100)		
temperature	Варіант представлення температури у °C або °F	ENUM(C, F)		
push_alert	Параметр що відповідає за налаштування відправки push-повідомлень	BOOLEAN		
push_notification	Параметр що відповідає за налаштування відправки push-повідомлень	BOOLEAN		

Permission — представлення перелічувального типу даних, що зберігає інформацію про види доступу у користувачів. Складається назви та значення одного права доступу, що відповідає бінарній масці прав доступу.

Role — об'єднання прав доступу у категорії користувачів, потрібне для гнучкого надання прав доступу різним категоріям користувачів. Саме тому має множинний зв'язок із таблицею користувачів.

Company — сутність із інформацією про конкретне підприємство, що автоматизується. Створюється окремо за запитом супер-адміністратора при заведенні у систему нового клієнта. Опис полів сутності наведений у таблиці 4.3.

Таблиця 4.3 — Опис полів таблиці Company

Назва поля	Опис поля	Тип даних	Властивості	
			PK	FK
id	Унікальний ідентифікатор підприємства	UUID	+	
name	Назва підприємства	VARCHAR(255)		

Site — сутність для репрезентації одного конкретного об'єкта підприємства. Несе в собі інформацію щодо назви, локації (географічних координат) для відображення на карті в панелі керування, країни місцезнаходження, видів тварин що утримуються на об'єкті, належить до певного підприємства. Назви полів та типи даних представлені в таблиці 4.4.

Таблиця 4.4 — Опис полів таблиці Site

Назва поля	Опис поля	Тип даних	Властивості	
			PK	FK
id	Унікальний ідентифікатор ферми	UUID	+	
name	Назва ферми	VARCHAR(255)		

Назва поля	Опис поля	Тип даних	Властивості	
			PK	FK
country	Країна у форматі ISO 3166-1 для ідентифікації найближчих серверів	VARCHAR(2)		
latitude	Координата широти місцезнаходження підприємства	DECIMAL(10,6)		
longitude	Координата довготи місцезнаходження підприємства	DECIMAL(10,6)		
product_type	Вид тварин які знаходяться на підприємстві	ENUM(cattle, pig)		

SiteGroup — об'єднання ферм для зручного групування об'єктів підприємства для їх фільтрації. З того, що варто відмітити, що дана сутність має тип з'єднання багато до багатьох із сутністю Site та має лише два поля: id (UUID) та name (VARCHAR(255)).

Airspace — це загальна репрезентація абстрактного поняття простору у якому здійснюються спостереження чи встановлені датчики. Дана сутність передбачає ієрархічну структуру, тож було використано циклічне посилання виду «сам на себе» для можливості об'єднання структур у комбінації. Опис властивостей наведено у таблиці 4.5.

Таблиця 4.5 — Опис полів таблиці Site

Назва поля	Опис поля	Тип даних	Властивості	
			PK	FK
id	Унікальний ідентифікатор ферми	UUID	+	
parent_id	Відповідний ідентифікатор батьківського Airspace	UUID		+
name	Назва простору для ідентифікації. Наприклад: Корпус 1, поверх 3	VARCHAR(255)		
type	Вид простору	ENUM(house, floor, room)		

Device — сутність, що описує фізичний пристрій для спостережень, його назву, серійний номер та положення у Airspace. Поля сутності описуються у таблиці 4.6.

Таблиця 4.6 — Опис полів таблиці Device

Назва поля	Опис поля	Тип даних	Властивості	
			PK	FK
id	Унікальний ідентифікатор запису про пристрій	UUID	+	
airspace_id	Відповідний ідентифікатор Airspace в	UUID		+

Назва поля	Опис поля	Тип даних	Властивості	
			PK	FK
	якому встановлено пристрій			
serial	Серійний номер пристрою	VARCHAR(128)		
name	Назва пристрою для ідентифікації	VARCHAR(255)		

Entry — сутність для запису вимірювань і показників що збирається пристроєм. Містить тип, для ідентифікації сенсора чи значення отриманого від нейронної мережі та саме значення. Поля необхідні для опису сутності представлені у таблиці 4.7.

Таблиця 4.7 — Опис полів таблиці Device

Назва поля	Опис поля	Тип даних	Властивості	
			PK	FK
id	Унікальний ідентифікатор запису про пристрій	UUID	+	
device_id	Відповідний ідентифікатор пристрою	UUID		+
type	Тип значення	ENUM(temperature, humidity, pressure, co2, cough)		
name	Назва пристрою для ідентифікації	VARCHAR(255)		

Batch — сукупність записів приладів за одним типом, що відповідає деякому проміжку часу. Потрібна для відстеження змін і продукування попереджень.

Alarm — сутність для опису відхилення, що було зафіксоване протягом певного періоду спостереження Batch.

Notification — таблиця необхідна для сервісної інформації щодо відправлених push-повідомлень до користувачів.

Note — сутність для ведення записів стосовно об'єктивних спостережень працівниками у рамках одного об'єкта підприємства.

Chat — сутність для опису процесу обміну повідомленнями між користувачами.

Message та Attachment — дві сутності необхідні для опису повідомлень користувачів. Завдяки ним, можливо отримувати повідомлення та прикріпляти до них файли.

5 ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

У даному будуть розглянуті технічні особливості реалізації інформаційної системи з виявлення захворювань худоби. Створене програмне забезпечення можна поділити на три складові:

- модуль розпізнавання кашлю в складі пристрою збору даних,
- серверна частина,
- користувацький веб-застосунок.

5.1 Вимоги до програмного продукту

Здійснивши огляд існуючих рішень, умов роботи підприємств для яких призначена дана система, аналогічних великих інформаційних систем з інших сфер, було сформовано переліки вимог до кожного з компонентів системи.

5.1.1 Вимоги до програмного модулю розпізнавання кашлю

Головною вимогою для модуля розпізнавання кашлю є його автономність, тобто можливість бути інтегрованим до інших програмних модулів. Також має бути передбачена можливість швидкого виконання алгоритму розпізнавання на низькопродуктивних обчислювальних пристроях. Програмний модуль також повинен бути кросплатформним і здатним компілюватися і запускатися на вбудованих системах із різною архітектурою центральних процесорів.

5.1.2 Вимоги до серверної частини

Для серверної частини було сформовано наступний перелік нефункціональних вимог:

- система повинна запобігати доступу до даних неавторизованими або користувачами без відповідного доступу;

- спроможність підтримувати велику кількість вхідних з'єднань із сторони клієнтських веб-застосунків та пристроїв збору даних;
- швидкодія обробки запитів системою;
- система має автоматично відновлюватися після збоїв в процесі використання;
- система має бути стійкою до помилок введення зі сторони користувача.

5.1.3 Вимоги до клієнтського застосунку

Серед нефункціональних вимог до клієнтського веб-застосунку потрібно виділити:

- простота та очевидність інтерфейсу при використанні користувачем;
- швидкодія інтерфейсу без потреби у великих обчислювальних ресурсах;
- швидкість завантаження файлів інтерфейсу на пристрої користувачів
- можливість роботи у старих веб-браузерах, що не підтримують останні стандарти мов програмування.

5.2 Засоби розробки

5.2.1 Засоби розробки модулю розпізнавання кашлю

Для реалізації алгоритму виявлення та розпізнавання звуків кашлю було обрано мову програмування Python, яка набрала своєї популярності серед дослідників для написання реалізацій різного типу алгоритмів машинного навчання. За використання стандартного оточення, мова є доволі простою у засвоєнні. За роки існування, було створену широку кількість різноманітних сторонніх бібліотек, які реалізують найчастіше затребувані математичні операції, алгоритми перетворення тощо. Інші мови, не мають такої підтримки з боку наукового співтовариства, тож вибір Python в якості мови програмування дозволяє пришвидшити розробку алгоритмів машинного навчання.

Серед існуючих бібліотек, що дозволяють створювати моделі нейронних мереж вибір зупинився на бібліотеці TensorFlow [28] завдяки наявності простих готових рішень для розробки. Для реалізації саме штучної нейронної мережі використовується включений до складу TensorFlow пакет Keras [16], що дає мод. Навчання було зроблено за допомогою Scikit-learn [24].

При розробці було використано наступні бібліотеки:

- NumPy;
- TensorFlow;
- TensorFlow Lite;
- Scikit-learn;
- Keras.

Завдяки використанню TensorFlow Lite, ми покриваємо можливість запуску попередньо тренованої нейронної мережі на низько-потужних пристроях як наприклад тих, що використовують мікроконтролер ESP32.

Для перевірки на забезпечення розробки було використано засіб розробки для інтегрованих систем — PlatformIO, що допомагає у перевірці можливості компіляції під різні пристрої.

Після того як нейронну мережу було треновано, ми можемо експортувати її коефіцієнти та використати у програмному забезпеченні пристрою. Етапи розробки і перенесення моделі з використанням TensorFlow Lite наведено на рисунку 5.1.

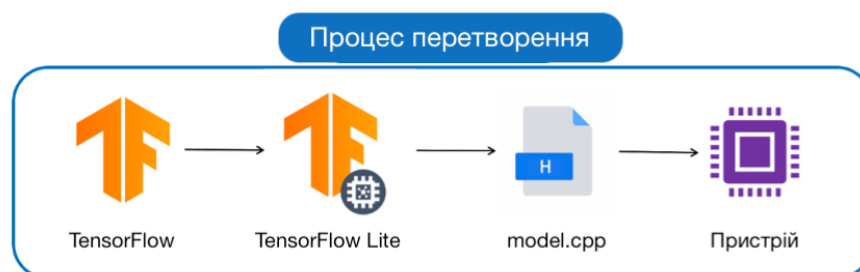


Рисунок 5.1 — Процес перенесення моделі на пристрій

Загальний процес роботи модулю розпізнавання кашлю наведено у додатку Е.

5.2.2 Засоби розробки серверної частини

Серверна частина інформаційної системи написана із застосуванням мови програмування Golang [п], через те що вона є швидкою, компільованою мовою із С-подібним синтаксисом. Також, Golang забезпечує просте використання в якості мови написання серверних рішень, завдяки вбудованому веб-серверу, що слухає вхідні запити та зручних методів трансформації об'єктів у різні структури даних. Можна сказати, що ця мова поєднує у собі швидкість та надійність С та простоту використання запозичену від Python.

Так як серверна частина передбачає використання веб-застосунком що виконується у браузері, для комунікації із API системи, використовується HTTPs з'єднання через його простоту, універсальність та широку підтримку різноманітними системами, зокрема старими браузерами.

На відміну від API, для реалізації каналу надходження даних від пристроїв збору інформації, було обрано протокол передачі даних MQTT, розроблений спеціально для передачі даних телеметрії від вбудованих система, датчиків тощо. В якості транспортного шару передбачає використання TCP протоколу, для забезпечення двостороннього зв'язку, який гарантуватиме надходження пакетів даних. Також, використання MQTT покликане зменшити розміри даних що передаються через Інтернет-канал ферми, це зменшить навантаження мережі підприємства та, для тих підприємств, що мають нестійкі лінії чи лінії із низькою пропускнуою здатністю. Порівняння розмірів пакетів для HTTP та MQTT наведені у таблиці 5.1.

