

КВАЛІФІКАЦІЙНА
РОБОТА МАГІСТРА

КРМ.АКС_м-21.00.00.000 ПЗ

Група АКС_м-24-1

Назар ТИТИШ
2025

Міністерство освіти і науки України
Івано-Франківський національний технічний університет нафти і газу
Факультет інформаційних технологій
Кафедра інформаційно-телекомунікаційних технологій і систем

Титиш Назар Володимирович

(прізвище, ім'я, по-батькові)

УДК 621.643:004.852
(індекс)

МАГІСТЕРСЬКА РОБОТА

Інтелектуальна система детектування витоків у трубопроводах з використанням
AWS SageMaker

(назва роботи)

Комп'ютеризовані системи управління та автоматика

(назва освітньої програми)

174 - Автоматизація, комп'ютерно-інтегровані технології та робототехніка

(шифр і назва спеціальності)

**Робота містить результати власних досліджень, використання ідей,
результатів і текстів інших авторів мають посилання на відповідне
джерело:**

Здобувач освітнього рівня _____ Н. В. Титиш

(підпис, ініціали та прізвище здобувача)

Науковий керівник _____ к.т.н., доц. Л.О. Штаєр

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

Допущено до захисту

Завідувач кафедри _____ д.т.н., проф. Л.М. Заміховський

(посада)

(підпис)

(дата)

(ініціали та прізвище)

Івано-Франківськ

2025

Івано-Франківський національний технічний університет нафти і газу

(повне найменування закладу вищої освіти)

Факультет Інформаційних технологій

Кафедра Інформаційно-телекомунікаційних технологій та систем

Освітній рівень Магістр

Спеціальність 174-Автоматизація, комп'ютерно-інтегровані технології та робототехніка

(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТТС, д.т.н. проф.

Л. М. Заміховський

« _____ » _____ 2025 року

З А В Д А Н Н Я

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТОВІ

Титишу Назару Володимировичу

1. Тема роботи Інтелектуальна система детектування витоків у трубопроводах з використанням AWS SageMaker

керівник роботи Штаєр Лідія Омелянівна, к.т.н.,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від " 30 " жовтня 2025 року № 690/7

2. Строк подання студентом роботи 22.12.2025 р.

3. Вихідні дані до роботи Матеріали та результати отримані під час проходження переддипломної практики, методи розробки програмного забезпечення, сервіси хмарної платформи AWS, алгоритми та методичні вказівки.

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

1 Аналіз методів контролю витоків та обґрунтування архітектури інтелектуальної системи

2 Розроблення методичного та алгоритмічного забезпечення класифікації станів трубопроводу

3 Реалізація інтелектуальної системи контролю витоків та оцінка її ефективності

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Контекстна діаграма процедури навчання моделей та тестування

Порівняльний аналіз якості моделей

Архітектура системи збору, опрацювання та аналізу діагностичних сигналів з використанням сервісів AWS

Мережева точка доступу до моделі

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Штаєр Л. О., к.т.н., доц. каф. ІТТС		

7. Дата видачі завдання листопад 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Термін виконання етапів роботи	Примітка
1	Аналіз методів контролю витоків та обґрунтування архітектури інтелектуальної системи	01.09.2025-25.10.2025	Виконано
2	Розроблення методичного та алгоритмічного забезпечення класифікації станів трубопроводу	26.10.2025-26.11.2025	Виконано
3	Реалізація інтелектуальної системи контролю витоків та оцінка її ефективності	27.11.2025-15.12.2025	Виконано
4	Оформлення результатів роботи	16.12.2025-22.12.2025	Виконано

Студент

Титиш Н. В.

(підпис)

(прізвище та ініціали)

Керівник роботи

Штаєр Л. О.

(підпис)

(прізвище та ініціали)

РЕФЕРАТ

Забезпечення цілісності трубопровідного транспорту є критичним завданням для енергетичної та екологічної безпеки. Існуючі методи детектування витоків часто демонструють нестабільну роботу при зміні експлуатаційних параметрів, таких як робочий тиск або частота тестового сигналу. Впровадження інтелектуальних систем на базі хмарних платформ (AWS SageMaker) дозволяє реалізувати масштабований моніторинг у режимі реального часу, що є вимогою сучасної концепції «Індустрії 4.0».

Наукова новизна отриманих результатів полягає у розробленні методу формування комбінованого вектора ознак, який дозволяє забезпечити інваріантність класифікатора до варіацій робочого тиску та несучої частоти сигналів та розробленні архітектури розгортання системи детектування витоків у хмарному середовищі, що забезпечує розділення логіки опрацювання даних та артефактів моделі.

Практична цінність роботи полягає у розробленні хмарного рішення, яке завдяки використанню сервісів AWS забезпечує автоматичне детектування стану трубопроводу з можливістю візуалізації аналітичних звітів для оперативного прийняття рішень. Результати роботи можуть бути використані при проєктуванні та модернізації систем автоматизованого контролю трубопровідних систем для підвищення їх надійності та екологічної безпеки.

АНОТАЦІЯ

Пояснювальна записка містить 84 сторінки, 31 рисунок, 18 джерел за переліком посилань.

Об'єкт дослідження – системи контролю витоків з трубопроводів.

Мета роботи – вдосконалення алгоритму опрацювання діагностичних сигналів в акустичному методі детектування витоків з трубопроводів шляхом розроблення методу класифікації станів, стійкого до зміни параметрів експлуатації, та реалізація інтелектуальної системи на базі AWS SageMaker.

Результатом роботи є розроблення методу класифікації станів з використанням комбінованого вектора ознак та ансамблевої моделі Random Forest, а також архітектури системи збору, опрацювання та аналізу звукових сигналів, які використовуються при виявленні витоків з трубопроводів.

Реалізація здійснена на базі сервісів хмарних обчислень AWS, зокрема, Amazon S3, AWS IAM, Amazon SageMaker.

У першому розділі проведено аналіз методів контролю витоків та алгоритмів опрацювання сигналів, зокрема, обґрунтовано вибір активного акустичного методу контролю витоків та запропоновано архітектуру інтелектуальної системи з використанням сервісів AWS.

У другому розділі представлено методичне та алгоритмічне забезпечення виявлення витоків, що включає розроблення алгоритму опрацювання сигналів для формування імпульсної характеристики трубопроводу та розроблення методу формування комбінованого вектора ознак для підвищення стійкості класифікатора до зміни робочого тиску та несучої частоти тестового сигналу.

У третьому розділі здійснено реалізацію, апробацію та оцінку ефективності системи, що включає реалізацію компонентів інтелектуальної системи на базі AWS та оцінку продуктивності ML-моделей.

Ключові слова: ДЕТЕКТУВАННЯ ВИТОКІВ, АКУСТИЧНИЙ КОНТРОЛЬ, SAGEMAKER, RANDOM FOREST, ХМАРНІ ОБЧИСЛЕННЯ.

ABSTRACT

The calculation and explanatory note contains 84 pages, 31 figures, 18 sources according to the list of references.

The object of the study is pipeline leak detection systems.

The purpose of the work is to improve the diagnostic signal processing algorithm in the acoustic leak detection method. This is achieved by developing a state classification method that is robust to changes in operating parameters, and by implementing an intelligent system based on AWS SageMaker.

The result of the work is the development of a state classification method using a combined feature vector and a Random Forest ensemble model, as well as the architecture of a system for collecting, processing, and analyzing sound signals used to detect pipeline leaks.

The implementation was carried out using AWS cloud computing services, specifically Amazon S3, AWS IAM, Amazon SageMaker.

The first chapter analyzes leak detection methods and signal processing algorithms. In particular, it justifies the choice of the active acoustic leak detection method and proposes the architecture of an intelligent system using AWS services.

The second chapter presents the methodological and algorithmic framework for leak detection. It includes the development of a signal processing algorithm to generate the pipeline impulse response and a method for creating a combined feature vector using entropy and relational components to increase the classifier's robustness to changes in operating pressure and the test signal's carrier frequency.

The third chapter covers the implementation, testing, and efficiency evaluation of the system, which includes the deployment of intelligent system components based on AWS and the performance assessment of the ML models.

Keywords: LEAK DETECTION, ACOUSTIC CONTROL, SAGEMAKER, RANDOM FOREST, CLOUD COMPUTING.

ЗМІСТ

ВСТУП.....	10
1 АНАЛІЗ МЕТОДІВ КОНТРОЛЮ ВИТОКІВ ТА ОБҐРУНТУВАННЯ АРХІТЕКТУРИ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ.....	12
1.1 Аналіз проблеми контролю технічного стану трубопроводів та класифікація витоків.....	12
1.2 Огляд існуючих методів виявлення витоків та алгоритмів опрацювання діагностичних сигналів.....	14
1.3 Обґрунтування вибору активного акустичного методу контролю витоків	21
1.4 Постановка задачі на розроблення інтелектуальної системи детектування витоків на базі AWS SageMaker	23
2 РОЗРОБЛЕННЯ МЕТОДИЧНОГО ТА АЛГОРИТМІЧНОГО ЗАБЕЗПЕЧЕННЯ КЛАСИФІКАЦІЇ СТАНІВ ТРУБОПРОВОДУ	25
2.1 Алгоритм попереднього опрацювання акустичних сигналів	25
2.2 Розроблення методу формування комбінованого вектора ознак з використанням ентропійних та реляційних складових.....	30
2.3 Обґрунтування вибору структури класифікатора для класифікації станів трубопроводу	38
2.4 Алгоритм детектування та локалізації витоків на основі різницевої імпульсної характеристики	44
3 РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ДЕТЕКТУВАННЯ ВИТОКІВ ТА ОЦІНКА ЇЇ ЕФЕКТИВНОСТІ	47

					КРМ.АКСм-21.00.00.000 ПЗ			
Зм.	Арк.	№ докум.	Підпис	Дата				
Розроб.	Титиш Н. В.				Інтелектуальна система детектування витоків у трубопроводах з використанням AWS SageMaker	Літ.	Арк.	Арк.шів
Перевір.	Штаер Л. О.						8	84
Реценз.						ІФНТУНГ АКСм-24-1		
Н. Контр.	Возний А.В.							
Затверд.	Заміховський							

3.1	Результати порівняльного аналізу ефективності нелінійних класифікаторів на основі комбінованого вектора ознак	47
3.2	Архітектура та конфігурація хмарного середовища AWS	51
3.3	Програмна реалізація підсистеми машинного навчання в середовищі Amazon SageMaker	59
3.4	Розгортання інтелектуального сервісу моніторингу	65
ВИСНОВКИ.....		70
ПЕРЕЛІК ПОСИЛАНЬ НА ДЖЕРЕЛА		71
Додаток А Програмний код модуля train.py.....		74
Додаток Б Програмний код модуля predict.py		78
Додаток В Програмний код ноутбука 02_Leak_detection_inference.ipynb		82

					КРМ.АКСм - 21.00.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми. Нафтогазовий комплекс України є стратегічною складовою енергетичної безпеки держави, що потребує впровадження високоефективних систем моніторингу цілісності трубопровідного транспорту. Витоки з трубопроводів спричиняють не лише значні економічні збитки, а й створюють ризики екологічних катастроф. Традиційні методи контролю часто демонструють низьку ефективність в умовах змінного робочого тиску та високого рівня техногенних шумів.

Широке впровадження концепцій «Індустрії 4.0» та «Інтернету речей» (IoT) обумовлює необхідність переходу від локальних діагностичних пристроїв до розподілених інтелектуальних систем. Використання хмарних платформ, зокрема Amazon Web Services (AWS), дозволяє реалізувати масштабовані алгоритми машинного навчання, здатні опрацьовувати великі масиви акустичних даних у режимі реального часу. Таким чином, розроблення методів класифікації станів трубопроводів, стійких до динамічних умов експлуатації, та їх реалізація у хмарному середовищі є актуальним науково-практичним завданням.

Мета дослідження — вдосконалення алгоритму опрацювання діагностичних сигналів в акустичному методі детектування витоків з трубопроводів шляхом розроблення методу класифікації станів, стійкого до зміни параметрів експлуатації, та реалізація інтелектуальної системи на базі AWS SageMaker.

Ідея роботи полягає у використанні комбінованого вектора ознак, що включає ентропійні та реляційні складові акустичного сигналу, та його опрацюванні ансамблевим класифікатором Random Forest у межах хмарної MLOps-архітектури, що забезпечує стабільність детектування витоків при зміні несучої частоти та тиску в системі.

					КРМ.АКСм - 21.00.00.000 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

Об'єкт дослідження — системи та процеси інтелектуального контролю витоків з трубопроводів.

Предмет дослідження — методи опрацювання діагностичних акустичних сигналів, алгоритми формування комбінованих векторів ознак, моделі машинного навчання для класифікації станів технічних об'єктів та архітектура хмарних сервісів для реалізації систем технічної діагностики.

Методи дослідження: методи теорії інформації та кодування для розрахунку ентропійних показників акустичних сигналів; методи машинного навчання (ML) для побудови та навчання моделі класифікації Random Forest.; технології хмарних обчислень (Cloud Computing) для проектування архітектури системи моніторингу на базі Amazon SageMaker.

					КРМ.АКСм - 21.00.00.000 ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ МЕТОДІВ КОНТРОЛЮ ВИТОКІВ ТА ОБҐРУНТУВАННЯ АРХІТЕКТУРИ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ

1.1. Аналіз проблеми контролю технічного стану трубопроводів та класифікація витоків

Трубопровідний транспорт є одним із найбільш безпечних та економічних способів переміщення природного газу, нафти та інших речовин [1]. Забезпечення надійного транспортування первинних енергоресурсів є стратегічно важливим завданням для держави, особливо для України, яка володіє однією з найбільших у Європі нафто- і газотранспортних систем [2]. Газотранспортна система (ГТС) України розглядається як критична інфраструктура, що має ключове значення для національної безпеки, економічної стабільності та геополітичного позиціонування. Історично ГТС України була головним транзитним шляхом постачання газу до Європейського Союзу, хоча обсяги транзиту суттєво знизились і були фактично припинені з 2025 року внаслідок агресії РФ та зміни енергетичних пріоритетів [3]. Попри це, завдяки модернізації та технічній переорієнтації система зберегла можливість імпорту газу з країн ЄС.

Контроль технічного стану трубопроводів є необхідною складовою системи технічної діагностики, мета якої — попередження, пошук та локалізація аварійних станів елементів трубопровідних систем.

Основною причиною зміни технічного стану трубопроводів є поява витоків, що призводить до значних екологічних та економічних втрат, а також часто супроводжується людськими втратами [4]. Причини відмов магістральних трубопроводів поділяють на три основні групи [2]:

— техногенні: внутрішня корозія труб, дефекти зварювальних швів, дефекти металургійного походження;

					KPM.AKCM - 21.00.00.000 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

— природні: землетруси, зсуви ґрунту, аномальні температурні або метеорологічні умови;

— антропогенні: помилки експлуатації, пошкодження під час землерийних робіт та несанкціонований доступ (врізання) або диверсія.

Через зовнішні фактори та втручання трубопроводи особливо схильні до пошкоджень. За висновками Європейської Групи (EGIG), втручання сторонніх осіб спричиняє близько 50 % аварійних витоків [1].

Аварійні витoki традиційно поділяються на "великі" та "малі" [1,2]. Малі витoki (або свищі) — це ті, відносний розмір яких не перевищує 1% від номінальної пропускної здатності трубопроводу. Це відповідає діаметру 5–10 мм, типовому для наскрізних корозійних дефектів. Основна проблема полягає у виявленні саме малих витоків, оскільки їхній вплив на технологічні параметри роботи трубопроводу, такі як тиск, є несуттєвим (наприклад, максимальна зміна тиску газу може становити лише 0,05 МПа). Для рідин, таких як нафта, виявлення малих витоків ускладнюється тим, що отвори можуть затягуватися шаром парафіну та механічних домішок, що містяться в продукті, якщо розмір отвору менший за певний мінімум [1].

Системи детектування (виявлення) витоків повинні відповідати низці вимог, включаючи чутливість (здатність швидко виявляти малі витoki), надійність, точність і стійкість до неідеальних умов. Незважаючи на значну кількість розробок, існуючі методи часто не задовольняють вимогам щодо точності, оперативності локалізації та здатності виявляти малі витoki [1,2]. Достовірність традиційних методів, зокрема акустичного контролю, суттєво залежить від зміни технологічних умов транспортування. Через це виникає критична потреба у розробці вдосконалених підходів до формування ознак стану, які будуть стійкими до експлуатаційної невизначеності. Сучасні напрями досліджень зосереджені на впровадженні інтелектуальних рішень, таких як машинне навчання (ML) та штучний інтелект (AI) [5] для прогнозного технічного обслуговування та раннього виявлення аномалій. Використання AI

					KPM.AKCM - 21.00.00.000 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

забезпечує можливість цілодобового обслуговування, що знижує навантаження на персонал та підвищує ефективність роботи. Постійний моніторинг ГТС, зокрема із застосуванням інноваційних технологій, є також однією з вимог європейських стандартів, що сприяє підвищенню довіри міжнародних партнерів [3].

1.2. Огляд існуючих методів виявлення витоків та алгоритмів опрацювання діагностичних сигналів

Забезпечення надійної експлуатації трубопроводів вимагає постійного вдосконалення методів контролю витоків, оскільки існуючі системи можуть бути дорогими у впровадженні або мати недостатню чутливість, особливо для виявлення "малих" витоків. Системи детектування витоків мають першочергове значення для запобігання фінансовим втратам та екологічним катастрофам [1].

Методи контролю витоків традиційно поділяються на дві основні групи за принципом збору даних:

— внутрішні / обчислювальні методи: ґрунтуються на моніторингу внутрішніх параметрів трубопроводу, таких як тиск, потік та температура, за допомогою польових вимірювальних приладів. Вони працюють на основі алгоритмічних або математичних моделей;

— зовнішні методи: використовують локальні, спеціалізовані датчики (наприклад, інфрачервоні камери, акустичні мікрофони, оптоволоконні системи) для контролю зовнішніх параметрів трубопроводу.

Детальнішу класифікацію методів контролю разом з пояснення суті методу наведено на рис. 1.1.

					КРМ.АКСм - 21.00.00.000 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 1.1 – Методи контролю/детектування витоків

Активні методи контролю (на основі тестових сигналів) передбачають зовнішній вплив на систему з подальшим аналізом її реакції.

Сутність методу гідравлічних тестів [2] полягає у застосуванні контрольованих змін фізичних параметрів робочого середовища (рідини або газу) всередині трубопроводу та моніторингу результуючих відгуків. Зазвичай це включає вимірювання динаміки тиску та витрати на різних ділянках системи. У герметичному трубопроводі тиск і витрата мають узгоджену залежність, що описується гідродинамічними рівняннями. При виникненні витoku в системі створюється додатковий вихідний потік, який порушує цей баланс. Аналіз хвилі тиску, що поширюється вздовж трубопроводу після імпульсного тестового сигналу, дозволяє локалізувати місце аномалії, оскільки витік виступає як точка відбиття хвилі, змінюючи її амплітуду та час проходження.

Сутність акустичних методів полягає у використанні фізичного явища виникнення звукових (від 20 Гц до 20 кГц) та ультразвукових (вище 20 кГц) коливань у місці негерметичності [2]. При проходженні робочого середовища крізь малий отвір виникає високошвидкісний потік (турбулентність), який

генерує характерний шум. Спеціалізовані акустичні давачі або гідрофони розміщуються на поверхні трубопроводу або в контакті з рідиною. Шум витoku реєструється двома або більше давачами [2]. Використовуючи метод кореляції часу затримки, місце витoku L визначається на основі різниці часу Δt надходження звуку до двох давачів, відстані D між ними та швидкості звуку c у середовищі:

$$L = \frac{1}{2}(D - c * \Delta t), \quad (1.1)$$

де L – відстань від першого давача до витoku.

Пасивні методи передбачають постійний моніторинг природних фізичних або хімічних ознак, які є прямим наслідком витoku.

Метод аналізу змін параметрів включає безперервний моніторинг основних технологічних параметрів системи для виявлення відхилень від нормального стану (аналіз відхилень від норми). Принцип дії полягає у використанні системи SCADA (Supervisory Control and Data Acquisition) та LDS (Leak Detection System) для збору даних про тиск, температуру, витрату, щільність тощо. Витік спричиняє раптову чи поступову зміну цих параметрів. Наприклад, балансовий метод – це порівняння вхідної та вихідної витрати. Якщо вхідна витрата Q_{in} перевищує вихідну Q_{out} на величину, що більша за допустиму похибку, фіксується витік:

$$Q_{leak} = Q_{in} - Q_{out} - Q_{storage}. \quad (1.2)$$

У методі градієнту тиску ведеться постереження за несподіваним зниженням градієнта тиску між двома точками. У методі аналізу складу навколишнього середовища проводиться моніторинг хімічного складу ґрунту, ґрунтових вод або повітря навколо трубопроводу на наявність компонентів робочого середовища.

Існує метод, що полягає у використанні давачів на витоках. Він полягає у фізичному розміщенні спеціалізованих давачів у потенційних або відомих місцях ризику (наприклад, у місцях з'єднань, відводів, під землею вздовж траси). Ці давачі безпосередньо реагують на присутність витікаючої речовини.

					КРМ.АКСм - 21.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

Прикладами можуть бути оптоволоконні давачі (використання кабелів, що реагують на контакт з рідиною або напругу); хімічні давачі (детектори газу (наприклад, метану, водню, вуглеводнів), що розміщуються в ґрунті або повітрі та спрацьовують при перевищенні порогової концентрації); сенсорні стрічки/кабелі — спеціальні провідні матеріали, електричний опір яких різко змінюється при контакті з рідиною.

Група теплових методів передбачає використання тепловізійних систем для реєстрації температурних аномалій, спричинених витоком. За принципом дії витік рідини або газу, що має температуру, відмінну від температури навколишнього середовища, призводить до локальної зміни теплового поля. Розширення газу при витoku зазвичай викликає значне локальне охолодження. Рідинні витки можуть викликати як охолодження (випаровування), так і нагрівання (якщо рідина гаряча). Інфрачервоні камери (тепловізори), розміщені на літальних апаратах або ручних пристроях, реєструють ці зміни, візуалізуючи витік.

Метод реєстрування звукових коливань від витoku є пасивним варіантом акустичного контролю [2], зосередженим на безперервній реєстрації природних шумів витoku без введення тестового сигналу. Принцип дії: давачі встановлені вздовж траси і постійно прослуховують характерний шум, що генерується високошвидкісним потоком рідини/газу через дефект. Дані про шум збираються і піддаються частотному аналізу та фільтрації для відділення шуму витoku від фонових шумів (наприклад, від роботи насосів, руху транспорту). Цей метод підходить для моніторингу повільних витоків у режимі реального часу.

В контексті сучасних технологій також використовується класифікація за типом обробки даних: апаратні (використовують фізичні пристрої), програмні, інтелектуальні та гібридні [5]. При цьому достовірність відомих методів, включаючи акустичний контроль, суттєво залежить від зміни технологічних умов транспортування.

Серед внутрішніх (обчислювальних) методів найбільш поширеними є:

					KPM.AKCM - 21.00.00.000 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

— системи моніторингу на основі SCADA та CPM. Система Supervisory Control And Data Acquisition (SCADA) є найпоширенішим методом, який відстежує такі дані, як тиск, швидкість потоку та температура [6,7]. Додатково застосовується система обчислювального моніторингу трубопроводу (Computational Pipeline Monitoring, CPM), яка базується на фундаментальних наукових принципах, зокрема, законах збереження енергії, маси та імпульсу;

— метод балансу маси. Ґрунтується на принципі збереження маси (потік на вході повинен дорівнювати потоку на виході в трубопроводі без витоків). Будь-який значний дисбаланс вказує на витік [2]. Хоча цей метод є простим і недорогим, він нечутливий до малих витоків, особливо у великих трубопроводах. Існують вдосконалені методи, такі як компенсований масовий баланс [6];

— метод аналізу хвилі негативного тиску. При виникненні витoku генерується хвиля розширення (хвиля негативного тиску), яка поширюється від місця витoku в обидва боки зі швидкістю звуку в середовищі [2,8]. Локалізація витoku здійснюється шляхом аналізу різниці часу прибуття хвилі до датчиків і відомої швидкості її поширення. Хоча метод має високу чутливість, його точність знижується через високий рівень шуму та обмежену роздільну здатність датчиків тиску [8];

— моделі перехідних процесів. Моделі перехідних процесів у реальному часі (Real-Time Transient Model, RTTM) використовують математичні моделі потоку, ґрунтовані на фізичних законах збереження маси, імпульсу та енергії, дозволяючи розраховувати параметри потоку в будь-якій точці в реальному часі [6]. Розширені моделі E-RTTM інтегрують RTTM зі статистичними методами, що дозволяє уникнути хибних спрацювань і підвищити чутливість [6].

Акустичні та зовнішні методи поділяються за наступним принципом:

— акустичні методи. Базуються на реєстрації та аналізі звуків або ультразвукових коливань, які генеруються рідиною чи газом, що витікає через

					KPM.AKCM - 21.00.00.000 ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

отвір [2,8]. Вони можуть бути активними (зі створенням тестових коливань у середовищі) або пасивними (на основі акустичної емісії, АЕ);

— метод кореляції шуму витoku. Встановлює місцезнаходження витoku шляхом порівняння шумів, зафіксованих двома давачами [2]. Різниця в часі приходу шуму до давачів (часова затримка), отримана за допомогою взаємнокореляційної функції (ВКФ), дозволяє обчислити відстань до витoku. Кореляційний аналіз є ефективним, але обмежений високим рівнем фонового шуму та швидким згасанням сигналу в трубах, особливо неметалевих [8];

— опрацювання акустичних сигналів. Використовуються вдосконалені алгоритми, такі як фазове перетворення (Phase Transform, PHAT) для покращення придушення шумів та точності визначення часової затримки (Time Delay Estimation, TDE) [9];

— оптоволоконні системи. Встановлення кабелів уздовж трубопроводу для моніторингу змін температури (Distributed Temperature Sensing, DTS) або вібрацій (Distributed Acoustic Sensing, DAS) [2,6]. Ці системи забезпечують моніторинг у реальному часі;

— інспекційні методи. Включають візуальний огляд (людьми, дронами або навченими собаками) та внутрішню діагностику за допомогою інтелектуальних поршнів (PIGs), які здійснюють повноцінну діагностику стану труби зсередини [2,3].