Таблиця 5.1 — Порівняння обсягів даних HTTP та MQTT

Параметр	MQTT, Байт	HTTP, Байт
Встановлення з'єднання	5572	2261
Розрив з'єднання	376 (optional)	0
Сума для 1 повідомлення	6336	5546

Параметр	MQTT, Байт	HTTP, Байт
Сума для 10 повідомлень	9829	55,460
Сума для 100 повідомлень	44,748	554,600

Додатковою характеристикою, протоколу MQTT у порівнянні із HTML є можливість отримувати статус стосовно наявності підключення певного пристрою без додаткових запитів зі сторони серверу. Ми завжди можемо отримати дані від MQTT брокера стосовно пристроїв що зараз підключені та надсилають повідомлення.

Між внутрішніми компонентами системи передбачається використання двох різновидів комунікації:

- прямого виклику сервісу;
- обмін повідомленнями за моделлю публікація-підписка.

Для прямого виклику, і використання між внутрішніми сервісами системи, вибрано модель обміну даними gRPC [п] замість REST. Це було зроблено завдяки порівнянню ключових параметрів наведених у таблиці 5.2.

API-інтерфейси REST зазвичай побудовані на HTTP 1.1, який використовує модель взаємодії запит-відповідь. Це означає, що коли мікросервіс отримує кілька запитів від більш ніж одного клієнта, він повинен обслуговувати їх по одному, що уповільнює роботу всієї системи. API REST також можуть використовувати HTTP 2, вони, як і раніше, обмежені моделлю запит-відповідь і не використовують підтримку HTTP 2 для двонаправленого потокового зв'язку.

На відміну від цього, gRPC використовує HTTP 2 і користується перевагами підтримки HTTP 2 як для взаємодії з клієнтом, так і для двонаправленого зв'язку. Таким чином, gRPC може керувати "унарними" взаємодіями, що відповідають HTTP 1.1, коли клієнт надсилає один запит, а сервер надсилає одну відповідь. Водночас клієнти можуть також відкривати довготривалі з'єднання, в яких кожен виклик RPC відкриває новий потік HTTP 2, також відомий як двонаправлений.

Також на відміну від REST і RPC, gRPC долає проблеми, пов'язані зі швидкістю і вагою, і пропонує більшу ефективність під час передавання повідомлень,

використовуючи формат обміну повідомленнями Protobuf. Серед характеристик Protobuf можна виділити наступні:

- незалежний від платформи та мови;
- серіалізує і десеріалізує структуровані дані для передачі в бінарному форматі;
- передача даних відбувається швидше, тому що Protobuf зменшує розмір повідомлень і служить легким форматом обміну повідомленнями.

Таблиця 5.2 — Порівняння моделей REST та gRPC

Характеристика	gRPC	REST
Протокол передачі даних	HTTP 2.0	HTTP 1.1
Формат повідомлень	Protobuf (Protocol Buffers)	JSON
Генерація коду	Вбудована	Сторонні рішення
Комунікація	Одностороння або двостороння	Одностороння

5.2.3 Засоби розробки клієнтського застосунку

Великого вибору серед мов програмування та технологія, що можуть працювати у браузері, нажаль не існує. Єдиною доступною для виконання у браузері мовою є і залишається JavaScript [п], а мовами для описання структури та стилів веб-сторінки залишаються HTML та CSS відповідно.

JavaScript — доволі проста, інтерпретована мова програмування для використання у клієнтській та серверній веб-розробці. Мова відноситься до динамічно-типізованих, об'єктно-орієнтованих мов програмування, а також іноді відноситься до С подібних. Головними перевагами є швидкодія, простота використання та відсутність строгої типізації.

Нажаль, динамічна типізація мови створює додаткові можливості для помилок зі сторони розробника під час написання програмного забезпечення. Через це, багатьма розробниками та компаніями було створено свої альтернативні мови, що

слугуватимуть заміною JavaScript під час написання коду, а перед передачею на сервер транслюватимуться у нього.

Тому, для розробки клієнтського веб застосунку було використано мову програмування TypeScript [п], що позиціонується як мова-надмножина над JavaScript та додає додатковий шар абстракції такий як наявність типів, інтерфейсів, пакетів.

Варто також зауважити, що для забезпечення вимоги про можливість застосування програмного забезпечення у старих браузерях, необхідно передбачити додаткову складову — спеціальний компілятор у JavaScript стандарту мови ECMAScript 5.1 [п], який підтримується навіть такими браузерами як Internet Explorer 10. В якості такого компілятора було обрано SWC [п], через значний прирість у швидкодії в порівнянні із класичними варіантами Babel [п], або стандартним TypeScript.

Така сама ситуація і з HTML та CSS, тож виходом із положення є застосування мов що транслюються у стандартний вигляд, або використання тільки JavaScript для маніпуляції над DOM.

Жодна з мов програмування чи інтерпретаторів не передбачає вбудованого розділення вихідного коду на модулі, розділення файлів чи інших інструментів, що передбачають структурування коду. Тож для спрощення роботи над проектом, простішому використанню інших мов для розробки застосунку використовуються спеціальні сервісні програми — пакувальники вихідного коду. Їх застосування значно пришвидшує процес роботи та дає додаткові можливості оптимізації.

В рамках проекту було використано пакувальник Webpack [п], що став своєрідним стандартом в спільноті розробників веб-застосунків. За його допомогою можна вирішити наступні задачі:

- використати систему модулів ESMODULE [п] або CommonJS [п];
- використати спеціальні програми перетворюючи мови в автоматичному режимі під час фінальної зборки проекту;
- автоматизувати процес зменшення коду;
- об'єднання вихідного використаних бібліотек у одне вихідне дерево коду;

- розділення об'єднаного коду для використання технології «лінивого-завантаження»;
- виконання операцій по вставці коду, стилів та сервісних тегів в HTML-файл.

Для пришвидшення розробки користувацького інтерфейсу, і його шару презентації використовується бібліотека ReactJS [п], яка надає в своєму складі набір інструментів для роботи зі станами системи, операціям із DOM та відображенню. ReactJS був створений як засіб, покликаний спростити створення динамічних веб-сторінок без використання серверної складової для генерації HTML документів. Для зручності розробки в рамках бібліотеки ReactJS також надходить спеціальне розширення для синтаксису мови JavaScript — JSX []. Завдяки цьому додається звична складова використання певних тегів для виконання коду компонентів програми.

Важливим моментом при виборі технології було також і те, що для ReactJS існує можливість створення нативних мобільних застосунків для платформ iOS та Android із застосуванням подібної бібліотеки — ReactNative []. Ця технологія додатково дозволяє створювати мобільні застосунки використовуючи JavaScript або інші мови, які в нього транслюються. Ідея ReactNative полягає в створенні додаткових проміжних абстракцій до складових та компонентів, що будуть мапитися на вже існуючих нативних рішень чи бібліотек у iOS чи Android. При цьому залишаючись кросплатформним рішенням. Взаємодія ReactNative частини із ОС пристрою проілюстровано на рисунку 5.2.

Серед цікавих особливостей бібліотеки ReactNative є і те, що вона може бути використана не тільки для розробки мобільного застосунку, а також і просто, як набір компонентів. Щоб зменшити час для майбутньої розробки мобільного застосунку, задачею було знайти спосіб розділення кодової бази між веб та мобільними складовими. Для цього існує додаткова бібліотека react-native-web [п], покликана створити додаткові компоненти, що будуть відображати компоненти бібліотеки ReactNative на HTML відповідники, що вже можуть бути інтерпретовані веб браузером. Таким чином ми зможемо використовувати більшість вже створених

компонентів інтерфейсу (кнопок, таблиць, меню) як у браузері так і в нативному застосунку згодом.

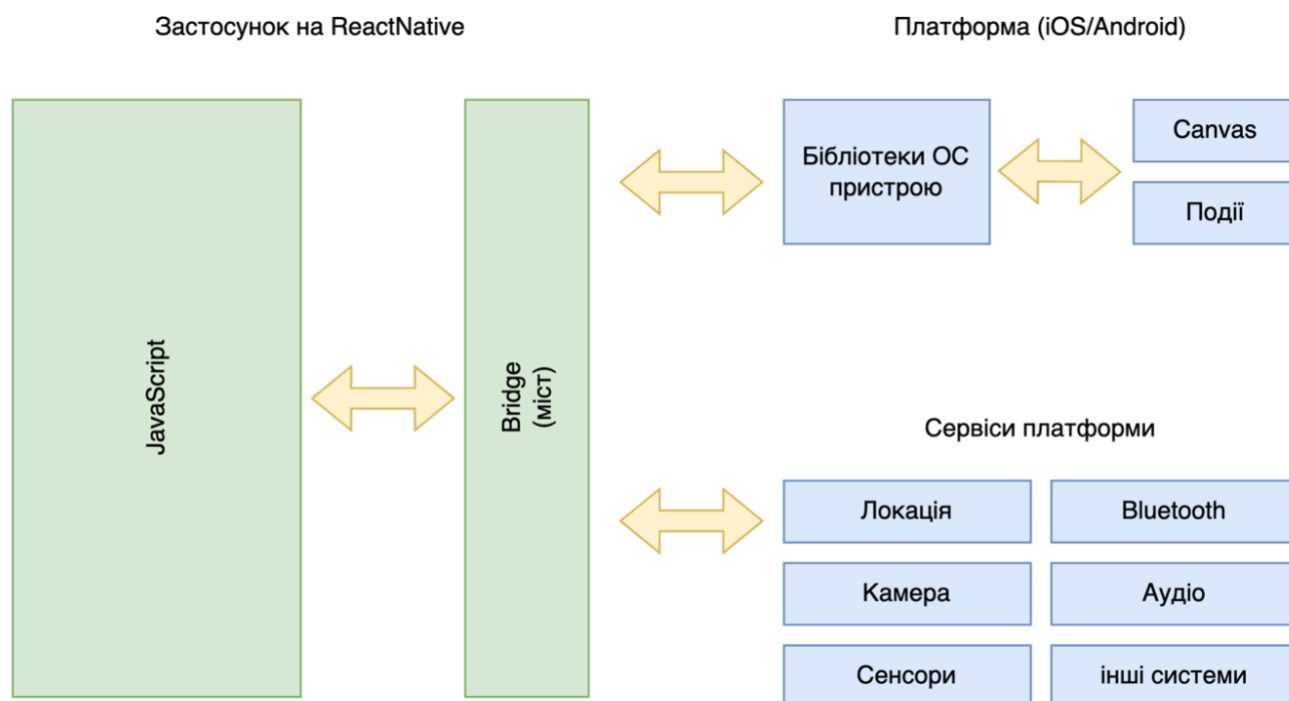


Рисунок 5.2 — Трансформація взаємодія коду застосунку на ReactNative

Наступним кроком, що важливо було вирішити — використання бібліотеки станів, що дозволить структурувати дані, що надходять від серверної частини для подальшого перенесення у презентації. Для цієї задачі використовується бібліотека Redux [п] для управління станами компонент та застосунків загалом. Ідея полягає в тому, що стан застосунку не може бути змінений напряму, тільки через використання спеціальних об'єктів переходу між станами. З точки зору підходу, Redux надає можливість контролювати стан застосунку за допомогою створення скінченних-автоматів. На рисунку 5.3 проілюстрована відмінність використання двостороннього зв'язку між компонентами та низхідного, за допомогою Redux.