Завдяки розвитку технологій промислового інтернету речей (IIoT), який є підмножиною Інтернету речей (IoT), стало можливим впровадження інтелектуальних систем моніторингу, які забезпечують цілодобову роботу та підвищують ефективність [7]. Методи, що використовують машинне навчання (ML) та штучний інтелект (AI) поділяються за наступними ознаками:

— класифікація та регресія. Використовуються штучні нейронні мережі (ШНМ/ANN), алгоритми опорних векторів (SVM), дерева рішень (DT) та випадковий ліс (Random Forest, RF) [1];

					KPM.AKCM - 21.00.00.000 ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

— глибоке навчання (Deep Learning, DL). Використовуються архітектури згорткових нейронних мереж (CNN) та рекурентні нейронні мережі (RNN/LSTM) для аналізу часових рядів або зображень (спектрограм) акустичних сигналів [8,10];

— інженерія ознак. Для навчання моделей ML необхідно одержати з сигналів інформативні ознаки у часовій та частотній областях, такі як середньоквадратичне значення, асиметрія, пікове значення, ентропія та спектральні характеристики [4]. Ентропійні оцінки демонструють перспективність для підвищення ефективності розпізнавання, оскільки зменшення потужності сигналу мало впливає на апертуру оцінок ентропії [1];

— гібридні підходи. Комбінують переваги різних методів для підвищення точності та стійкості до змін режиму експлуатації [1]. Зокрема, система PipePatrol об'єднує технологію динамічного моделювання в режимі реального часу (балансовий метод) із модулем розпізнавання моделей витоків [1].

Узагальнена характеристика методів детектування витоків щодо алгоритму опрацювання сигналів наведена на рис. 1.2.

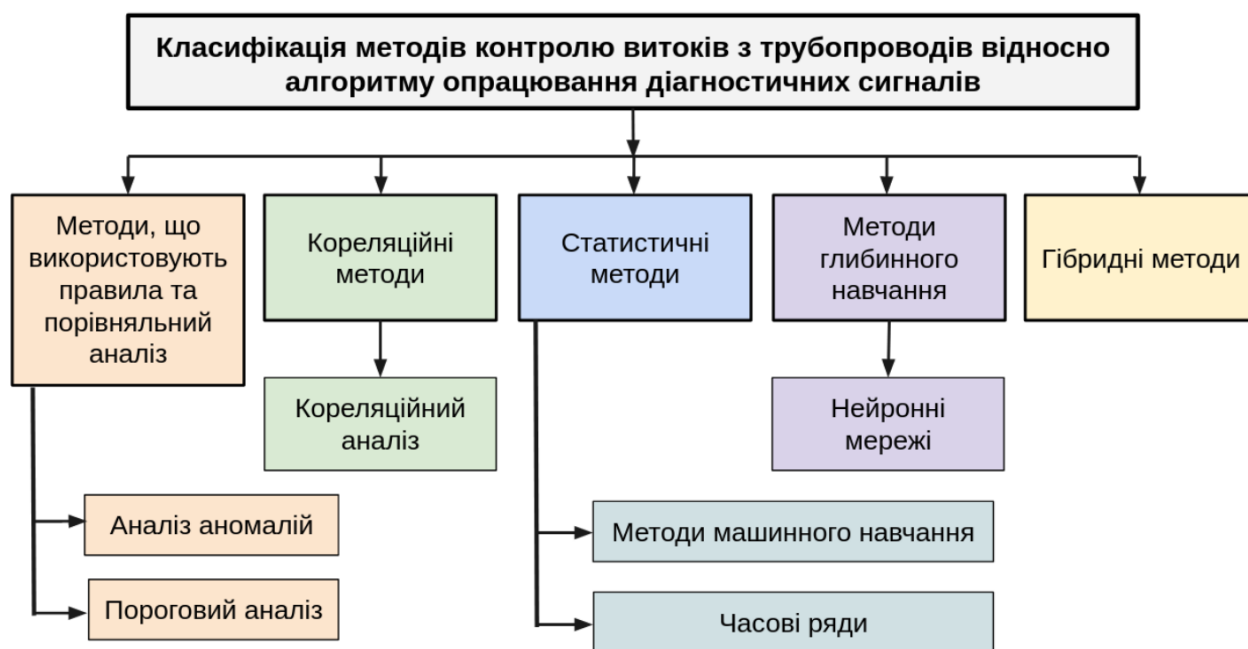


Рисунок 1.2 – Алгоритми опрацювання діагностичної інформації в системах детектування витоків

Незважаючи на значні досягнення, більшість ML-моделей мають обмеження: вони вимагають великих обсягів якісних навчальних даних [8], часто схильні до перенавчання (overfitting), а їхня висока обчислювальна складність обмежує застосування в режимі реального часу на обладнанні з обмеженими ресурсами. Ці проблеми визначають необхідність подальших досліджень, спрямованих на розширення інформативних складових діагностичних сигналів [1] та розроблення методів класифікації, які є стійкими до експлуатаційної невизначеності (змін тиску, температури та характеристик тестового сигналу).

1.3. Обґрунтування вибору активного акустичного методу контролю витоків

Вибір активного акустичного методу контролю витоків для інтелектуальної системи обґрунтований його потенціалом до дистанційного виявлення та здатністю забезпечувати високоякісну діагностичну інформацію, незалежно від середовища транспортування [2,10,11].

Переваги активного акустичного методу полягають в тому, що він є активним методом, який ґрунтується на створенні тестових коливань у середовищі заповнення трубопроводу [11]. Це дозволяє фіксувати відбиті хвилі від неоднорідностей трубопроводу (наприклад, витоків, запірної арматури або відгалужень). Основна перевага методу полягає у прямій взаємодії з середовищем транспортування через гідроакустичний канал, що дозволяє реєструвати найменш спотворену діагностичну інформацію порівняно з іншими способами контролю [2]. Для підвищення ефективності виявлення використовується генерування завадостійких сигналів (зокрема, код Баркера або ЛЧМ-сигнали), які у поєднанні з кореляційною обробкою значно підвищують співвідношення «сигнал/шум» і збільшують відстань сканування (дальність виявлення витоку). Кінцевим діагностичним результатом є імпульсна

					КРМ.АКСм - 21.00.00.000 ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

характеристика трубопроводу (IX), яка формується шляхом обчислення взаємкореляційної функції (ВКФ) зареєстрованого та тестового сигналів та її обвідної (за допомогою перетворення Гільберта) [1,2]. Факт виявлення витoku та його координата встановлюється шляхом віднімання експериментальної IX від еталонної (отриманої за відсутності витoku), формуючи різницеву імпульсну характеристику. Координата витoku визначається за часом затримки приходу відбитого сигналу, встановленим за максимумом ВКФ.

Переваги над альтернативними методами можуть бути визначені в наступному:

— на противагу традиційним обчислювальним методам (балансу маси або моніторингу тиску), акустичний метод контролю є менш схильним до проблеми низької чутливості до «малих» витоків (втрати менше 1% від номінальної пропускної здатності) [1];

— на відміну від пасивних методів акустичної емісії (АЕ), які реєструють шум витoku (сигнали АЕ швидко загасають, обмежуючи дальність контролю), акустичний метод контролю використовує поздовжні акустичні сигнали, які згенеровані в середовищі заповнення трубопроводу. Відповідно, відомі як вигляд тестового сигналу, так і його параметри (частота несучої, форма огибаючої і т.д.);

— обраний метод дозволяє здійснювати контроль за наявності одностороннього доступу до середовища транспортування, що є перевагою над іншими кореляційними методами, які вимагають дві точки контролю [11].

Незважаючи на переваги, традиційні акустичні методи мають критичний недолік — їхня достовірність суттєво залежить від зміни технологічних умов транспортування (тиску, температури, характеристик тестового сигналу). Для підвищення стійкості до цієї експлуатаційної невизначеності пропонується вдосконалити алгоритм, який описано в [2]. Це досягається шляхом дослідження можливості розширення вектора ознак акустичного сигналу складовими, що дозволяють компенсувати синфазні завади та підвищити стійкість класифікатора

					КРМ.АКСм - 21.00.00.000 ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

до змін робочого тиску і характеристик сигналу. Розроблення інтелектуальної системи, побудованої на базі AWS SageMaker, буде використовувати методи машинного навчання (ML) для класифікації станів трубопроводу на основі досліджених та обраних стійких ознак, які дозволять підщидити стійкість до змінних умов експлуатації трубопроводу.

1.4. Постановка задачі на розроблення інтелектуальної системи детектування витоків на базі AWS SageMaker

Незважаючи на значну кількість розробок, існуючі методи контролю, включаючи акустичний, часто не задовольняють вимогам щодо точності, оперативності локалізації та здатності виявляти «малі» витoki (втрата 1% від номінальної пропускної здатності). Акустичні методи, які ґрунтуються на створенні та реєстрації тестових коливань, мають високий потенціал, але їхня достовірність суттєво залежить від зміни технологічних умов транспортування (експлуатаційної невизначеності). Ця складність посилюється високим рівнем завад і малою амплітудою корисного сигналу від витoku, що призводить до низької надійності класифікації.

Необхідність мінімізації економічних та екологічних збитків вимагає розроблення підходів до формування ознак стану, які є стійкими до експлуатаційної невизначеності. Це зумовлює потребу у впровадженні Промислового Інтернету речей (IIoT), оскільки такі підходи дозволяють створювати системні архітектури, які є більш ощадливими, гнучкими, продуктивними та ефективними порівняно з класичною промисловою автоматизацією.

Метою роботи є удосконалення алгоритму опрацювання діагностичних сигналів в акустичному методі контролю витоків з трубопроводів шляхом розроблення методу класифікації станів, стійкого до зміни параметрів

					KPM.AKCM - 21.00.00.000 ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

експлуатації, та реалізація цієї моделі як інтелектуальної системи на базі AWS SageMaker.

Для досягнення поставленої мети необхідно вирішити такі основні завдання:

— розроблення стійкого алгоритму формування ознак: обґрунтувати та розробити вдосконалений метод формування вектора ознак, який має забезпечити компенсацію синфазних завад, спричинених змінами робочих умов та бути стійким до зміни параметрів тестового сигналу;

— провести вибір та оптимізацію ML-моделі: здійснити вибір оптимальної архітектури класифікатора, здатної до ефективного узагальнення на нових, раніше невідомих даних. Це включає порівняльний аналіз лінійних, нелінійних та ансамблевих методів (Random Forest);

— спроектувати архітектуру на базі AWS: розробити масштабовану, надійну та безпечну архітектуру системи IIoT з використанням сервісів AWS Cloud. Архітектура має передбачати використання Amazon S3 для зберігання діагностичних wav-файлів, AWS Lambda для попереднього опрацювання сигналів та Amazon SageMaker для тренування, розгортання і використання моделей ML;

— реалізувати програмні та алгоритмічні компоненти системи та провести дослідження, включаючи тестування на незалежній вибірці даних, щоб підтвердити стійкість класифікатора до зміни робочого тиску та характеристик тестового сигналу.

					КРМ.АКСм - 21.00.00.000 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

2 РОЗРОБЛЕННЯ МЕТОДИЧНОГО ТА АЛГОРИТМІЧНОГО ЗАБЕЗПЕЧЕННЯ КЛАСИФІКАЦІЇ СТАНІВ ТРУБОПРОВОДУ

2.1 Алгоритм попереднього опрацювання акустичних сигналів

Попереднє опрацювання акустичних сигналів розпочинається зі збору та збереження діагностичних даних, які мають формат wav-файлів і фіксуються акустичними давачами.

Схематичне представлення експериментального стенду, який було використано для генерування тестових сигналів та збору діагностичних сигналів показано на рис. 2.1 [2]. Позначення на рис. 2.1 наступні: А, В – точки підключення вставки для генерування та реєстрації імпульсів; М – манометр; С, D та Е – ділянки, які підлягають заміні на ділянку з вмонтованим імітатором витоку.

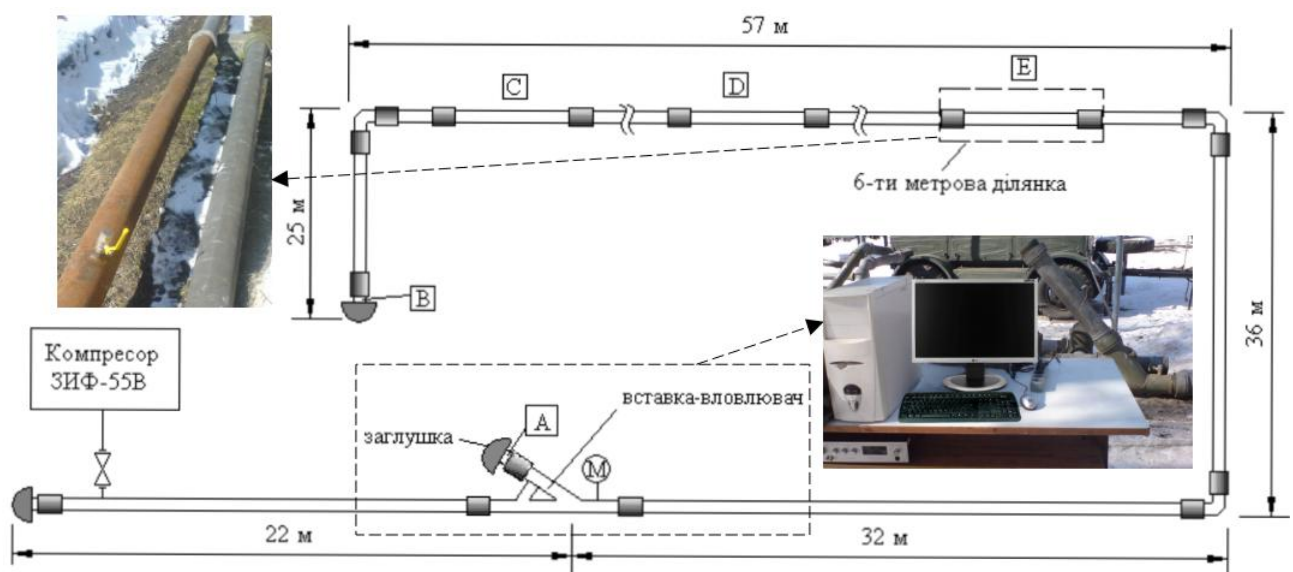


Рисунок 2.1 – Схема експериментального стенду для реєстрації акустичних сигналів

Для реєстрації акустичного відгуку системи використовувалась два акустичні давачі (мікрофони МКЭ-3), які були розташовані в середовищі

					КРМ.АКСм - 21.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

заповнення трубопроводу на відстані 0,442 м один від одного. Аналогові сигнали з мікрофонів оцифровані за допомогою звукової карти ESI Juli@ з частотою дискретизації 192 кГц (24-біт). Управління експериментом (генерування тестових сигналів та запис даних) здійснювались за допомогою комп'ютера (Celeron 1,5 ГГц, 1024 Мб RAM) з звуковою картою ESI Juli@ та підсилювача Radiotekhnika У-101.

Існуючий алгоритм [2] опрацювання діагностичних сигналів в акустичному методі контролю витоків з трубопроводів включав формування імпульсної характеристики трубопроводу на основі нормування ВКФ зареєстрованого сигналу з тестовим для моменту початку контролю. Наявність витоку вносить додаткові ознаки відбиття до зареєстрованого сигналу, які ідентифікуються шляхом обчислення ВКФ тестового та реєстрованого сигналу та побудовою різницевої характеристик між імпульсною характеристикою та одержаною. Підвищення стійкості алгоритму досягалось з використання завадостійкого кодування тестового сигналу та уточненням швидкості поширення акустичних коливань в середовищі заповнення трубопроводу шляхом побудови ВКФ між каналами реєстрування відбитих сигналів та відомій відстані встановлення давачів.

На архітектурному рівні для виконання початкових етапів, таких як перетворення формату, нормалізація даних або фільтрування, використаний сервіс Amazon Lambda, який автоматично запускається при потраплянні файлів до сховища Amazon S3. Цей етап є критичним, оскільки в умовах трубопровідного транспорту корисний сигнал часто має малу амплітуду на фоні високого рівня завад.

Для підвищення завадостійкості інформації про неоднорідності в трубопровідній системі при генеруванні первинних хвиль (тестовий сигнал) використовують фазоманіпульований код Баркера (завадостійкі сигнали). Використання завадостійких сигналів (наприклад, коду Баркера) як тестових додатково підвищує співвідношення «сигнал/шум» завдяки властивості

					КРМ.АКСм - 21.00.00.000 ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

зберігати відношення максимуму автокореляційної функції до значення у всіх інших точках на рівні розрядності коду. Приклад тестового сигналу показано на рис. 2.2.

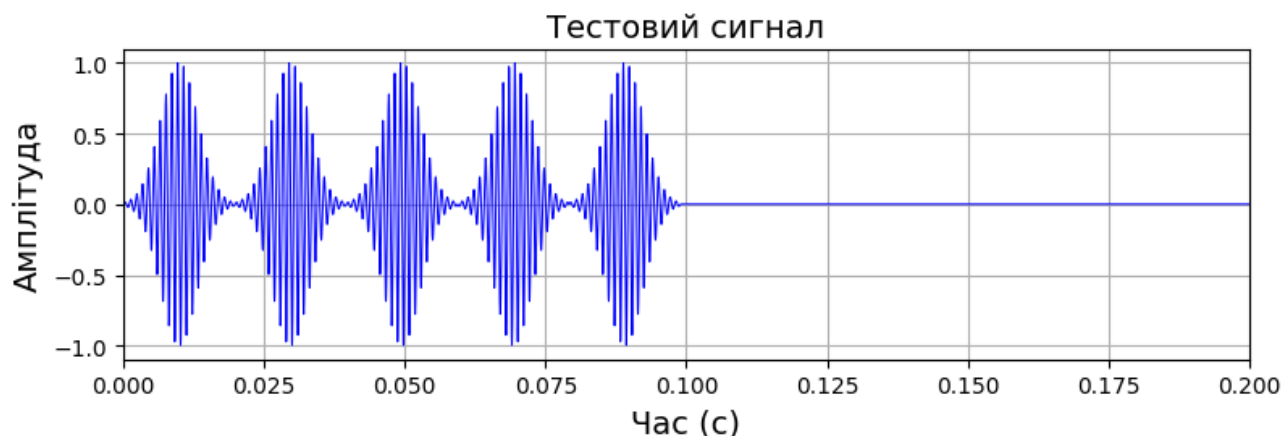


Рисунок 2.2 – Тестовий сигнал у формі коду Баркера

Для підвищення співвідношення «сигнал/шум», яке є необхідною умовою для якісної ідентифікації витоків, застосовуються кореляція з вхідним сигналом. Кореляція допомагає усунути випадковий шум, оскільки він некорельований із переданим сигналом. Використання додаткового мікрофону, розташованого на відстані 0,442 м від іншого мікрофона, дозволяє визначити (уточнити) швидкість поширення акустичних сигналів при зміні умов транспортування (рис. 2.3).

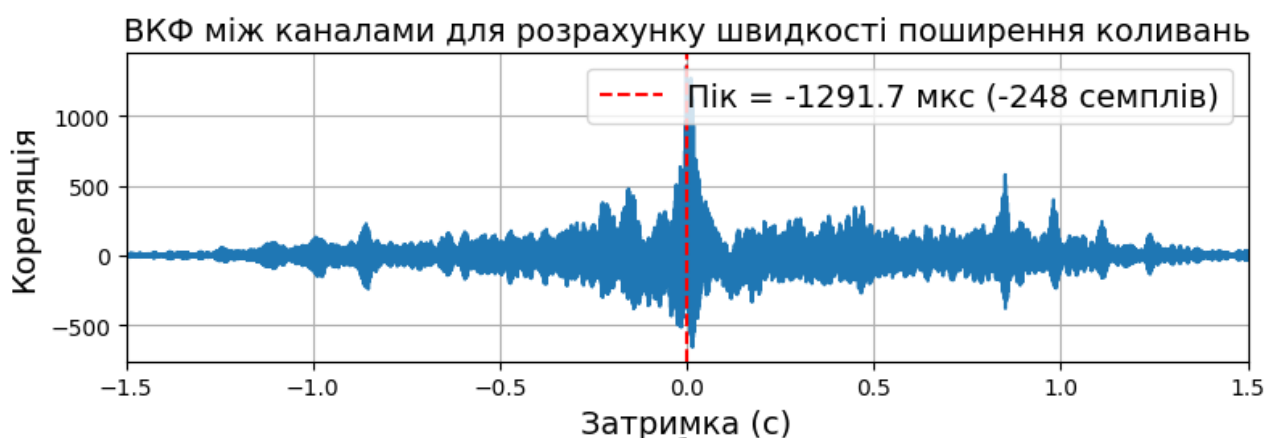
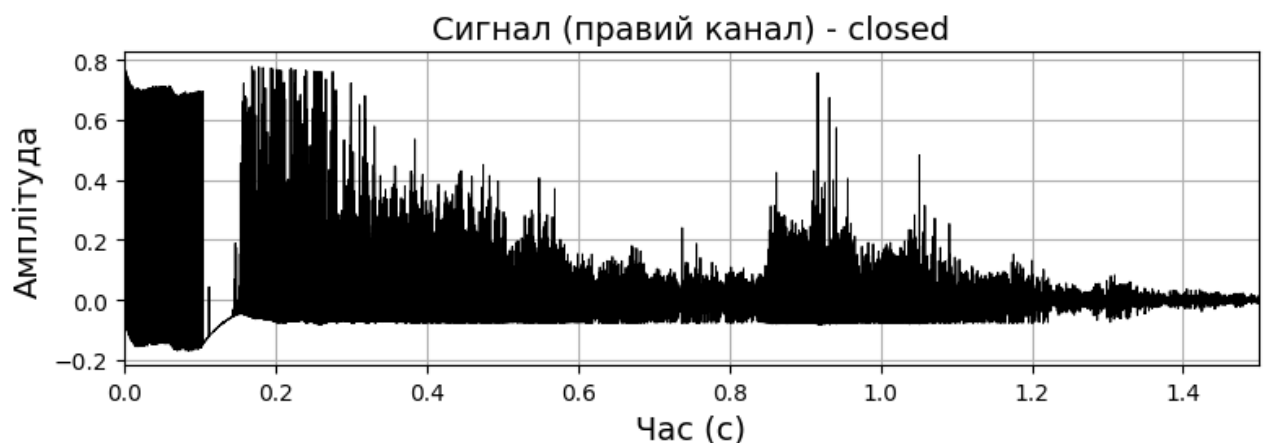


Рисунок 2.3 – Взаємкореляційна функція між каналами реєстрування (два мікрофони)

Формування імпульсної характеристики трубопроводу (ІХ) відбувається шляхом обчислення взаємкореляційної функції (ВКФ) тестового сигналу та зареєстрованого сигналу [2]. Приклад реєстрованого сигналу з правого каналу для випадку закритого трубопроводу (відсутній витік) воказано на рис. 2.4, а.



а



б

Рисунок 2.4 – Зареєстрований сигнал (а) та ВКФ тестового сигналу та зареєстрованого при відсутності витоку (стан трубопроводу «закрито»)

Для однозначного визначення координати неоднорідності (витоку) застосовується перетворення Гільберта. Модуль перетворення Гільберта перетворює осциляційну форму імпульсу в гладку аналітичну функцію (обвідну), що має тільки один максимум. Ця обвідна є імпульсною характеристикою трубопроводу (ІХ) (рис. 2.4, б). Пошук локальних максимумів

модуля перетворення Гільберта дозволяє визначити час запізнення відбитого сигналу. Час запізнення, встановлений за максимумом ВКФ, використовується для обчислення відстані до витоку.

Приклад випадку реєстрування відбитого сигналу з правого каналу та обчислення ВКФ показано на рис. 2.5 (рисунки а та б відповідно).

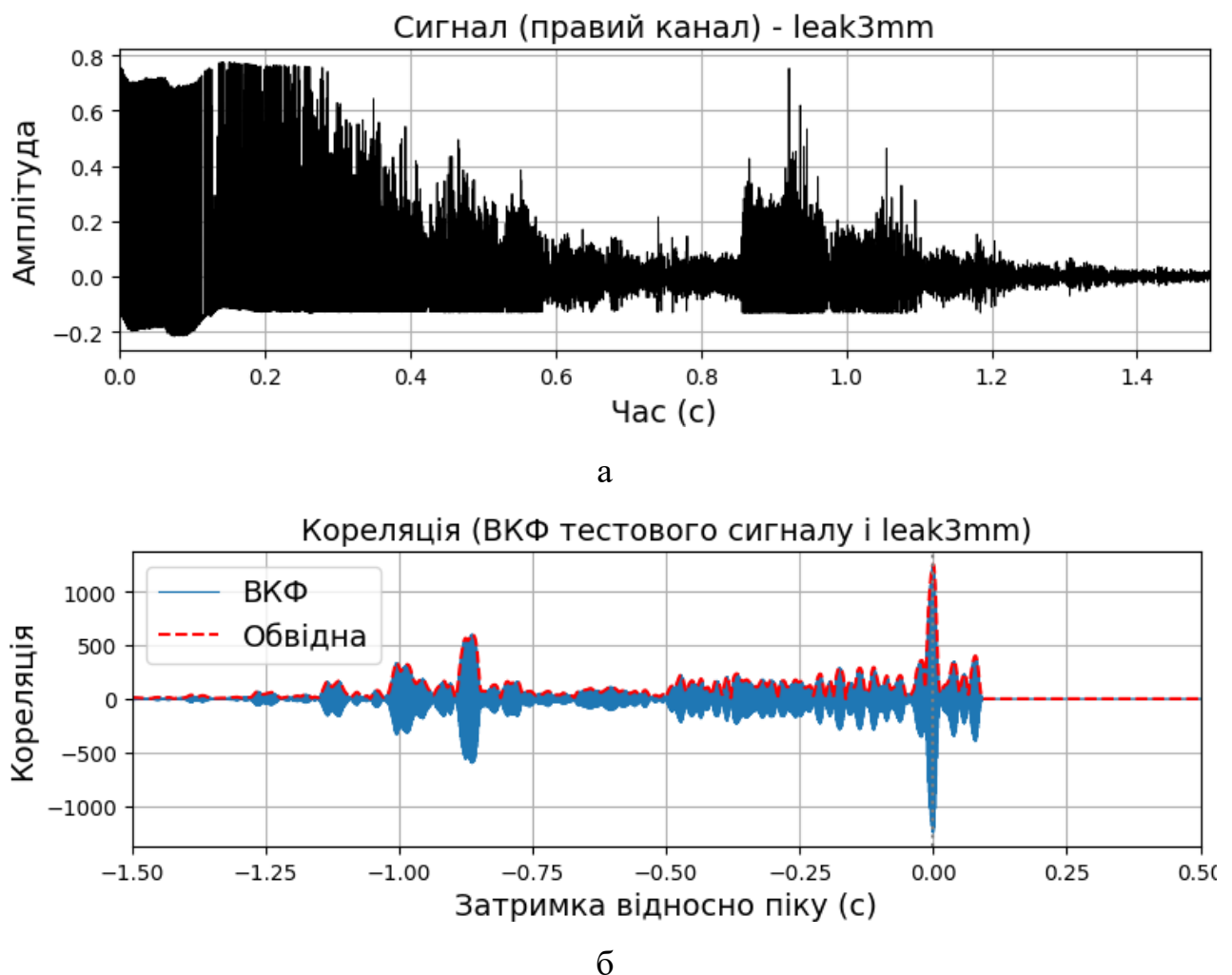


Рисунок 2.5 – Зареєстрований сигнал (а) та ВКФ тестового сигналу та зареєстрованого при витоку розміром 3 мм