Повна послідовність виконання оновлення вигляду користувацького інтерфейсу у відповідь на дії користувача наведена на UML-діаграмі послідовностей у додатку Ж.

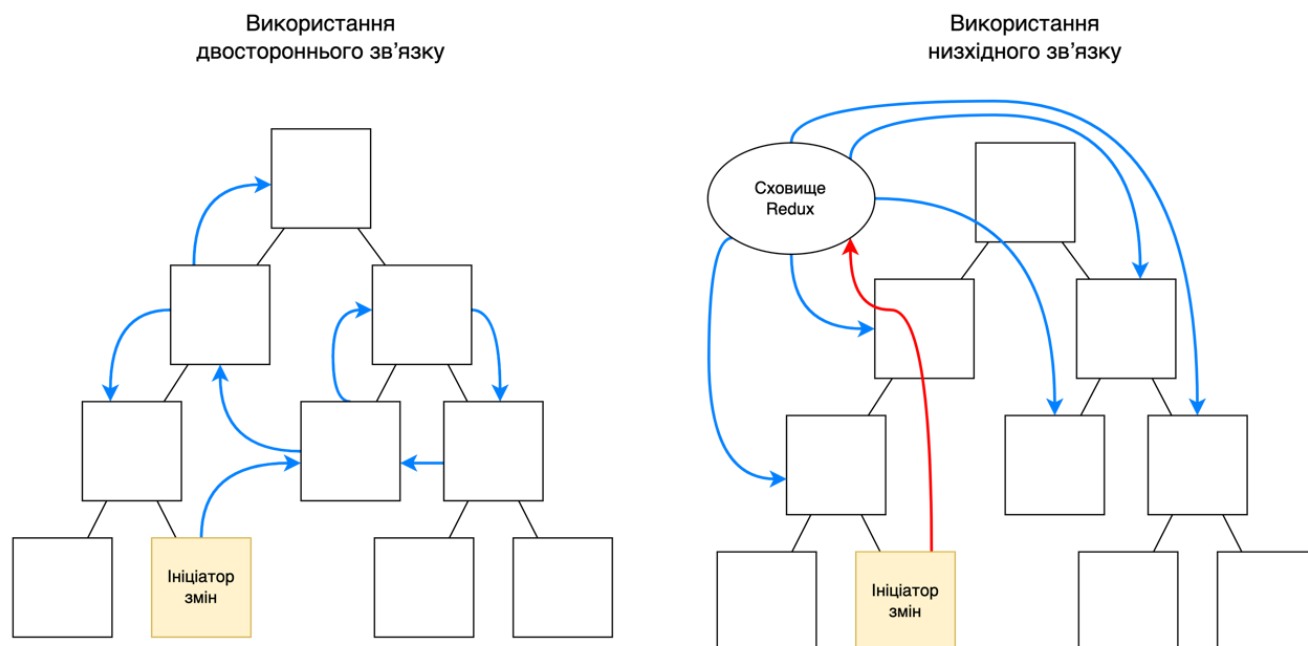


Рисунок 5.3 — Різниця зв'язків між компонентами

5.3 Розгортання програмного забезпечення

Клієнтська складова та серверна складова інформаційної системи розгорнуті із використанням хмарного середовища Google Cloud Platform зі своїми особливостями.

5.3.1 Розгортання серверної частини

Серверна частина розроблена із застосуванням безсерверної архітектури, тож для її розгортання необхідно використати декілька продуктів-складових Google Cloud Platform (GCP):

- сервіс авторизації Firebase Authentication, що створить нам окрему базу даних користувачів для простішої інтеграції із іншими сервісами від GCP;
- сервер MQTT на основі RabbitMQ із налаштуванням автоматичного збільшення кількості екземплярів;
- подієвий рушій Google PubSub для забезпечення спілкування між сервісами;

- хмарний екземпляр PostgreSQL бази даних, як центральна база даних системи;
- сервіси підняття окремих docker контейнерів Cloud Run, що ми використаємо для розгортання окремих сервісів, а саме сервісу сповіщень, повідомлень та API;
- сервіс Cloud Functions для виконання лямбда-функцій, що будуть підписані на події у PubSub;
- API Gateway буде використовуватись як єдиний модуль розподілення та переадресації запитів між складовими розгорнутими у Cloud Run.

Докладна ілюстрація використання модулів та їх взаємозв'язок продемонстровано на діаграмі розгортання, що наведена у додатку И.

Для автоматизації процесу розгортання використовується платформа Terraform [п], що дозволяє створити файли конфігурації проєктів та на їх основі здійснювати розгортання усіх компонентів системи без залучення користувацького інтерфейсу чи командної строки. Також це дозволяє налаштувати неперервну інтеграцію та доставку коду (CI/CD).

5.3.2 Розгортання клієнтської частини

В якості середовища розгортання клієнтського веб-застосунку використовується хмарне рішення Firebase Hosting, що є частиною екосистеми Google Cloud Platform.

Також як шар між клієнтом та користувацьким веб-застосунком було використано рішення від Cloudflare [10].

Серед переваг використання Cloudflare можна також виділити наступне:

- зручно керувати доменом, записами DNS та переадресаціями;
- захист нашої інформаційної системи від DDoS-атак, що є особливо важливим для забезпечення безперебійної роботи підприємств;
- можливість застосування Cloudflare Workers;

– використання кешування за допомогою CDN, що розподіляє нединамічний контент мережею серверів компанії на різних локаціях, таким чином, що швидкість доступу до даних від користувачів зростає.

На рисунку 5.4 зображено схему розгортання користувацького веб-застосунку, де клієнтський пристрій взаємодіє із окремими складовими системи. Спочатку для доступу до сторінки та коду, що буде виконано у веб-браузері клієнту встановлюється з'єднання із сервером імен, звідки підключення переадресується на

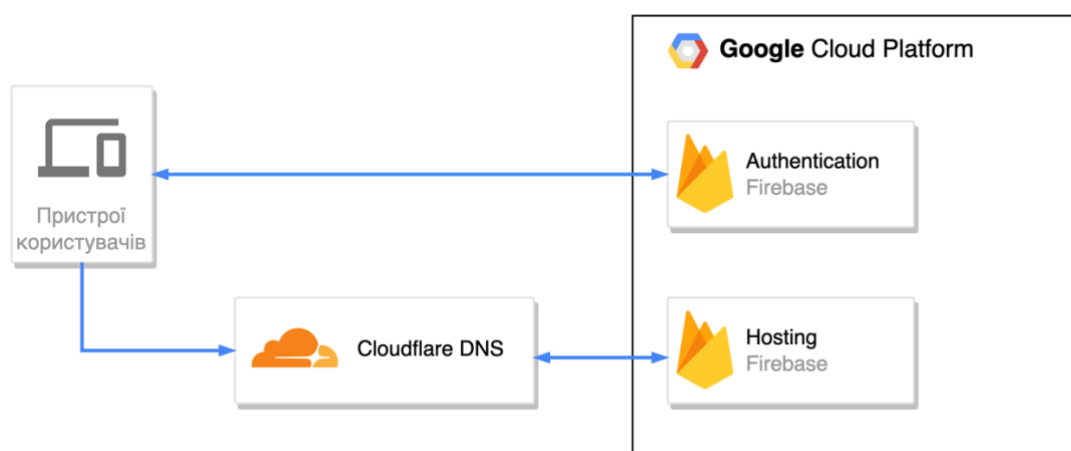


Рисунок 5.4 — Розгортання користувацького веб-застосунку

6 ІНСТРУКЦІЯ КОРИСТУВАЧА

6.1 Початок роботи

Перед початком роботи із інформаційною системою, передбачається, що користувач виконав наступні кроки:

- зв'язався із адміністратором системи для створення компанії чи доєднання користувача до вже існуючої системи;
- якщо користувачі або компанія вперше використовують систему, необхідно впевнитися, що пристрої для збору даних було встановлено вірно та вони працюють.

Для відкриття користувацького веб-застосунку, необхідно використати веб-браузер встановлений на пристрої користувача та перейти на сторінку застосунку.

6.2 Авторизація у системі

Для авторизації необхідно ввести свій пароль у форму зображену на рисунку 6.1. У разі потреби, можна скористатися формою «забули пароль?» та відновити доступ до акаунту.

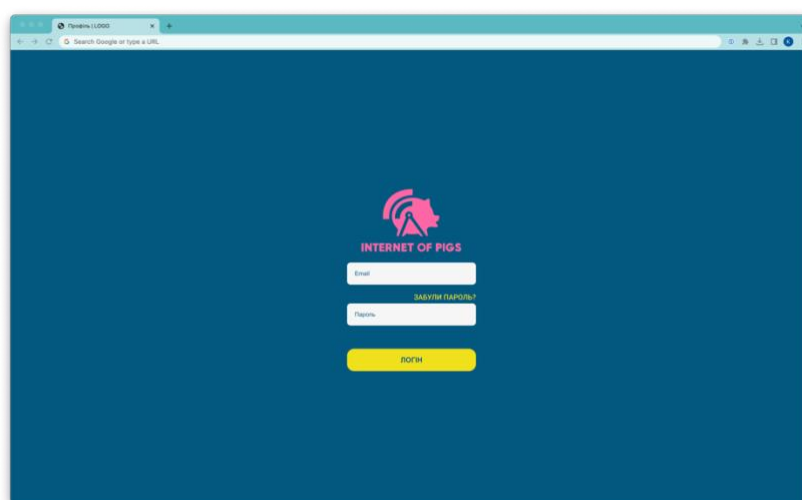


Рисунок 6.1 — Сторінка входу

Після авторизації, в залежності від ролі, користувача буде перенаправлено далі.

6.3 Огляд доступних об'єктів підприємств

У інтерфейсі користувача одразу відкривається перелік об'єктів підприємств у меню ліворуч та мапа, для швидкого доступу до маркерів підприємств. На рисунку 6.2 зображено головний екран, що зустрічає користувача та на якому ми бачимо ферми що потребують нашої уваги.

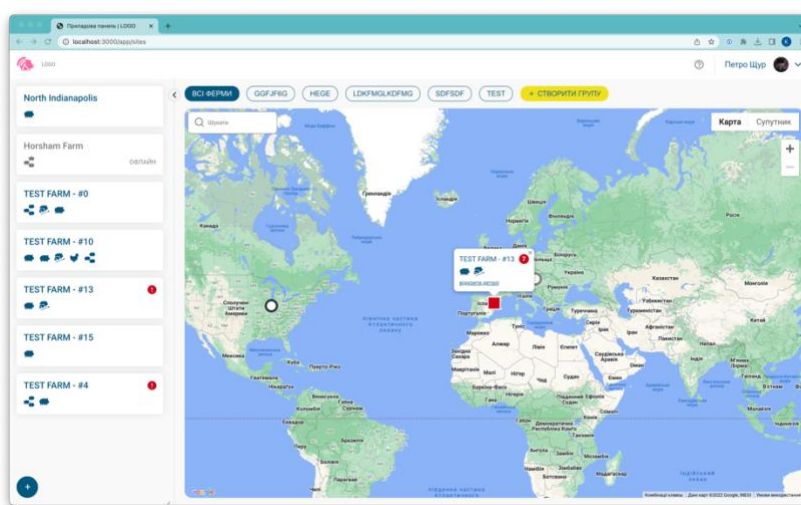


Рисунок 6.2 — Головна сторінка застосунку

6.4 Управління і огляд об'єкту підприємства

Перейшовши у детальний огляд об'єкт підприємства, ми можемо одразу на одній сторінці отримувати інформацію щодо наступних даних:

- поточний стан погоди у місцевості де розташовано об'єкт;
- оглянути одразу усі простори для спостережень та їх статуси;
- отримати інформацію про надіслані попереджень від системи.

На рисунку 6.3 продемонстровано головну сторінку із усіма показниками об'єкту. Червоними рамками виділено ті простори спостережень, на які треба

звернути увагу, сірим — вимкнені зі спостережень, та для яких можна почати спостереження.

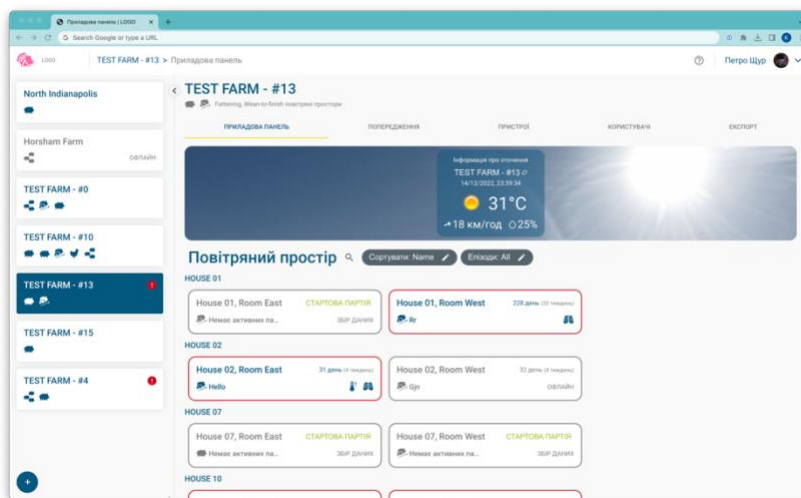


Рисунок 6.3 — Головна сторінка об'єкту

6.5 Перегляд показників спостережень

Для докладнішого вивчення даних що надходять із одного конкретного простору, можливо перейти до сторінки із наведеною у вигляді графіків інформацією.

На рисунку 6.4 представлено графік спостережень за ознаками респіраторних захворювань протягом останнього місяця. Лінія у графіку відповідає значенню «коефіцієнту здоров'я» від 0 до 100. Коефіцієнт означає, що при відсутності зафіксованих у цей день інцидентів із кашлем тварин, він набуває значення близьке до 100 (здорові). Нижче під графіком здоров'я можна переглянути зміни у температурі, вологості та відносному вмісту діоксиду вуглецю разом із графіками відповідних показників для зовнішнього середовища. Таким чином, відділ охорони здоров'я може провести аналіз анамнезу тварин призначити нове чи відкоригувати їх поточне лікування.

Для того щоб зафіксувати свої об'єктивні спостереження, користувачі можуть використовувати сторінку журналу, де для кожного окремого дня існує секція, із

варіантами додавання лікарських призначень або текстових нотаток. На рисунку 6.5 відображено сторінку із журналом спостережень за останні 4 тижні.

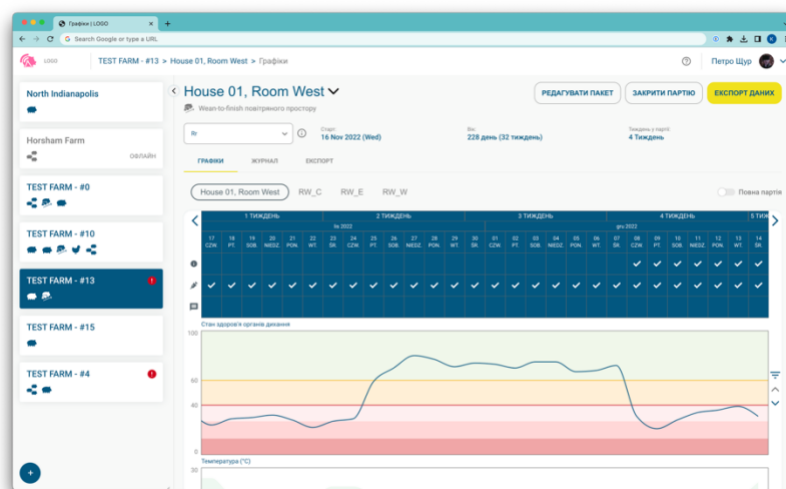


Рисунок 6.4 — Сторінка із показниками спостережень

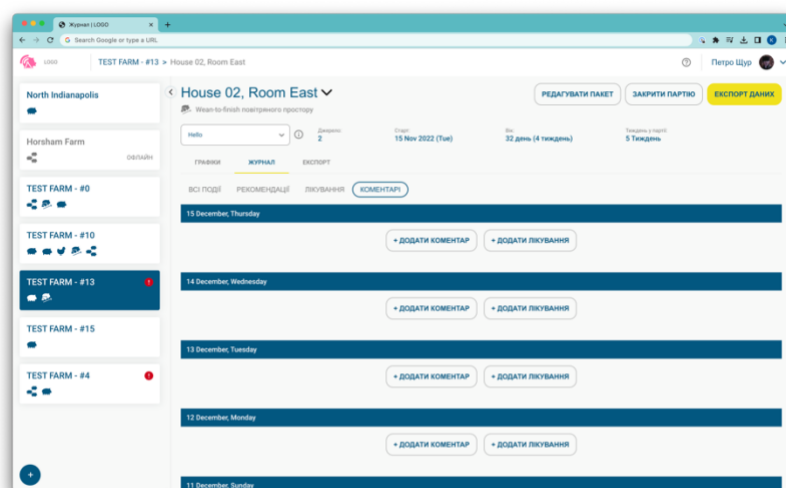


Рисунок 6.5 — Сторінка журналу спостережень

6.6 Перегляд попереджень

Коли важливо отримати інформацію про усі попередження, що надсилає система, та немає прив'язки до якогось конкретного простору спостережень, можна використовувати сторінку зі списком усіх попереджень отриманих для даного об'єкту підприємства за весь час. Екран попереджень відображено на малюнку 6.7.

Серед попереджень які відправляє система розділяємо наступні типи:

- виявлені ознаки кашлю;
- стрімка зміна у температурі приміщення;
- зміна статусу приладу для спостережень на «поза мережею».

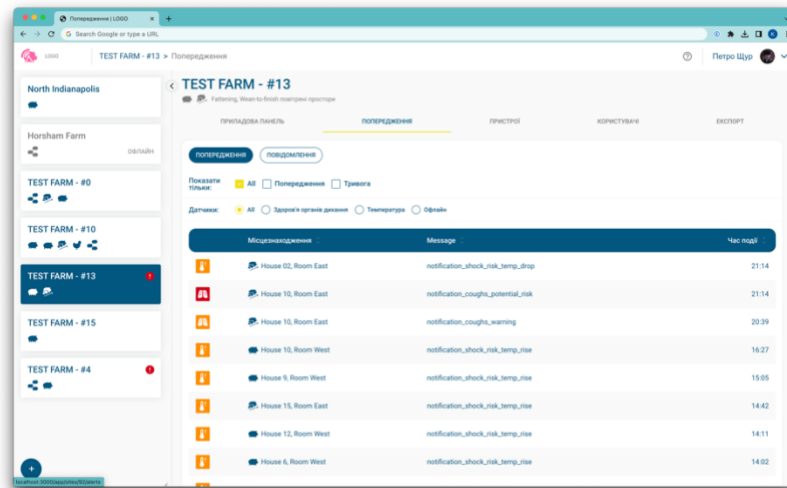


Рисунок 6.7 — Список усіх попереджень від системи

6.7 Перегляд та відправка повідомлень

Для перегляду повідомлень, необхідно перейти в розділ роботи із повідомленнями обравши користувача почати спілкування у чаті (рис 6.8).

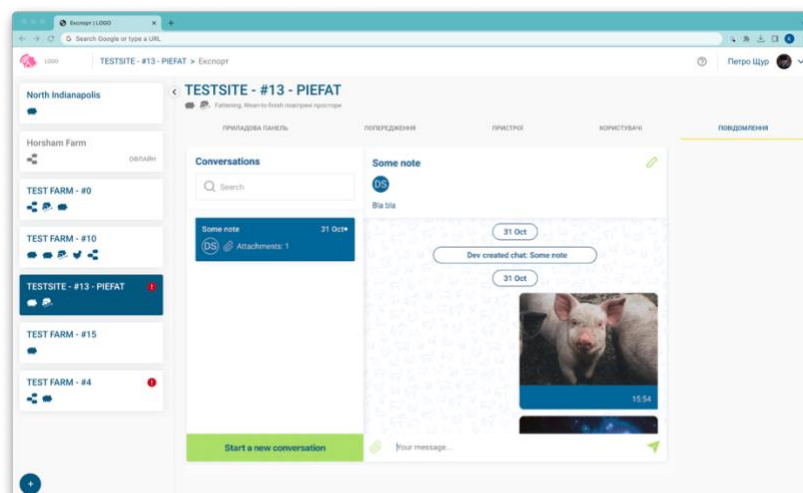


Рисунок 6.8 — Вікно повідомлень

6.8 Створення нового об'єкту

Для створення нового об'єкту, необхідно мати права доступу адміністратора підприємства. Після авторизації, на сторінці огляду усіх об'єктів (рис. 6.1) слід натиснути на синю кнопку із плюсом зліва знизу списку усіх ферм. Після цього користувача перенаправить на сторінку створення нового об'єкту зображену на рисунку 6.9.

У рамках першого кроку необхідно заповнити обов'язкові поля:

- назви;
- країни;
- позначити чи були попередньо налаштовані пристрої.

Далі необхідно обрати місцерозташування об'єкту на мапі.

Примітка — На мапі слід обрати точку, що відповідатиме географічному центру об'єкту.

Після завершення кроку, необхідно натиснути на кнопку «Наступний» зверху справа. Варто зауважити, що в разі потреби, користувач може покинути екран натиснувши на кнопку «Назад» чи обравши іншу сторінку в хлібних крихтах, при цьому прогрес додавання нового об'єкту буде збережений в системі.

Наступним кроком є конфігурація структури об'єкта. На рисунку 6.10, зображено форму введення відповідної структури за допомогою побудови ієрархічного дерева. Після натискання завершення заповнення і натискання кнопки переходу, у вікні буде запропоновано ввести ідентифікатори встановлених пристроїв, після завершення цього кроку ферму буде створено і додано до списку. Та додані пристрої з'являться на екрані усіх доступних пристроїв на об'єкті (рис 6.11).