З наведених рис. 2.4 та 2.5 видно, що алгоритм виявлення витоку потребує вдосконалення. Достовірність акустичного методу контролю витоків [2,11] суттєво залежить від зміни технологічних умов транспортування (експлуатаційної невизначеності), таких як тиск і температура, що ускладнює

надійну ідентифікацію витоків. Для подолання цієї експлуатаційної невизначеності та вирішення задачі класифікації акустичних сигналів необхідне застосування інтелектуальних методів. Методи машинного навчання (ML) та штучного інтелекту (AI) є ключовими для розроблення алгоритму класифікації станів, який буде стійким до зміни параметрів експлуатації [12–15]. Моделі ML, зокрема ансамблеві методи, можуть використовувати розширені вектори ознак для компенсації синфазних завад, спричинених зміною робочих умов, та підвищення здатності класифікатора до ефективного узагальнення на нових даних. Це забезпечує можливість організації автоматизованого моніторингу та прогнозного технічного обслуговування, що знижує навантаження на персонал і підвищує ефективність роботи. Для реалізації такої системи необхідне використання AWS SageMaker для тренування, розгортання та використання моделей машинного навчання [16].

2.2 Розроблення методу формування комбінованого вектора ознак з використанням ентропійних та реляційних складових

Ідея роботи полягає у підвищенні стійкості алгоритму контролю стану трубопроводу за допомогою методів машинного навчання (ML) для класифікації його можливих станів в умовах експлуатаційної невизначеності. Для порівняння ефективності класифікаторів були обрані три архітектури: лінійна (перцептрон), нелінійна (багатошаровий перцептрон, MLP) та ансамблева (Random Forest).

Дані для навчання та тестування були зібрані для моделювання п'яти різних станів трубопроводу:

- герметичний стан (closed) — відсутність витoku;
- чотири стани для витoku різного діаметру (leak1mm, leak3mm, leak5mm, leak10mm). Витоки імітувалися через отвори діаметрами 1 мм, 3 мм, 5 мм та 10 мм відповідно (рис. 2.1).

					КРМ.АКСм - 21.00.00.000 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

Для забезпечення різноманітності даних та перевірки стійкості моделей акустичні сигнали були відібрані ті, які реєстровані за змінних умов експлуатації. Експлуатаційна невизначеність створена шляхом використання даних з варіацією параметрів тестового сигналу та робочого тиску [2]: тестовий сигнал у записах генерувався за принципом завадостійкого кодування (використовувався 5-ти розрядний код Баркера). Його характеристики змінювалися через:

- обвідну сигналу (прямокутна, Гаусова, синусоїдна форми);
- частоту несучої (використовувалися значення 480 та 960 Гц).

Зміна робочого тиску присутня у записах, які було зареєстровано як окремі серії записів сигналів для всіх модельованих станів при різних значеннях тиску (відсутній тиск; 1; 1,4; 1,6 атмосфер) [2].

Такий підхід дозволив отримати експериментальні дані, необхідні для розроблення методу класифікації станів, стійкого до зміни параметрів експлуатації.

Для розроблення методу формування вектору ознак для навчання моделей машинного навчання було розроблено програмний модуль мовою Python 3.10 (наведено в додатку А). На рис. 2.6 представлена схема інформаційної взаємодії компонентів під час етапу навчання класифікатора. Процес реалізовано за принципом "чорної скриньки" (модуль train.py), який перетворює вхідні неопрацьовані дані та налаштування у готові до використання артефакти моделі та аналітичні звіти.

Архітектура процесу включає три основні блоки:

— блок вхідних даних та конфігурації: аудіо файли — вхідний датасет, що складається з колекції .wav файлів, розмічених за п'ятьма класами станів трубопроводу (closed, leak1mm, leak3mm, leak5mm, leak10mm); конфігураційні параметри — набір гіперпараметрів та прапорців (USE_ENTROPY_FEATURES, USE_RATIO_FEATURES, CLASSIFIER_TYPE), які дозволяють гнучко керувати експериментом, вмикаючи або вимикаючи певні групи ознак та обираючи тип алгоритму; бібліотека utils — допоміжний модуль, що містить реалізацію

					KPM.AКСм - 21.00.00.000 ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

функцій попереднього опрацювання сигналів, синхронізації стереоканалів та математичних алгоритмів вилучення ознак;

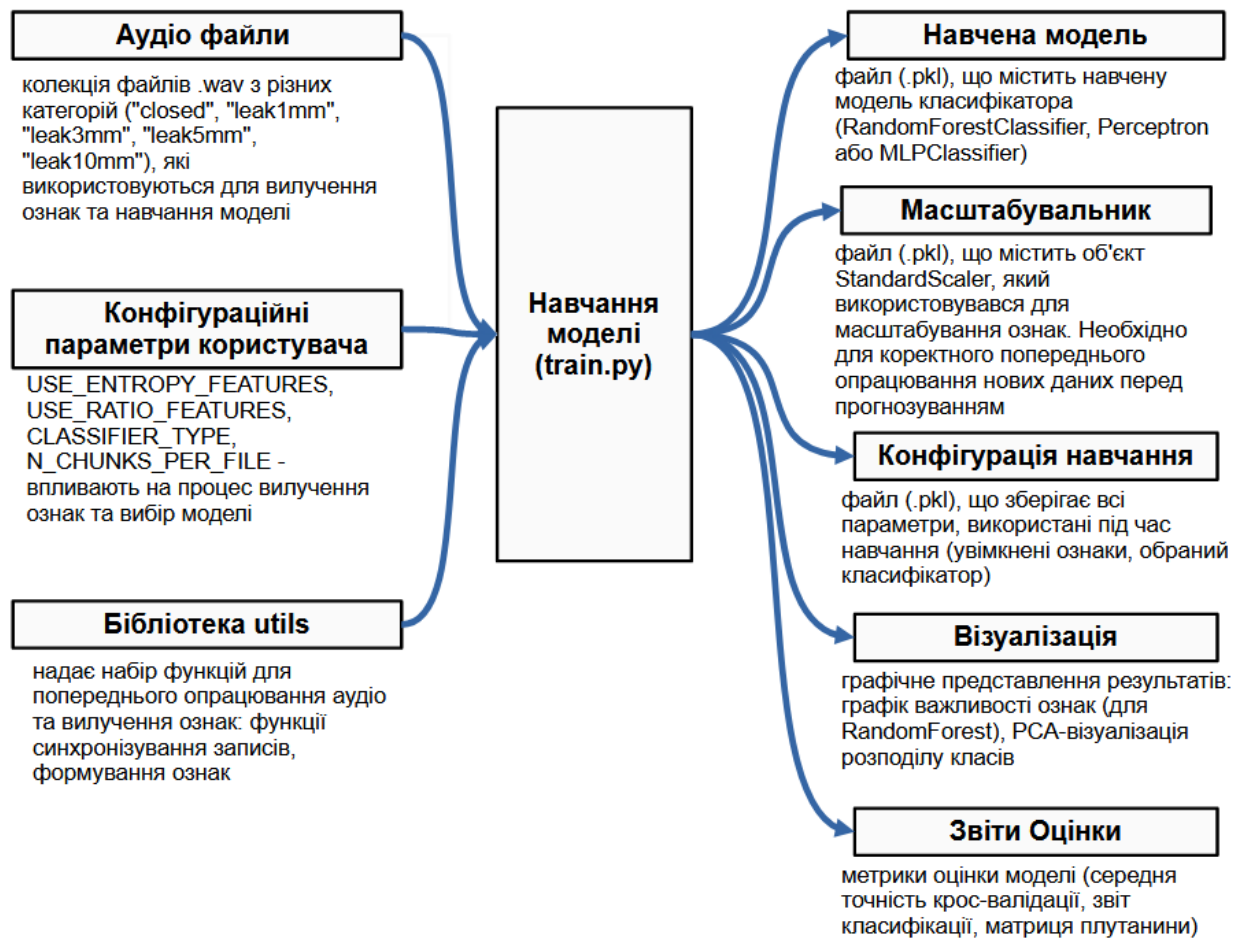


Рисунок 2.6 – Функціональна схема інформаційних потоків програмного модуля навчання моделей

— блок train.py — головний виконуваний скрипт, який оркеструє процес: завантажує дані, викликає функції з utils, формує вектори ознак, проводить навчання обраної моделі та валідацію;

— блок вихідних артефактів та звітності: серіалізовані об'єкти (файли .pkl). Модуль зберігає три файли для подальшого використання: навчена модель — класифікатор RandomForest, MLP або Perceptron з підібраними вагами; масштабувальник — об'єкт StandardScaler, необхідний для нормалізації нових даних у тому ж діапазоні, що й навчальні; конфігурація навчання — метадані про

те, з якими налаштуваннями була створена модель; аналітика та візуалізація — генерація графіків важливостей ознак, PCA-проекцій та текстових звітів з метриками якості (точність, F1-score, матриці плутанини) для оцінки ефективності експерименту.

Вибір інформативних ознак акустичного сигналу для забезпечення його стійкості до експлуатаційної невизначеності (змін робочого тиску та характеристик тестового сигналу), здійснювався у два етапи.

На першому етапі було зібрано набір ознак, які є традиційними для завдань акустичного діагностування [4,12]. Цей початковий вектор включав:

- енергетичні ознаки (енергія для двох каналів — energy_ch1, energy_ch2 та їх співвідношення energy_ratio);
- статистичні характеристики (середньоквадратичне відхилення rms1, rms2 та пікове значення амплітуди peak1, peak2 для кожного каналу);
- спектральні ознаки (середньозважене значення частот spec_centroid1, spec_centroid2 та спектральна ширина смуги spec_bw1, spec_bw2);
- кореляційні ознаки (максимальне значення нормалізованої взаємно-кореляційної функції між каналами norm_corr).

Базовий набір продемонстрував високу чутливість до зміни несучої частоти тестового сигналу, що спричиняло помилки у класифікації (табл. 2.1).

Таблиця 2.1 — Помилки класифікації базового набору ознак, спричинених варіаціями частотних характеристик сигналу (лінійна модель)

Тип помилки	Сценарій плутанини (Істина → Прогноз)	Кількість помилок (шт.)
Хибна тривога (False Positive)	Closed → Leak (5 мм, 10 мм)	10
Пропуск витoku (False Negative)	Leak (3 мм, 5 мм) → Closed	6
Помилка оцінки	Leak X mm → Leak Y mm	9
ВСЬОГО	—	25

Високий рівень помилок класифікації (загалом 25 випадків), зафіксований при використанні лінійної моделі на базовому наборі ознак, пояснюється прямою залежністю енергетичних параметрів від інтерференційної картини акустичного поля у трубопроводі. Показники хибна тривога (False Positive) та пропуск витoku (False Negative) є елементами матриці плутанини і є класичними метриками, на основі яких визначають якість моделі [17]. Аналіз фізичних процесів дозволяє виділити три основні механізми виникнення помилок (за табл. 2.1):

1 ефект зміщення вузлів стоячої хвилі (причина хибних тривог). Найбільшу кількість помилок (10 випадків) склали "хибні тривоги", коли герметичний трубопровід (closed) був ідентифікований як аварійний. Це явище зумовлене тим, що у замкнутому просторі трубопроводу тестовий сигнал утворює систему стоячих хвиль. Зміна несучої частоти тестового сигналу призводить до просторового зміщення вузлів (точок мінімальної амплітуди) та пучностей (точок максимальної амплітуди). Внаслідок цього, у точці розташування датчика може виникнути локальне зростання амплітуди сигналу виключно через інтерференцію, без реального джерела шуму. Базовий набір ознак, що спирається на абсолютні значення енергії (energy, rms), помилково інтерпретує такий інтерференційний сплеск як наявність витoku великого діаметра;

2 спектральна деформація сигналу витoku (причина пропуску цілі). Небезпечні помилки пропуску витоків (6 випадків) пов'язані зі зміною умов поширення звуку при варіаціях робочого тиску. Зміна тиску впливає на густину середовища та швидкість звуку, що, у свою чергу, призводить до зсуву спектральних компонентів шуму витoku. У деяких режимах це викликає ефект енергетичного "провалу" на контрольованих частотах або маскування корисного сигналу гармоніками тестового сигналу. Модель, навчена на фіксованих абсолютних значеннях спектру, перестає розпізнавати "патерн" витoku, сприймаючи його як фоновий шум труби без дефектів;

					KPM.AKCM - 21.00.00.000 ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

3 спотворення міжканального балансу (причина плутанини класів витоків). Помилки в оцінці розміру витоків (9 випадків) зумовлені порушенням стабільності співвідношення енергій між каналами стерео-сигналу. Оскільки затухання акустичної хвилі є частотно-залежним, зміна частоти тестового сигналу нелінійно змінює пропорцію амплітуд, що реєструються лівим та правим давачами. Базова модель, яка не використовує нормовані реляційні ознаки, виявляється нездатною адаптуватися до цих змін, що призводить до некоректної класифікації суміжних класів (наприклад, витік 3 мм ідентифікується як 5 мм).