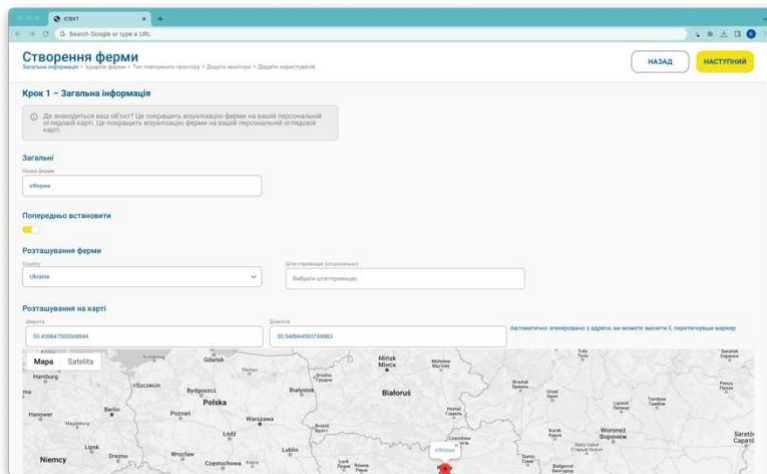


Рисунок 6.9 — Перший крок створення нового об'єкта

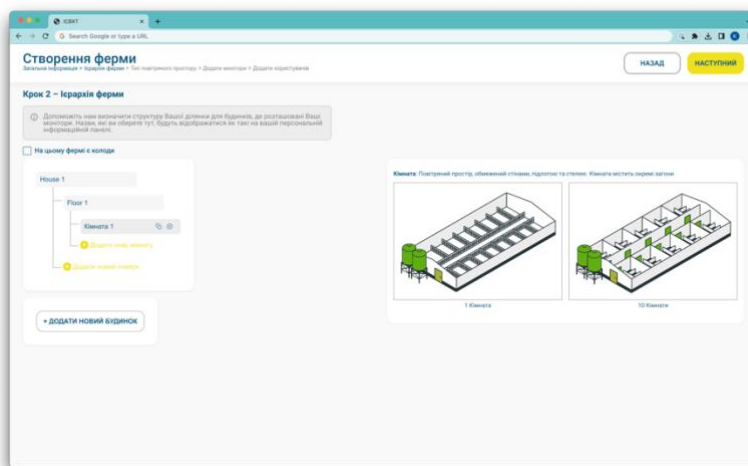


Рисунок 6.11 — Крок з конфігурації просторів для спостережень

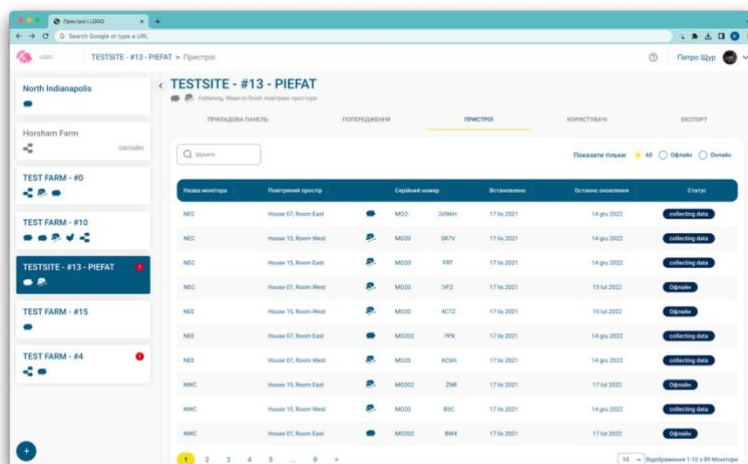


Рисунок 6.10 — Список усіх пристроїв зареєстрованих на об'єкті

6.9 Панель адміністратора

Для супер-адміністратора, передбачається додаткова можливість, а саме перегляд статистики та даних усіх основних представлених в рамках інформаційної системи сутностей:

- об'єктів підприємств;
- пристроїв збору даних;
- просторів для спостережень;
- користувачів системи.

На рисунках 6.11 та 6.12 представлено основні екрани доступні адміністратору системи для отримання додаткових даних поза основним користувацьким інтерфейсом.

The screenshot displays the administrator interface for site monitoring. At the top, there are three summary cards: 'Total sites' with a value of 61 (up 1 compared to last week), 'Online sites' with a value of 51% (2% compared to last week), and 'Subscriptions requiring action' with a value of 0 (0 compared to last week). Below these is a search bar and an 'Export table' button. The main part of the interface is a table with the following columns: ID, Name, Country, Owner, Supplier, Last connection, Gateways (Total, Offline), Monitors (Total, On, Warning, Alarm), and a final column with a '0' value. The table contains several rows of data, including sites like 'Site', 'EXPERIMENT', 'Raphi Industriale', 'Horseshoe Farm', 'Miran_site', 'MulligansTEST', and 'new'.

ID	Name	Country	Owner	Supplier	Last connection	Gateways		Monitors					
						Total	Offline	Total	On	Warning	Alarm		
109	Site	Belgia	No owner		8 days	1	1	3	1	0	0	0	3
129	EXPERIMENT	Belgia	No owner		0 mins	4	0	26	24	0	0	0	2
27	Raphi Industriale	Italy	Stary Zydziakowice	ATS	0 mins	2	0	17	0	0	0	0	17
26	Horseshoe Farm	Wenicy	No owner		538 days	1	1	2	0	0	0	0	2
122	Miran_site	Wenicy	No owner		35 days	0	0	0	0	0	0	0	0
131	MulligansTEST	Belgia	No owner		37 days	1	0	2	0	0	0	0	0
141	new	Argentina	Southwest		28 days	1	0	4	0	0	0	0	0

Рисунок 6.11 — Панель адміністратора

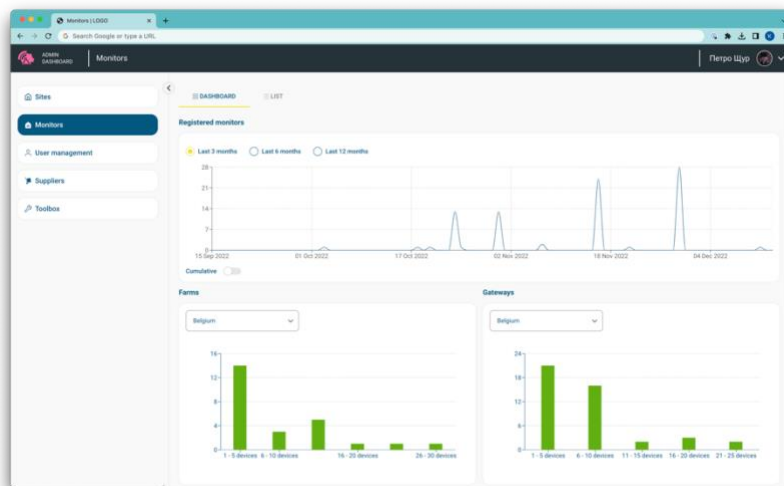


Рисунок 6.12 — Статистика пристроїв усієї системи

6.10 Виникнення помилок

У разі виникнення помилки на сторони веб-застосунку, система сповіщає користувача про це переводячи у «безпечну зону» (рис 6.13). Після цього, розробникам буде автоматично відправлено звіт про помилку.

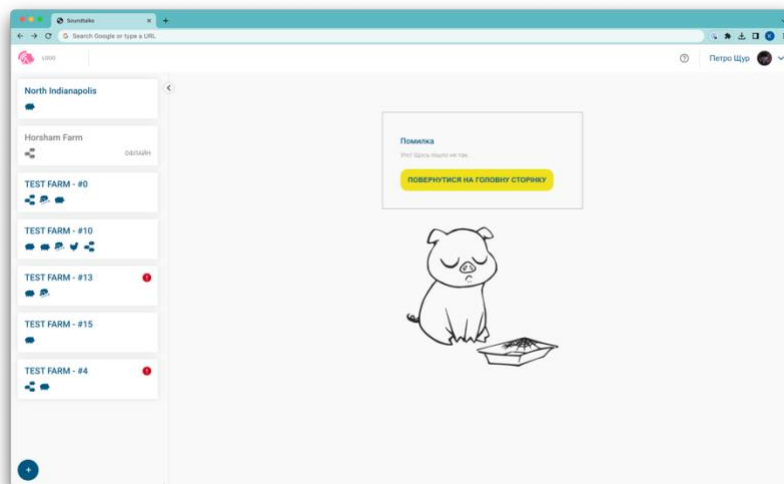


Рисунок 6.13 — Безпечна зона застосунку

7 СТАРТАП-ПРОЄКТ

Тваринництво, у різних його формах, є важливою складовою буденності майже кожної людини. Як і кожна система діяльності, цей процес може бути удосконалено і оптимізовано. Розробка інформаційної системи з виявлення захворювань худоби може запропонувати підприємствам комплексне рішення, пункти якого приведені у даному розділі.

7.1 Опис ідеї проєкту

Пропонується власна концепція інформаційної системи, яка за допомогою нейронних мереж, має за мету допомогти вчасно розпізнати поширення хвороби серед тварин. У таблиці 7.1 наведено опис ідеї рішення, та вигоди, які надає його застосування для визначених напрямків.

Таблиця 7.1 — Опис ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розробити програмне забезпечення, за допомогою якого визначати і фіксувати зміни середовища на тваринному підприємстві (кашель тварини) і вчасно інформувати кінцевого користувача про відхилення.	1. Інформатизація бізнес-процесів сільсько-господарських підприємств	Програмне забезпечення спрямоване на поліпшення взаємодії співробітників і ветеринарів підприємства, вчасне інформування про поточні показники і відхилення (ознаки хвороби), а також забезпечити безперервний онлайн моніторинг середовища в якому знаходяться тварини.

Зміст ідеї	Напрямки застосування	Вигоди для користувача
	2. Покращення умов утримання тварин	Завдяки автоматизації процесу моніторингу середовища, у підприємців буде можливість своєчасно надати допомогу хворим тваринам, що мають покращити показники лікування тварин.
	3. Покращення процесу догляду за тваринами задіяними в сільськогосподарській діяльності	Програмне забезпечення має на меті допомогти завчасно виявити захворювання і не допустити розповсюдження його на здорових тварин.

Також, проведено аналіз потенційних техніко-економічних переваг ідеї розроблюваного стартап-проєкту. В таблиці 7.2 наведені результати досліджень, а саме:

- визначено перелік техніко-економічних властивостей та характеристик ідеї;
- визначено попереднє коло конкурентів (проєктів-конкурентів) або товарів-замінників чи товарів-аналогів, що вже існують на ринку, та проведено збір інформації щодо значень техніко-економічних показників для ідеї власного проєкту та проєктів-конкурентів відповідно до визначеного вище переліку;
- проведено порівняльний аналіз показників: для власної ідеї визначено показники, що мають: гірші значення (W, слабкі), аналогічні (N, нейтральні) значення, кращі значення (S, сильні).

Для порівняння були обрані два найбільш схожі конкуренти:

- Farm Health Monitor від Farm Health Guardian;
- Nanoenvi FARM від ENVIRA IoT.

Таблиця 7.2 — визначення сильних, слабких та нейтральних характеристик ідеї

№ п/п	Техніко-економічні характеристики ідеї	Концепції конкурентів			Слабкі (W), нейтральні (N), та сильні (S) сторони
		Мій проєкт	Farm Health Monitor	Nanoenvi	
1.	Наявність датчиків зчитування зміни середовища	+ (Фіксація акустичної інформації)	-	+ (Збір показників добробуту: температура, відносна вологість, тощо)	S
2.	Розпізнавання ознак респіраторних хвороб	+	-	-	S
3.	Наявність автоматизованого збору інформації	+	-	+	N
4.	Централізоване сховище даних	+	+	+	N
5.	Наявність попереджень про потенційні аномалії	+	-	+	S

№ п/п	Техніко-економічні характеристики ідеї	Концепції конкурентів			Слабкі (W), нейтральні (N), та сильні (S) сторони
		Мій проєкт	Farm Health Monitor	Nanoenvi	
6.	Можливість комунікації команди охорони здоров'я підприємства	+	+	-	S

Провівши порівняльний аналіз наведений у таблиці 7.2, було визначено, що розроблювана система має значні переваги (сильні сторони) перед конкурентами, тобто є конкурентоспроможною.

7.2 Технологічний аудит ідеї

Розглянемо наявність та доступність технологій, необхідних для реалізації проєкту (таблиця 7.3).

Таблиця 7.3 — Технологічна здійсненність ідеї проєкту

№ п/п	Ідея проєкту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Відслідковування показників мікро-клімату	За допомогою датчиків температури, вологості, вмісту частинок PM2.5 та мікрокомп'ютерів, що можуть відправляти інформацію на сервер	Технологія вже наявна на ринку	Доступні та поширені

№ п/п	Ідея проєкту	Технології її реалізації	Наявність технологій	Доступність технологій
2	Запис звуку та розпізнання кашлю	Для розпізнавання звуків кашлю тварин, необхідно здійснювати запис та розпізнання за допомогою НМ.	Розпізнавання звуків за допомогою НМ	Технологія доступна

За результатами аналізу таблиці, зроблено висновок, що технологічна реалізація проєкту можлива і визначено технологічний шлях, яким доцільно реалізувати ідею проєкту.

7.3 Аналіз ринкових можливостей запуску стартап-проєкту

Підприємства які займаються тваринництвом поширені по всьому світу і є невід'ємною частиною ринкової економіки. Для моніторингу і автоматизації таких складних систем необхідні інноваційні рішення, побудовані на нейронних мережах, машинному навчанні і інших інноваційних підходах вирішення комплексних задач. Оскільки вони тільки починають виходити на ринок, та через складність побудови подібних рішень, конкуренція у цій галузі є низькою, що відкриває можливості для нових гравців.

Детальні характеристики потенційного ринку стартап-проєкту наведені у таблиці 7.4.

Таблиця 7.4 — Характеристика потенційного ринку стартап-проєкту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн/ум.од	\$ 5-8 млн.
3	Динаміка ринку (якісна оцінка)	Зростає

№ п/п	Показники стану ринку (найменування)	Характеристика
4	Наявність обмежень для входу (вказати характер обмежень)	Обмеження відсутні
5	Специфічні вимоги до стандартизації та сертифікації	Специфічні вимоги відсутні
6	Середня норма рентабельності в галузі (або по ринку), %	18-20%

За результатами аналізу таблиці зроблено висновок що, за попереднім оцінюванням, ринок є привабливим для входження.

В таблиці 7.5 визначено потенційні групи клієнтів, їх характеристики, та сформовано орієнтовний перелік вимог до товару для кожної групи.

Таблиця 7.5 – Характеристика потенційних клієнтів стартап-проєкту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Моніторинг і автоматизація процесу догляду за тваринами задіяними в сільськогосподарській діяльності	Підприємства, які займаються сільськогосподарською діяльністю, ферми, які займаються тваринництвом і приватні фермери	На підприємствах зазвичай присутнє розподілення задач між робітниками, тобто є люди, які	Наявність попереджень про аномалії та зміни в середовищі, з метою визначення хворої тварини, збереження

№ п/ п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
			займаються виключно моніторингом здоров'я тварин, ферми, навіть великі, часто можуть належати одній людині або сім'ї, де кожен з членів має декілька різнопланових задач, і тому особливо потребують автоматизації процесів.	даних, і можливість поділитися інформацією з іншими співробітниками та ветеринарами.
2.	Покращення умов утримання тварин з	Підприємства, які займаються	Епідемії та інші	Можливість своєчасного

№ п/ п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
	ціллію недопущення втрати ресурсу	сільськогосподарсько ю діяльністю, ферми, які займаються тваринництвом і приватні фермери	проблеми, які викликають смерть тварин є критичними для всіх груп клієнтів, і чим меншим є підприємство , тим відчутнішим є фактор втрати ресурсу.	виявлення хворих тварин, недопущення розповсюдженн я хвороби (ізоляція) і надання вчасної кваліфікованої допомоги тваринам.

Після визначення потенційних груп клієнтів необхідно провести аналіз ринкового середовища. Розглянемо фактори, що перешкоджають та сприяють впровадженню проєкту, наведені у таблиці 7.6 та таблиці 7.7 відповідно.

Таблиця 7.6 — Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Велика вартість обладнання	Оскільки для отримання даних необхідно встановлювати спеціальні датчики акустичної активності, вартість такого обладнання буде пропорційна масштабам господарства.	Можливість позичати таку техніку на період дії контракту, введення тестового періоду, щоб підприємства могли зрозуміти для себе цінність даного продукту, спеціальні знижки в залежності від масштабу господарства.
2	Помилкова інтерпретація даних	Нейронні мережі потребують часу для навчання і покращення якості аналізу, тому на початку використання продукту можлива помилкова інтерпретація даних.	Знижки першим клієнтам, обов'язкова вимога перегляду нотифікацій про аномалії спеціалістом.

Таблиця 7.7 — Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Відсутність пропозиції системи «під ключ»	Відсутність на ринку аналогів, які пропонують комплексну систему, розширює потенційний ринок та рівень зацікавленості до продукту.	Розвиток маркетингової стратегії, яка наголошує на мультифункціональності системи, що дає можливість користувачам не тільки відслідковувати ситуацію в режимі реального часу, а також систему сповіщень про аномалії, і можливість спілкування співробітників в одному місці.
2	Інноваційність підходу	Збільшує зацікавленість користувачів і якість користувацького досвіду	Гарна презентація продукту може також послугувати подальшому розвитку використання нейронних мереж та машинного навчання для автоматизації складних процесів. Подальше удосконалення нейронної мережі за рахунок навчання з отриманих даних на стороні клієнта.

Проведемо аналіз пропозиції та визначимо загальні риси конкуренції на ринку (таблиця 7.8).

Таблиця 7.8 — Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції — чиста	На ринку існують схожі рішення, які, в той же час, не вирішують проблему поставлену перед даною системою	Пропонується унікальне рішення на базі інноваційних технологій, яке покращується в процесі його використання, оскільки це і є особливістю навчання нейронних мереж
2. За рівнем конкурентної боротьби — міжнародний	Проект призначений для використання на міжнародному ринку	Необхідність урахування потреб міжнародного ринку, та підтримки глобалізації, перекладу різними мовами
3. За галузевою ознакою — внутрішньогалузева	Продукт задовольняє потреби в одній галузі – пов'язаній з визначенням ознак респіраторних захворювань у тварин.	Інноваційність підходу до вирішення комплексного завдання
4. Конкуренція за видами товарів — товарно-видова	З появою аналогів на ринку, з'явиться конкуренція між аналогічними продуктами	Необхідність регулярного моніторингу ринку з метою створення унікального досвіду для користувача
5. За характером конкурентних переваг — нецінова	Конкуренція за рахунок надання унікального досвіду	Автоматизація комплексного підходу до вирішення складних рішень

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
6. За інтенсивністю — не марочна	Продукт є новим на ринку	Компанія має орієнтуватися на надання якісних послуг, правильна робоча взаємодія є важливим аспектом для розвитку продукту

Інформаційна система виявлення захворювання худоби має гарні можливості: інноваційний підхід до вирішення нішевої проблеми потенційно має гарні шанси легкого виходу на ринок.

Проведемо більш детальний аналіз умов конкурентності в галузі за М. Портером (таблиця 7.9).

Таблиця 7.9 — Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальник	Клієнти	Товари-замінники
	Nanoenvi FARM	Farm Health Monitor	Інтернет	Підприємства, що займаються тваринництвом, приватні фермери	Немає товарів-замінників
Висновки	Конкуренція середня, стартап-проект пропонує інший підхід до вирішення проблеми, з більшими широкими можливостями використання	Конкурент вже є на ринку, проте являє собою лише неавтоматизовану систему для спілкування між співробітниками, і не надає можливість відслідковувати стан тварин безпосередньо	Постачальники не диктують умови роботи на ринку	Клієнти хочуть отримати результат, і оптимізувати процеси на підприємстві, а отже мають високі очікування	Обмежень немає через відсутність товарів-замінників

Далі необхідно розглянути фактори конкурентоспроможності стартап-проекту, що дозволять продукту залишатися затребуваним на ринку (таблиця 7.10).

Таблиця 7.10 — Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проєктів значущим)
1.	Інноваційність	Використання новітніх технологій, таких як нейронні мережі, дозволяє розширити можливості автоматизації складних процесів, які зазвичай контролюються людиною. А саме вчасне розпізнання хвороби, і сповіщення систему охорони здоров'я компанії про заклик до дій щодо тварин які потребують уваги і лікування.
2	Система, що самовдосконалюється	Нейронні мережі дозволяють оновлювати алгоритм розпізнання акустичних ознак хвороби тварин, що в свою чергу покращує точність результатів.
3	Швидкість обробки інформації	Надає користувачу можливість завчасно отримати сповіщення про аномалії, та попередити розповсюдження хвороби на інших тварин
4	Безперервний онлайн моніторинг	Надає користувачу можливість віддалено відслідковувати стан тварин навіть у неробочі години і отримувати оповіщення про поточні аномалії.
5	Система під ключ	Збір метрик, синхронізація та сповіщення відбуваються в рамках однієї комплексної системи, що економить час користувача на налаштування цих процесів.

За визначеними факторами конкурентоспроможності проведемо аналіз сильних та слабких сторін стартап-проєкту у таблиці 7.11.

Таблиця 7.11 — Порівняльний аналіз сильних та слабких сторін системи

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з стартап проектом						
			-3	-2	-1	0	+1	+2	+3
1	Інноваційність	20		+					
2	Система, що самовдосконалюється	17			+				
3	Швидкість обробки інформації	18		+					
4	Безперервний онлайн моніторинг	20	+						
5	Система під ключ	20	+						

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities). Матриця наведена у таблиці 7.12.