Наведені фізичні механізми підтверджують, що використання абсолютних енергетичних характеристик є недостатнім для побудови системи детектування витоків в умовах змінних експлуатаційних параметрів, що обґрунтовує необхідність переходу до інваріантних ентропійних та реляційних ознак.

Оцінка важливості ознак для класифікатора Random Forest при такому векторі ознак показана на рис. 2.7.

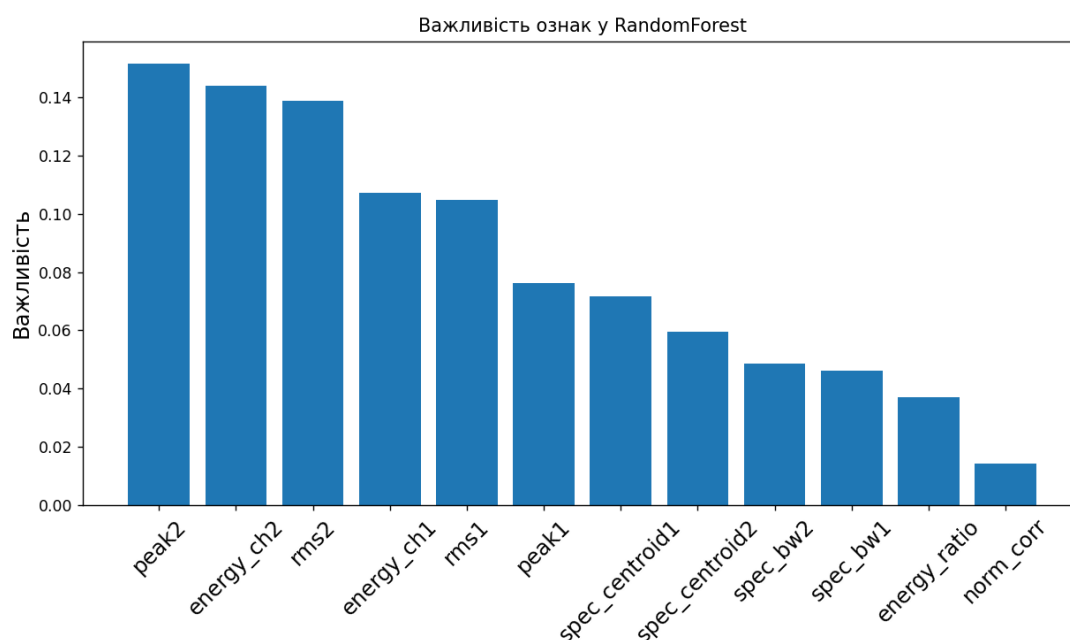


Рисунок 2.7 – Аналіз важливості ознак для базового набору ознак для моделі
Random Forest

На другому етапі для розширення ознак та компенсації наведених вище недоліків вектор ознак був доповнений:

— ентропійними ознаками (entropy1, entropy2), які розраховувалися для кожного каналу шляхом усереднення ентропії Шенона у ковзних вікнах по всьому зразку. Використання ентропійних оцінок є важливим, оскільки зменшення потужності сигналу мало впливає на апертуру оцінок ентропії. Розмір ковзного вікна був обраний як 2048 відліків. Це дозволило аналізувати локальні властивості акустичного сигналу, згенерованого за принципом 5-ти розрядного коду Баркера, в межах одного кодового імпульсу без впливу перехідних процесів між розрядами;

— реляційними ознаками (відношеннями) rms_ratio та entropy_ratio. Вони були додані для зменшення чутливості моделей до змін тиску та для виділення диференційного ефекту витоку.

Після введення додаткових ознак зміни в помилках, які показані в табл. 2.1, наведено в табл. 2.2.

Таблиця 2.2 — Ефективність усунення помилок класифікації шляхом введення реляційних та ентропійних ознак

Тип помилки	Група ознак	Динаміка помилок (шт.)
Хибна тривога (Closed → Leak) Плутанина через високий тиск	Реляційні ознаки (ratios)	10 → 0
Пропуск витоку (Leak → Closed) Маскування витоку резонансом	Ентропійні ознаки (entropy)	6 → 0
Помилка оцінки (Leak X → Leak Y) Плутанина розмірів	Комбінація (entropy + ratios)	9 → 2
ЗАГАЛОМ	Весь вектор	25 → 2

Порівняльний аналіз, наведений у табл. 2.2, демонструє, що розширення простору ознак дозволило вирішити ключові проблеми нестабільності:

— кількість помилок ідентифікації герметичного стану знизилася з 10 до 0. Це підтверджує, що реляційні ознаки (співвідношення сигналів з каналів) виступають ефективним фільтром синфазних завад, роблячи класифікатор інваріантним до змін робочого тиску;

— зменшення кількості пропусків аварійних станів з 6 до 0. Це стало можливим завдяки ентропійним ознакам, які реагують на структурну хаотичність сигналу, властиву турбулентним витокам, і не залежать від абсолютної амплітуди сигналу, яка може варіюватися;

— залишкові помилки класифікації (між класами 3 мм та 5 мм) носять локальний характер і не є критичними для безпеки експлуатації, оскільки факт наявності витоку вірно зафіксований в обох випадках.

Аналіз важливості ознак для комбінованого вектора моделі Random Forest показав, що введення цих додаткових ознак є обґрунтованим та вагомим для навчання моделей (рис. 2.8), оскільки для моделі Random Forest, наприклад, ентропія тепер посідає перше місце за важливістю.

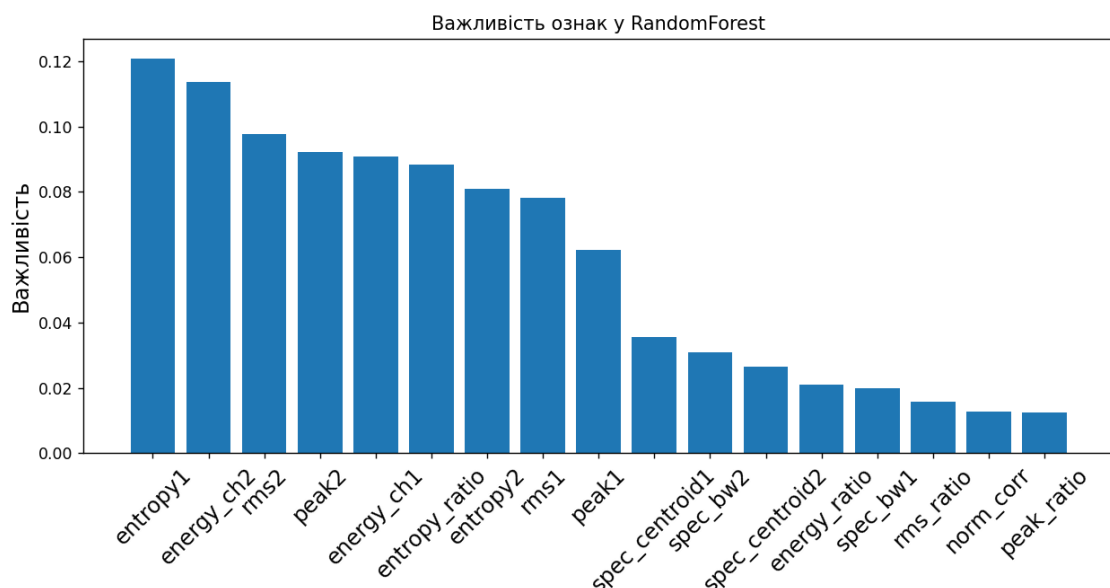


Рисунок 2.8 – Аналіз важливості ознак для комбінованого вектора моделі Random Forest

Отже, завдяки комбінації базових, ентропійних та реляційних ознак, вдалося побудувати стійку модель, здатну ефективно опрацьовувати діагностичні дані, отримані в умовах різних експлуатаційних параметрів.

2.3 Обґрунтування вибору структури класифікатора для класифікації станів трубопроводу

Вибір ансамблевого класифікатора Random Forest (RF) (Випадковий ліс) для класифікації станів трубопроводу обґрунтований необхідністю створення моделі, яка демонструє високу точність та стійкість в умовах експлуатаційної невизначеності. Традиційні методи контролю, зокрема акустичний метод [11], суттєво залежать від змін технологічних параметрів транспортування (таких як тиск і характеристики тестового сигналу). Для подолання цієї залежності було обрано підхід на основі машинного навчання (ML), що вимагає обґрунтованого вибору архітектури класифікатора.

Порівняльний аналіз, проведений в рамках дослідження з використанням модуля train.py (додаток А, рис. 2.6), підтвердив, що задача класифікації акустичних сигналів витоків у запропонованому просторі ознак має нелінійний характер. Це виявилось при порівнянні ефективності трьох архітектур: лінійної (Перцептрон), нелінійної (багатошаровий перцептрон, MLP) та ансамблевої (Random Forest).

Лінійний класифікатор (Перцептрон), який намагається розділити класи гіперплощиною, продемонстрував систематичну нездатність (точність < 90%) розрізнити стани з подібними енергетичними характеристиками, але різною фізичною природою (наприклад, малий витік) (рис. 2.9—2.10). Візуалізація розподілу класів у просторі головних компонент (рис. 2.9) демонструє наявність складних структурних перекриттів між кластерами. Особливо критичним є часткове накладання області, що відповідає стану «норма» (closed), з областями станів з витокami (leak3mm, leak5mm). Така геометрична конфігурація свідчить

					КРМ.АКСм - 21.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

про те, що у базовому енергетичному просторі ознак класи не є лінійно розділюваними — неможливо провести пряму лінію (або гіперплощину), яка б чітко відокремила сигнали витоку від шумів герметичного трубопроводу без суттєвих помилок.

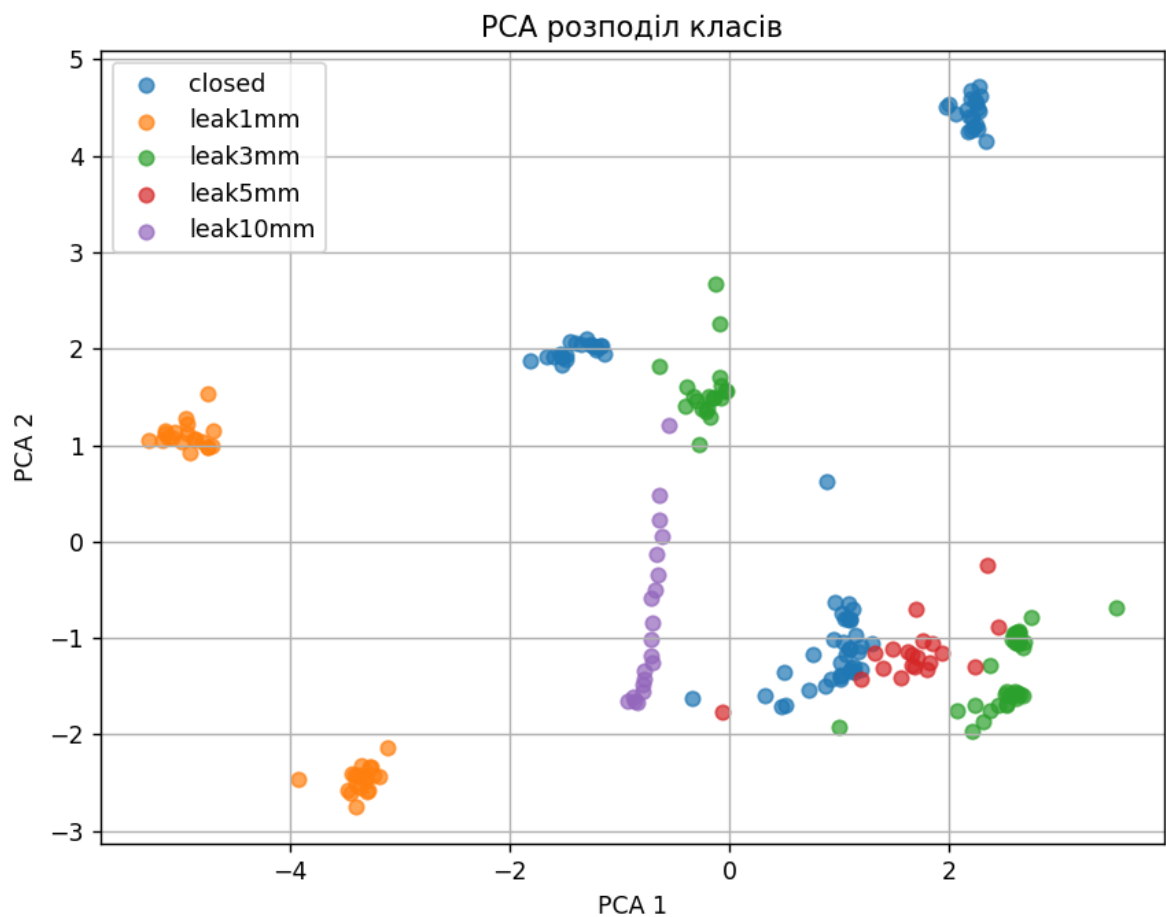


Рисунок 2.9 – Візуалізація аналізу головних компонент (PCA) розподілу класів у просторі базових ознак (без ентропії та відношень)

Кількісне підтвердження нелінійності задачі можна побачити з матриці плутанини лінійного перцептрона (рис. 2.10). Зафіксовано 5 випадків, коли витік діаметром 3 мм (leak3mm) був помилково класифікований як безпечний стан closed. Це помилки пропуску (False Negatives), які критичними.

Спостерігається значна дисперсія в розпізнаванні стану закрито. 9 зразків класу closed були помилково віднесені до категорій великих витоків (leak5mm —

4 помилки, leak10mm — 5 помилок). Це хибні спрацювання (False Positives). Лінійний класифікатор, реагуючи на зростання амплітуди через резонансні явища, не зміг адаптувати границю рішень, яка б враховувала нелінійну природу акустичного відгуку.

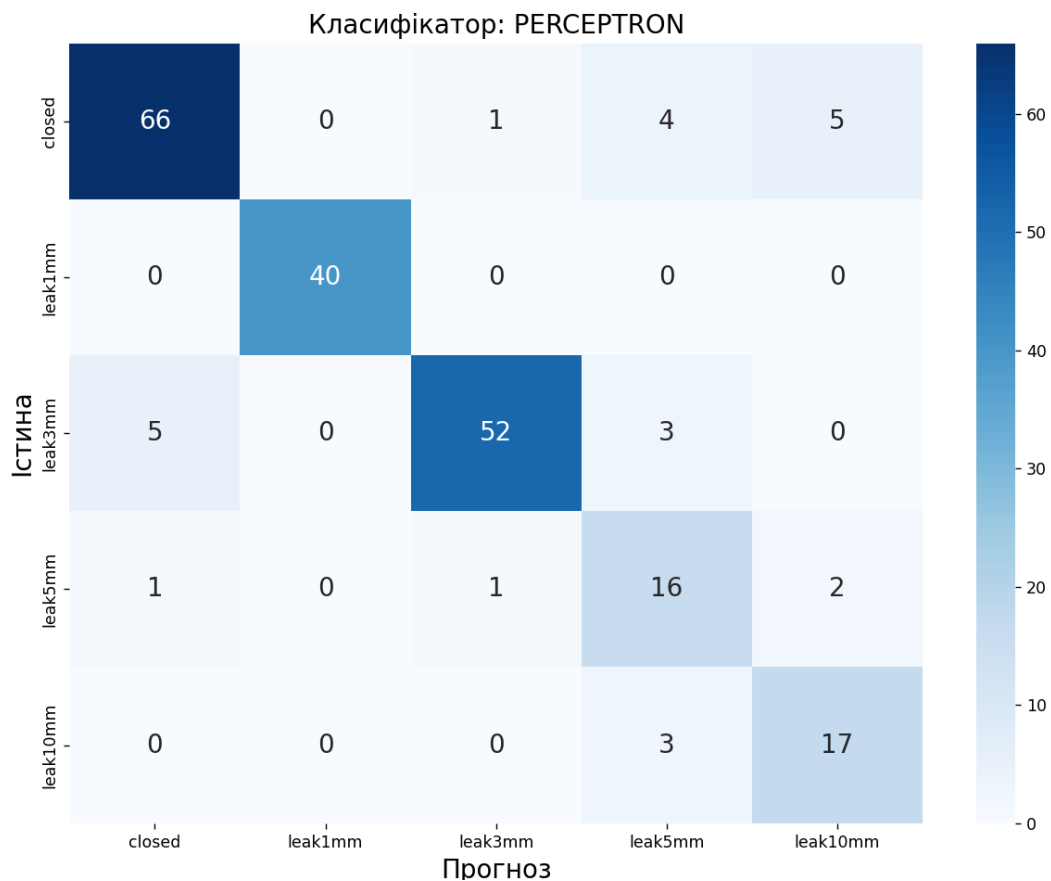


Рисунок 2.10 – Матриця плутанини для перцептрона (без ентропії та відношень)

Модель також допустила помилки при розрізненні розмірів витоків (рис. 2.10) (наприклад, 3 помилки, де leak10mm визначено як leak5mm), що вказує на неможливість лінійного ранжування діаметрів отворів на основі лише базових ознак.

Додавання ентропійних та реляційних ознак дозволило покращити якість класифікації моделі перцептрона, досягнувши середньої точності 0.9216 ± 0.0303 . Проте рівень помилок (табл. 2.3) залишився достатньо високим (17 шт.).

Таблиця 2.3 — Кількісні показники покращення лінійної моделі шляхом введення реляційних та ентропійних ознак

Показник	Базовий набір ознак	Повний набір (Entropy + Ratios)
Точність на класі closed	66/76 (87%)	76/76 (100%)
Хибні тривоги (False Positives)	10 помилок (Closed → Leak)	0 помилок
Пропуски витоків (False Negatives)	6 помилок (Leak → Closed)	1 помилка
Загальна кількість помилок	25	17

Матриця плутанини для комбінованого вектору ознак показана на рис. 2.11.

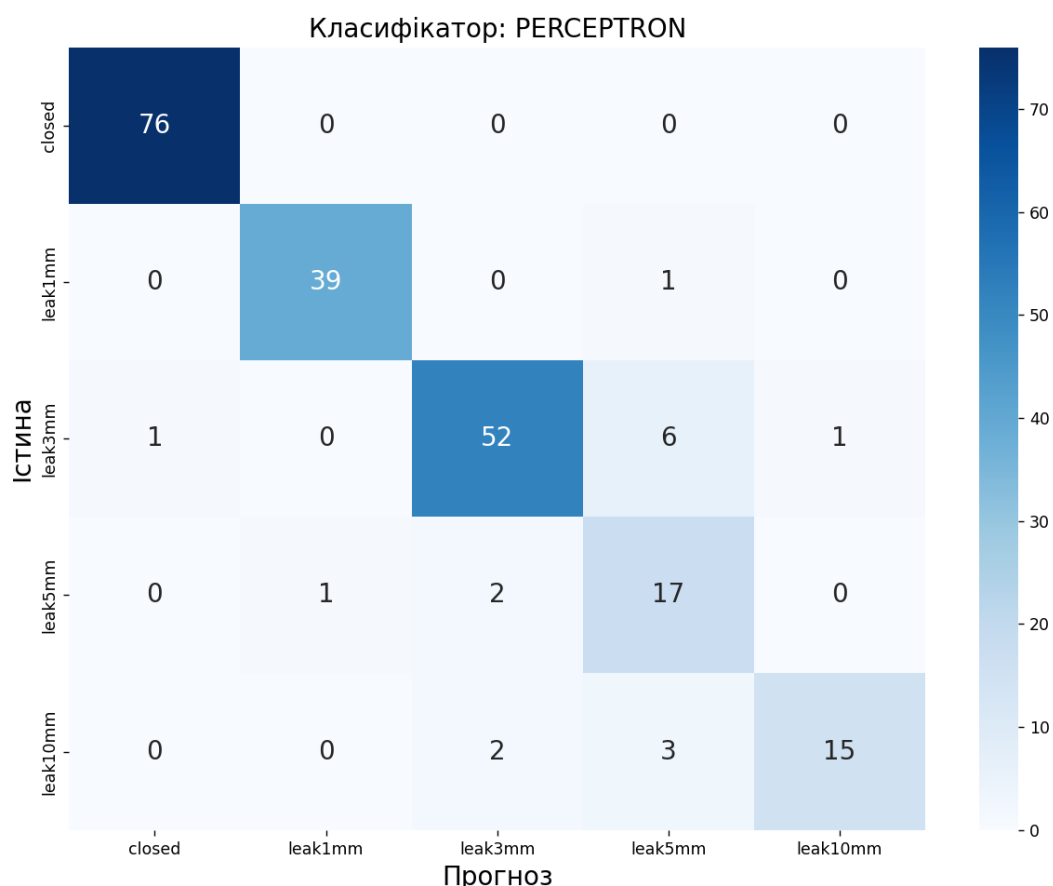


Рисунок 2.11 – Матриця плутанини для перцептрона (комбінований вектор ознак)

Введення ентропійних та реляційних ознак дозволило лінійній моделі вирішити бінарну задачу "закрито / витік" (зменшивши критичні помилки I та II роду майже до нуля). Однак, задача класифікації розміру витіку (багатокласова задача) залишилася нерозв'язаною для лінійної архітектури. Отже, сукупність візуальних даних РСА та структури помилок у матриці плутанини (загалом 25 помилок) за рис. 2.9—2.10 та 17 помилок після формування комбінованого вектору ознак (рис. 2.11) доводить, що границя між класами «закрито» та «витік» має складну, нелінійну форму. Це робить застосування простих лінійних класифікаторів (типу Perceptron) неефективним.

Перехід до нелінійних архітектур забезпечив покращення точності. Багатошаровий перцептрон (MLP), завдяки апроксимаційним властивостям функцій активації ReLU, та ансамблевий метод Random Forest, що формує складні кусково-лінійні границі рішень, змогли коректно ідентифікувати кластери класів, які мають складну геометричну структуру та взаємні перекриття. Це підтверджує гіпотезу про суттєву нелінійність залежностей між діагностичними ознаками (ентропія, співвідношення енергій) та технічним станом трубопроводу. При цьому якість моделі (середня точність) MLP при використанні комбінованого вектору ознак становить 0.9862 ± 0.0113 , а для Random Forest — 0.9908 ± 0.0113 . Приклад матриці плутанини для моделі MLP показано на рис. 2.12, а для Random Forest – на рис. 2.13.

Ключовим критерієм для вибору моделі, окрім точності, є її здатність до узагальнення (генералізації), особливо при тестуванні на даних з невідомими моделям варіаціями тиску та характеристиками тестового сигналу. Обидві нелінійні архітектури (MLP та Random Forest) досягли майже ідеальних результатів на навчальній вибірці (F1-score 98,6%–99,1%). Аналіз тестування на незалежній вибірці, яка містить дані, отримані за інших експлуатаційних умов, буде проведено в розділі 3.

					КРМ.АКСм - 21.00.00.000 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

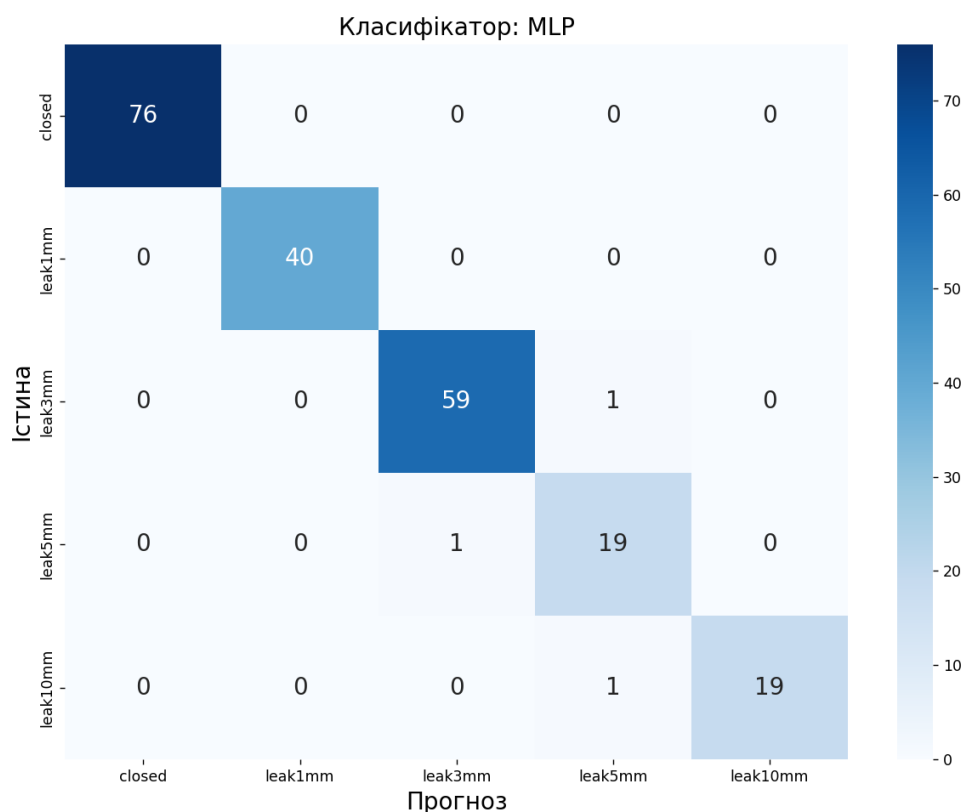


Рисунок 2.12 – Матриця плутанини для MLP (комбінований вектор ознак)