Таблиця 7.12 — SWOT-аналіз стартап-проекту

<p>Сильні сторони:</p> <ol style="list-style-type: none"> Інноваційність Швидкість обробки інформації Система, що самовдосконалюється Безперервний онлайн моніторинг Система під ключ 	<p>Слабкі сторони:</p> <ol style="list-style-type: none"> Велика вартість обладнання
<p>Можливості:</p> <ol style="list-style-type: none"> Відсутність пропозицій системи «під ключ» 	<p>Загрози:</p> <ol style="list-style-type: none"> Недостатній попит на новий продукт

2. Інноваційність підходу	2. Страх користувачів щодо впровадження новітніх технологій у сталу систему
3. Стабільна галузь використання	

Отже, стартап-проект має слабкі сторони та загрози, проте його сильні сторони та можливості мають стати рушійною силою для запуску і реалізації проекту. Розглянемо альтернативи ринкової поведінки для введення проекту на ринок та орієнтований оптимальний час їх реалізації для зменшення ризиків (таблиця 7.13).

Таблиця 7.13 — Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Розробка мобільного застосунку або мобільної версії веб-застосунку	Середня ймовірність отримання ресурсів	6 місяців
2	Знайти партнерів серед лідерів агропромислових компаній	Висока ймовірність отримання ресурсів	3 місяці
3	Створити експериментальну інтеграцію за власний кошт із одним із існуючих підприємств	Висока ймовірність отримання ресурсів	1 місяць

7.4 Розроблення ринкової стратегії проекту

При розробленні ринкової стратегії, спочатку необхідно визначити стратегію охоплення ринку – описати цільові групи потенційних споживачів (таблиця 7.14).

Таблиця 7.14 — Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачі в сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Підприємства, які займаються сільськогосподарською діяльністю	Висока готовність	Висока	Не висока конкуренція	Середня складність, треба витратити час і ресурси щоб довести корисність системи
2	Ферми, які займаються тваринництвом	Готові розглянути як можливість покращення існуючих процесів.	Середня	Низька конкуренція	Середня складність, треба витратити час і ресурси щоб довести корисність системи
3	Приватні фермери	Низька готовність, при малих	Мала	Низька конкуренція, проте	Складно через консервативність приватних підприємців і

№ п/ п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачі в сприйняти продукт	Орієнтовн ий попит в межах цільової групи (сегменту)	Інтенсивніс ть конкуренції в сегменті	Простота входу у сегмент
		масштабах може бути фінансово не вигідно.		малий попит	закритість до нових технологій

За результатами аналізу потенційних груп споживачів, було обрано наступні цільові групи для яких пропонуватиметься продукт:

- підприємства, які займаються сільськогосподарською діяльністю;
- великі ферми, які займаються тваринництвом,
- приватні фермери.

Пропонується працювати зі всім ринком та обрати стандартизовану програму — масовий маркетинг. Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку (таблиця 7.15).

Таблиця 7.15 — Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Розвиток у всіх доступних сегментах агропромислового ринку орієнтованого на тварин	Пропозиція покращення процесу з використанням новітніх технологій, який раніше неможливо було автоматизувати через брак необхідних технологій	Інноваційне рішення комплексної проблеми, яку зазвичай закриває людина, що виключає людський фактор і оптимізує процес виявлення захворювань серед тварин.	Стратегія диференціації

Далі необхідно визначити стратегію конкурентної поведінки (таблиця 7.16).

Таблиця 7.16 — Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Так	Компанія шукатиме нових споживачів	Так, компанія буде копіювати найкращі рішення, покращуючи їх і нівелюючи недоліки, якщо буде така потреба	Стратегія заняття конкурентної ніші.

На основі попередніх стратегій представлених у таблицях 7.14 – 7.16, розробляється стратегія позиціонування (таблиця 7.17), що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 7.17 — Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Швидкість обробки інформації та наявність сповіщень про відхилення	Стратегія диференціації	Наявність автоматичного збору інформації та навченої системи реагування на відхилення від норми (акустичні ознаки респіраторного захворювання)	Якість продукту, оптимізація процесів, зручність
2	Безперервний онлайн моніторинг	Стратегія диференціації	Зручний застосунок в якому вся інформація відображається в реальному часі	Якість продукту, оптимізація процесів, зручність
3	Система не вимагає додаткового налаштування від користувача	Стратегія диференціації	Система під ключ	Якість продукту, оптимізація процесів, зручність

За результатами виконання підрозділу маємо узгоджену ринкову стратегію проекту інформаційної системи виявлення захворювання худоби. Планується

охопити весь сегмент ринку агропродукції, орієнтованого на тваринництво з використанням підходів зазначених у розділі.

7.5 Розроблення маркетингової програми стартап-проєкту

Одним із основних етапів розроблення стартап-проєкту є необхідність сформулювати маркетингову концепцію товару, який отримає споживач. Для цього підсумуємо результати попереднього аналізу конкурентоспроможності товару (таблиця 7.18).

Таблиця 7.18 — Визначення переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Моніторинг і автоматизація процесу догляду за тваринами задіяними в сільськогосподарській діяльності	Наявність зручного застосунку, який буде оновлювати дані в режимі реального часу і повідомляти про аномалії.	Завдяки автоматизації моніторингу, у підприємців є можливість своєчасно надати допомогу тваринам. Програмне забезпечення має на меті допомогти завчасно виявити захворювання і не допустити розповсюдження.
2	Покращення умов утримання тварин з	Вчасне інформування про зміни в середовищі, і ознаки захворювання	ПЗ спрямоване на поліпшення взаємодії

	ціллю недопущення втрати ресурсу	тварин, можливість зберігати інформацію і обмінюватися нею з ветеринарами та іншими співробітниками підприємства.	співробітників і ветеринарів, вчасне інформування про поточні показники і відхилення (ознаки хвороби), а також забезпечити безперервний онлайн моніторинг середовища в якому знаходяться тварини.
--	----------------------------------	---	---

Далі розробимо трирівневу маркетингову модель товару, що представлена у таблиці 7.19.

Таблиця 7.19 — Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
І. Товар за задумом	інформаційна система, яка за допомогою нейронних мереж, має за мету допомогти вчасно розпізнати поширення хвороби серед тварин.		
ІІ. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	Наявність датчиків зчитування зміни середовища	Нм	Вт Тл
	Розпізнавання ознак респіраторних хвороб	М	Тх Тл
	Автоматизований збір інформації	М	Тх
	Централізоване сховище даних	М	Тх
	Надсилання повідомлень про потенційні аномалії	Нм	Тх

	Реалізація каналу комунікації команди охорони здоров'я підприємства	М	Тх
	Якість: пристрої мають відповідати стандартам захищеності IP68 та проходити тестування вибірками.		
	Пакування: коробка з перероблених екологічних матеріалів, можливо кукурудзи		
	Марка: Інформаційна система виявлення захворювання худоби		
III. Товар із підкріпленням	До продажу – презентація товару на тематичних агропромислових форумах, експериментальна інтеграція з одним із існуючих підприємств		
	Після продажу – повноцінна система під ключ, з ліцензіями на користування		
За рахунок чого потенційний товар буде захищено від копіювання: надання ліцензій без яких програмне забезпечення та технічне оснащення не зможе працювати			

У таблиці 7.19 для другого рівню моделі товару наведено властивості/характеристики товару за наступними критеріями:

- М/Нм – монотонні або немонотонні;
- Вр/Тх/Тл/Е/Ор – вартісні, технічні, технологічні, ергономічні або органолептичні (для продуктів харчування).

Інформаційну систему виявлення захворювання худоби планується захищати ліцензією, без якої програмне забезпечення та технічне оснащення не буде працювати. Окремі системні частини (такі як алгоритм розпізнання кашлю тварини) можна запатентувати, що також буде додатковою можливістю розвитку цієї технології в майбутньому.

Наступним кроком сформуємо систему збуту (таблиця 7.20).

Таблиця 7.20 — Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Оплата за користування сервісом інформування, підтримку та обслуговування системи	Реклама, послуги з підтримки та обслуговування продукту	Канал прямого маркетингу	Через відділ збуту

Останньою складовою маркетингової програми є розробка концепції маркетингових комунікацій (таблиця 7.21), основою на попередньому аналізі проведеному раніше у розділі.

Таблиця 7.21 — розробка концепції маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Користувачі очікують можливість віддалено отримувати показники стану	Інтернет, електронна пошта, тематичні агропромислові форуми	Інноваційний підхід, сповіщення про прояви захворювання в реальному часі,	Повідомлення має переконати потенційного користувача, що система значно полегшить	Інноваційне рішення комплексної проблеми

№ п/ п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонуванн я	Завдання рекламного повідомлення	Концепція рекламного звернення
	здоров'я тварин в реальному часі, а також бути сповіщені про будь-які ознаки захворюванн я серед тварин.		Сервіс комунікації співробітників , Система “під ключ”	процес догляду за тваринами, допоможе попередити розповсюдженн я хвороб, і не допустити їх зріст у епідемію.	

Висновки до розділу

У даному розділі було представлено зміст ідеї стартап-проекту інформаційної системи виявлення захворювання худоби, напрямки його застосування, проведено аналіз ринкових можливостей запуску стартап-проекту.

Система має значні переваги в порівнянні з конкурентами, а саме пропонує мультифункціональне рішення, яке дозволяє не лише відслідковувати стан тварин в реальному часі, а також отримувати нотифікації про аномалії, і передавати ці дані між співробітниками компанії за допомогою одного сервісу. Використання нейронних мереж дозволяє навчити систему розпізнавати кашель тварини як акустичну інформацію, і використовувати її для виявлення хвороб на агропромисловому підприємстві.

Під час аналізу ринкових можливостей запуску стартап-проекту було виявлено, що серед потенційних цільових груп клієнтів є високий попит на збереження основного промислового ресурсу. Агропромисловий комплекс є ваговою складовою світової економіки, і тваринництво є значущою частиною даної галузі. Тож її розвиток та оптимізація дає гарні можливості для успішної реалізації і подальшого розростання проекту.

На сьогодні, жодних вимог до сертифікації та стандартизації немає, що в свою чергу потенційно спрощує вихід системи на ринок.

Також під час аналізу ринкових можливостей, було виявлено, що система може мати неточності під час аналізу акустичних даних. В цьому полягає особливість використання нейронних мереж для використання продукту, оскільки ми можемо контролювати якість «навчальних матеріалів» для системи, а також, маючи справжніх клієнтів, оновлювати базу даними зі сторони замовників.

При реалізації проекту доцільно обирати розробку застосунку або мобільної версії веб-застосунку як альтернативу впровадження.

Оскільки галузь тваринництва є стабільною і попит на неї поступово збільшується, є потреба в оптимізації і удосконаленні існуючих підходів. Інформаційної система виявлення захворювання худоби надає підходяще рішення, яке має на меті допомогти в зниженні втрати тваринного ресурсу. Тому подальша імплементація проекту є доцільною і актуальною.

ВИСНОВКИ

Сільськогосподарські підприємства є важливою складовою життя людей. Попри розвиток науки і техніки, однією із найбільших проблем тваринництва залишається втрата його основного ресурсу — худоби. Серед причин втрати поголів'я, перше місце за частотою випадків займають саме втрати від респіраторно-інфекційних захворювань.

Вчасне виявлення пришвидшує реагування для усунення причини захворювання і таким чином допомагає уникнути розповсюдження.

В рамках даної роботи було оглянуто наявні на ринку рішення з онлайн відслідковування статусу сільськогосподарських підприємств та автоматизації їх бізнес-процесів пов'язаних із охороною здоров'я тварин. Серед існуючих аналогів, майже усі системи покривають тільки частину необхідних задач, а обробка акустичного оточення тварин для виявлення ознак кашлю не здійснюється або тільки планується їх розробка.

В зв'язку із цим було проведено детальний огляд технологій збереження, обробки і трансформації звукових сигналів, розглянуто існуючі методи розпізнавання звуків оточуючого середовища. В рамках дослідження, було виявлено, що для розпізнавання звуків кашлю підходять методи машинного навчання, а саме використання згорткових нейронних мереж для класифікації зображень мел-частотних кепстральних коефіцієнтів, які за допомогою кількох шарів згортки, можуть визначити чи було на вхідних звукових даних зафіксовано кашель.

Надалі, було спроектовано та розроблено серверну частину та клієнтський веб-застосунок інформаційної системи для збереження і обробки даних, що мають надходити від модулю розпізнавання кашлю та інших датчиків з фізичних пристроїв.

Для отриманого рішення, було зроблено технічний аудит, проаналізовано ринок та розроблено стратегію ринкового та маркетингового розвитку проєкту.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Державна служба статистики України. Виробництво та розподіл валового внутрішнього продукту за видами економічної діяльності. Валова додана вартість (у фактичних цінах; млн грн) [Електронний ресурс] / Державна служба статистики України // Набори даних - Портал відкритих даних Data.gov.ua. — Режим доступу: <https://data.gov.ua/dataset/e73d00ac-aba5-4dba-aa72-58181e465243/resource/3fbec822-0389-4575-b8e6-664df39e16db>. — Назва з екрана.
2. Числові та функціональні ряди. Ряди Фур'є. Метод. вказівки до вивчення теми дисципліни «Вища математика» для студентів енергетичних спеціальностей усіх форм навчання [Електронний ресурс]: навч. посіб. / уклад.: М. Черней, Г. Новикова, Н. Денисенко. — Київ : НТУУ «КПІ», 2016. — 62 с. — Режим доступу: https://ela.kpi.ua/bitstream/123456789/17673/1/Cherney_Ryadi_%202016.pdf (дата звернення: 15.11.2022). — Назва з екрана.
3. Abilisense sound recognition technology [Електронний ресурс] // Abilisense. — Режим доступу: <https://www.abilisense.com/tech> (дата звернення: 01.11.2022). — Назва з екрана.
4. Ahmed M. R. Automatic environmental sound recognition (AESR) using convolutional neural network [Електронний ресурс] / Md Rayhan Ahmed, Towhidul Islam Robin, Ashfaq Ali Shafin // International journal of modern education and computer science. — 2020. — Т. 12, № 5. — С. 41—54. — Режим доступу: <https://doi.org/10.5815/ijmecs.2020.05.04> (дата звернення: 08.11.2022). — Назва з екрана.
5. Animal health monitoring on farms [Електронний ресурс] // Envira IoT. — Режим доступу: <https://enviraiot.com/animal-health-monitoring-farms> (дата звернення: 01.11.2022). — Назва з екрана.
6. Audio analytic [Електронний ресурс] // Audio Analytic. — Режим доступу: <https://www.audioanalytic.com/about/> (дата звернення: 01.11.2022). — Назва з екрана.
7. Bansal A. Environmental Sound Classification: a descriptive review of the literature [Електронний ресурс] / Anam Bansal, Naresh Kumar Garg // Intelligent systems

with applications. — 2022. — С. 200115. — Режим доступу: <https://doi.org/10.1016/j.iswa.2022.200115> (дата звернення: 08.11.2022). — Назва з екрана.

8. Bhat G. S. Automated machine learning based speech classification for hearing aid applications and its real-time implementation on smartphone [Електронний ресурс] / Gautam Shreedhar Bhat, Nikhil Shankar, Issa M. S. Panahi // 2020 42nd annual international conference of the IEEE engineering in medicine and biology society (EMBC) in conjunction with the 43rd annual conference of the canadian medical and biological engineering society, Montreal, QC, Canada, 20—24 лип. 2020 р. — [Б. м.], 2020. — Режим доступу: <https://doi.org/10.1109/embc44109.2020.9175693> (дата звернення: 08.11.2022). — Назва з екрана.

9. Chachada S. Environmental sound recognition: a survey [Електронний ресурс] / Sachin Chachada, C. C. Jay Kuo // 2013 asia-pacific signal and information processing association annual summit and conference (APSIPA), Kaohsiung, Taiwan, 29 жовт. — 1 листоп. 2013 р. — [Б. м.], 2013. — Режим доступу: <https://doi.org/10.1109/apsipa.2013.6694338> (дата звернення: 08.11.2022). — Назва з екрана.

10. Cloudflare - the web performance & security company [Електронний ресурс] // Cloudflare. — Режим доступу: <https://www.cloudflare.com/> (дата звернення: 15.11.2022). — Назва з екрана.

11. Elbir A. Music genre classification and music recommendation by using deep learning [Електронний ресурс] / A. Elbir, N. Aydin // Electronics letters. — 2020. — Т. 56, № 12. — С. 627—629. — Режим доступу: <https://doi.org/10.1049/el.2019.4202> (дата звернення: 08.11.2022). — Назва з екрана.

12. Fancom | Helping poultry and pig producers to grow [Електронний ресурс] // Fancom. — Режим доступу: <https://www.fancom.com> (дата звернення: 01.11.2022). — Назва з екрана.

13. Farm Health Monitor [Електронний ресурс] // Farm Health Guardian. — Режим доступу: <https://farmhealthguardian.com/farm-health-monitor> (дата звернення: 01.11.2022). — Назва з екрана.

14. Grey J. M. Perceptual effects of spectral modifications on musical timbres [Електронний ресурс] / John M. Grey, John W. Gordon // The journal of the acoustical society of america. — 1978. — Т. 63, № 5. — С. 1493—1500. — Режим доступу: <https://doi.org/10.1121/1.381843> (дата звернення: 11.11.2022). — Назва з екрана.
15. gRPC [Електронний ресурс] // gRPC. — Режим доступу: <https://grpc.io> (дата звернення: 15.11.2022). — Назва з екрана.
16. Keras: the Python deep learning API [Електронний ресурс] // Keras: the Python deep learning API. — Режим доступу: <https://keras.io> (дата звернення: 15.11.2022). — Назва з екрана.
17. Logan B. Mel frequency cepstral coefficients for music modeling [Електронний ресурс] / Beth Logan // ISMIR 2000, Plymouth, Massachusetts, 23—25 жовт. 2000 р. — Cambridge, 2000. — С. 1—11. — Режим доступу: https://ismir2000.ismir.net/papers/logan_paper.pdf (дата звернення: 01.11.2022). — Назва з екрана.
18. Mushtaq Z. Spectral images based environmental sound classification using CNN with meaningful data augmentation [Електронний ресурс] / Zohaib Mushtaq, Shun-Feng Su, Quoc-Viet Tran // Applied acoustics. — 2021. — Т. 172. — С. 107581. — Режим доступу: <https://doi.org/10.1016/j.apacoust.2020.107581> (дата звернення: 08.11.2022). — Назва з екрана.
19. Palo H. K. Recognition of human speech emotion using variants of mel-frequency cepstral coefficients [Електронний ресурс] / Hemanta Kumar Palo, Mahesh Chandra, Mihir Narayan Mohanty // Advances in systems, control and automation. — Singapore, 2017. — С. 491—498. — Режим доступу: https://doi.org/10.1007/978-981-10-4762-6_47 (дата звернення: 08.11.2022). — Назва з екрана.
20. Porphyrio | Keep an eye on your poultry business [Електронний ресурс] // Porphyrio. — Режим доступу: <https://www.porphyr.io.com> (дата звернення: 01.11.2022). — Назва з екрана.
21. React — JavaScript-бібліотека для створення користувацьких інтерфейсів [Електронний ресурс] // React — JavaScript-бібліотека для створення користувацьких

інтерфейсів. — Режим доступу: <https://uk.reactjs.org/> (дата звернення: 15.11.2022). — Назва з екрана.

22. React Native · Learn once, write anywhere [Електронний ресурс] // React Native · Learn once, write anywhere. — Режим доступу: <https://reactnative.dev> (дата звернення: 15.11.2022). — Назва з екрана.

23. Ricardo C. M. Relational database systems [Електронний ресурс] / Catherine M. Ricardo // Encyclopedia of information systems. — [Б. м.], 2003. — С. 661—680. — Режим доступу: <https://doi.org/10.1016/b0-12-227240-4/00147-7> (дата звернення: 12.11.2022). — Назва з екрана.

24. Scikit-learn: machine learning in Python [Електронний ресурс] // scikit-learn: machine learning in Python. — Режим доступу: <https://scikit-learn.org/> (дата звернення: 15.11.2022). — Назва з екрана.

25. SEE SOUND [Електронний ресурс] // SEE SOUND by Wavio. — Режим доступу: <https://www.see-sound.com> (дата звернення: 01.11.2022). — Назва з екрана.

26. Sharma J. Environment sound classification using multiple feature channels and attention based deep convolutional neural network [Електронний ресурс] / Jivitesh Sharma, Ole-Christoffer Granmo, Morten Goodwin // Interspeech 2020. — ISCA, 2020. — Режим доступу: <https://doi.org/10.21437/interspeech.2020-1303> (дата звернення: 08.11.2022). — Назва з екрана.

27. Szandała T. Review and comparison of commonly used activation functions for deep neural networks [Електронний ресурс] / Tomasz Szandała // Bio-inspired neurocomputing. — Singapore, 2020. — С. 203—224. — Режим доступу: https://doi.org/10.1007/978-981-15-5495-7_11 (дата звернення: 11.11.2022). — Назва з екрана.

28. TensorFlow [Електронний ресурс] // TensorFlow. — Режим доступу: <https://www.tensorflow.org/> (дата звернення: 15.11.2022). — Назва з екрана.

29. Terraform by HashiCorp [Електронний ресурс] // Terraform by HashiCorp. — Режим доступу: <https://www.terraform.io> (дата звернення: 15.11.2022). — Назва з екрана.

30. Understanding machine learning: from theory to algorithms. — USA : Cambridge University Press, 2014.

31. USDA. 2015. “Cattle and Calves Death Loss in the United States Due to Predator and Nonpredator Causes, 2015”. — Fort Collins, CO : USDA—APHIS—VS—CEAH, 2017. — 77 с.

32. Zhang H. Robust sound event recognition using convolutional neural networks [Електронний ресурс] / Haomin Zhang, Ian McLoughlin, Yan Song // ICASSP 2015 - 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP), South Brisbane, Queensland, Australia, 19—24 квіт. 2015 р. — [Б. м.], 2015. — Режим доступу: <https://doi.org/10.1109/icassp.2015.7178031> (дата звернення: 08.11.2022). — Назва з екрана.