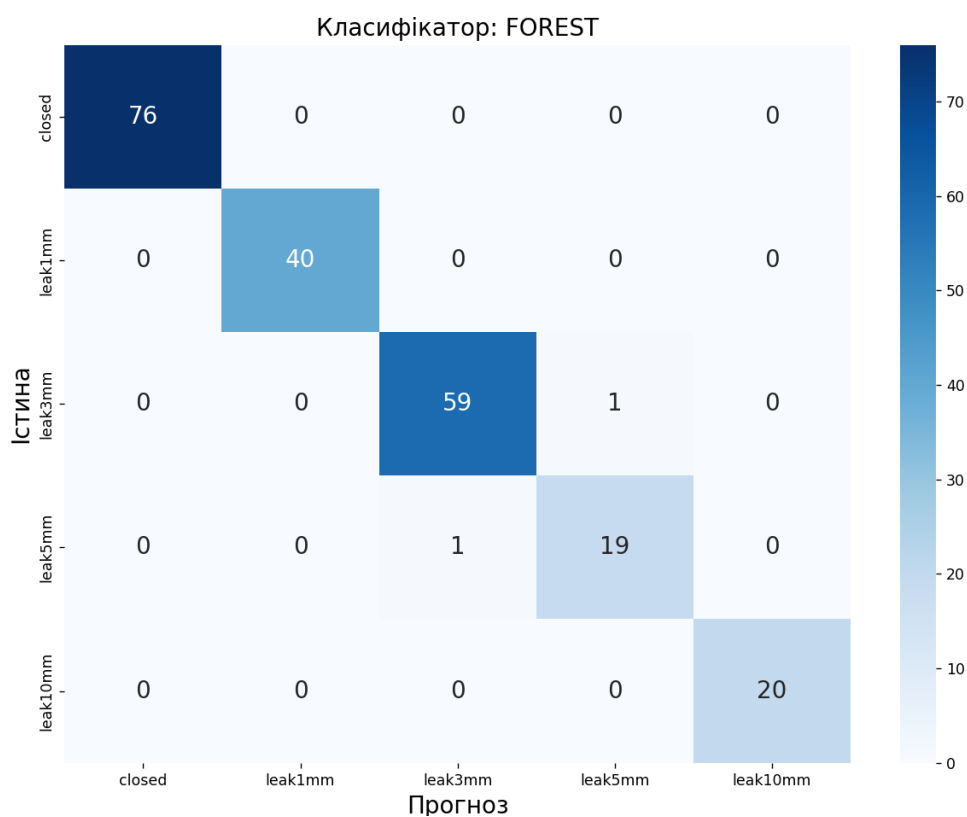


Рисунок 2.13 – Матриця плутанини для Random Forest (комбінований вектор ознак)

Таким чином, було обрано класифікатор нелінійної архітектури, оскільки він поєднує високу точність, необхідну для нелінійної задачі, з винятковою стійкістю та здатністю до узагальнення на гетерогенних даних, що є критичним для надійного функціонування системи моніторингу в реальних експлуатаційних умовах. В подальшому при тестуванні на відкладеній вибірці буде показано перевагу в бік використання Random Forest.

2.4 Алгоритм детектування та локалізації витоків на основі різницевої імпульсної характеристики

Розроблена інтелектуальна система використовує переваги активного акустичного методу контролю [11] (тестового методу діагностування), який ґрунтується на генеруванні тестових коливань у середовищі заповнення трубопроводу, та потужностей хмарних обчислень AWS [10], що забезпечують більше гнучкість і ефективність порівняно з класичною промисловою автоматизацією.

Етап збору даних та попереднього опрацювання (AWS Lambda) містить наступні кроки:

1 генерування тестового сигналу та збір даних: система генерує завадостійкий тестовий сигнал (наприклад, фазоманіпульований код Баркера) в середовищі транспортування. Акустичні давачі реєструють вторинні відбиті хвилі (сигнали-відгуки), що надходять від неоднорідностей трубопроводу (витоків, запірної арматури). Зареєстровані дані у формі wav-файлів з двох каналів реєстрування зберігаються та накопичуються у хмарному сховищі Amazon S3;

2 первинне опрацювання: при надходженні даних до Amazon S3 автоматично активується сервіс AWS Lambda для попереднього опрацювання та нормалізації сигналів. Це обчислення ВКФ між тестовим та зареєстрованим сигналами;

					KPM.AKCM - 21.00.00.000 ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

З формування імпульсної характеристики та інженерія ознак: за допомогою перетворення Гільберта обчислюється обвідна ВКФ, яка є імпульсною характеристикою трубопроводу. З цієї ІХ та сигналів-відгуків формується вектор ознак, який включає:

- базові ознаки (енергетичні, статистичні, спектральні);
- ентропійні складові (ентропія Шенона), які важливі для ідентифікації слабких відбитих сигналів;
- реляційні ознаки (співвідношення між оцінками каналів), необхідні для компенсації синфазних завад, викликаних зміною робочого тиску.

Етап інтелектуальної класифікації (AWS SageMaker) покриває функції класифікування стану трубопроводу. Сформований вектор ознак передається в Amazon SageMaker, де розгорнуто навчену модель Random Forest. Класифікатор здійснює виявлення витoku шляхом класифікації технічного стану серед п'яти можливих станів ("closed", "leak1mm", "leak3mm", "leak5mm", "leak10mm"). Вибір ансамблевого класифікатора Random Forest зумовлений його вищою здатністю до узагальнення на нових даних (буде показано в розділі 3) і стійкістю до експлуатаційної невизначеності (змін тиску та характеристик тестового сигналу) порівняно з нейромережевими моделями.

Етап локалізації та виведення результатів реалізує функції:

- локалізації витoku: якщо класифікатор Random Forest підтверджує наявність витoku, система переходить до точного визначення координати. Для цього використовується метод різницевої імпульсної характеристики. Шляхом віднімання еталонної ІХ (отриманої за відсутності витoku) від поточної виділяються нові локальні максимуми, які ідентифікують місце витoku. Координата витoku (L) обчислюється за формулою [2]:

$$L = c * \Delta T / 2, \quad (2.1)$$

де ΔT – час затримки приходу відбитої хвилі (визначений за максимумом ВКФ), а c – уточнена швидкість поширення коливань.

					КРМ.АКСм - 21.00.00.000 ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

Результати локалізації (координата витоку або факт його відсутності) та діагнози зберігаються в базі даних (DynamoDB), що забезпечує швидкий доступ. Для візуалізації та моніторингу результати виводяться на інтерактивні інтерфейси через Amazon QuickSight.

					КРМ.АКСм - 21.00.00.000 ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

3 РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ДЕТЕКТУВАННЯ ВИТОКІВ ТА ОЦІНКА ЇЇ ЕФЕКТИВНОСТІ

3.1 Результати порівняльного аналізу ефективності нелінійних класифікаторів на основі комбінованого вектора ознак

Для забезпечення об'єктивності порівняння архітектур та верифікації їхньої здатності до узагальнення, було розроблено спеціалізований програмний модуль валідації (predict.py). Функціональна схема інформаційних потоків цього модуля наведена на рис. 3.1, а код модуля – у додатку Б.



Рисунок 3.1 – Функціональна схема інформаційних потоків модуля прогнозування

Процес валідації реалізовано за принципом повної ізоляції тестових даних від процесу навчання. Як видно з рис. 3.1, модуль приймає на вхід масив нових аудіофайлів (.wav) та набір попередньо згенерованих артефактів:

- навчені моделі (.pkl): серіалізовані об'єкти класифікаторів (RandomForest, MLP, Perceptron), що містять фінальні вагові коефіцієнти;
- масштабувальники (.pkl): об'єкти StandardScaler, які зберігають параметри розподілу (середнє та дисперсію) навчальної вибірки. Це важливо для того, щоб нові дані були нормалізовані ідентично до навчальних;
- конфігурація навчання: файл метаданих, що гарантує використання тотожного набору ознак (наприклад, USE_ENTROPY=True) під час прогнозування;
- бібліотека utils: забезпечує ідентичність алгоритмів вилучення ознак на етапах навчання та прогнозування.

Результатом роботи модуля є аналітичні звіти, що включають матриці плутанини та метрики точності.

Дані, наведені в п. 2.3 підтвердили, що перехід до комбінованого вектора ознак (спектральна ентропія + реляційні характеристики) дозволяє досягти високої точності детектування стану трубопроводу лише за умови використання нелінійних методів класифікації. Ефективність ансамблевого методу (Random Forest) згідно з матрицею плутанини, отриманою в ході крос-валідації (рис. 2.13), досягла середньої точності 99.08%. Аналіз помилок показує високу стабільність моделі. Клас closed (закрито) ідентифіковано безпомилково (76 з 76 зразків вірно), що свідчить про повну компенсацію впливу змінного тиску завдяки реляційним ознакам. Зафіксовано повну відсутність пропусків аварійних станів (0 помилок типу False Negative). Єдині зафіксовані помилки (2 випадки) стосуються плутанини між суміжними класами витоків (leak3mm і leak5mm), що не є критичним для факту виявлення витoku.

Багатошаровий перцептрон (MLP) також продемонстрував високу здатність до навчання на комбінованому векторі ознак, досягнувши точності

					KPM.AKCM - 21.00.00.000 ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

98.62% на етапі крос-валідації. Однак, поглиблений аналіз показав його меншу стійкість порівняно з Random Forest (рис. 2.12). Хоча MLP також безпомилково класифікував стан спокою, він допустив на одну помилку більше при розрізненні класів витоків.

Суттєвим недоліком для MLP виявився значний "розрив узагальнення" (понад 16%) при переході до незалежних тестових даних (табл. 3.1), що вказує на схильність нейромережі до перенавчання на обмежених вибірках акустичних даних (рис. 3.2).

Таблиця 3.1 — Порівняльний аналіз здатності моделей до узагальнення

Модель	Етап 1: навчання(крос-валідація)	Етап 2: незалежна вибірка	Розрив узагальнення
Random Forest (ансамбль)	99.08%	94.31%	-4.77% (низький)
MLP (нейромережа)	98.62%	82.06%	-16.56% (критичний)

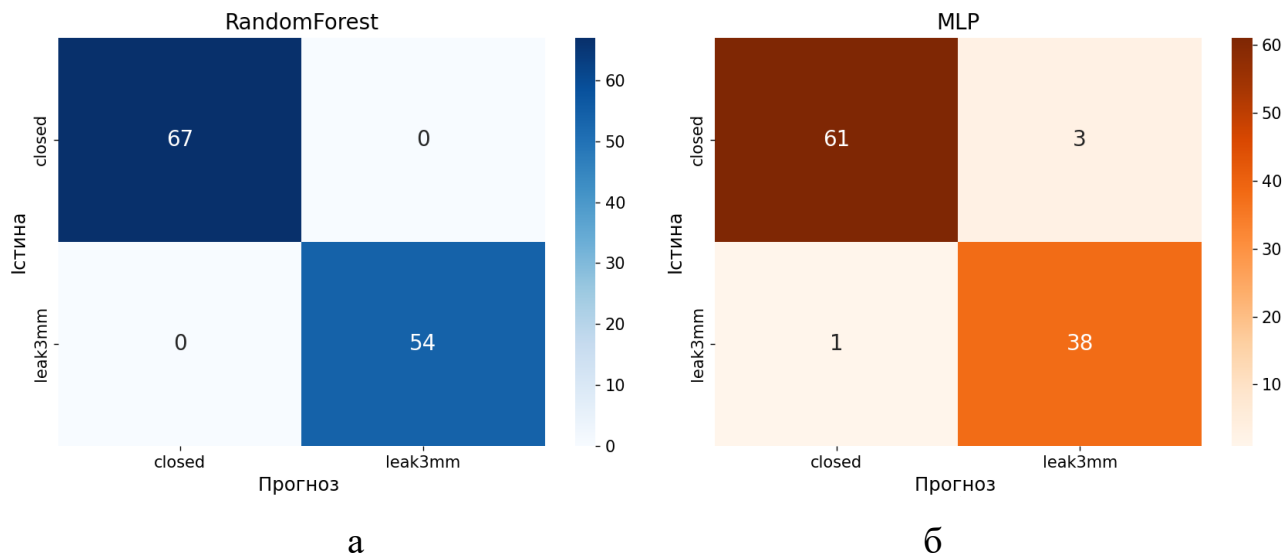


Рисунок 3.2 – Порівняння результатів застосування моделей Random Forest (а) та MLP (б) на тестовій вибірці

Для проведення фінального порівняльного тесту (результати якого наведено на рис. 3.2) було відібрано масив аудіофайлів, записаних в умовах, що імітують реальні експлуатаційні виклики:

— зміна тиску: записи містили фоновий шум герметичного трубопроводу при змінних режимах роботи насосного агрегату, що змінювало енергетичний профіль сигналу;

— було обрано пару класів «closed» (закрито) та «leak3mm» (малий витік). Це складний сценарій бінарної класифікації, оскільки акустична енергія малого витоку часто співмірна з рівнем шумів потоку, що створює високий ризик помилок.

За даними рис. 3.2, а результати Random Forest (ансамблевий метод) матриця плутанини демонструє ідеальну роботу ансамблевого класифікатора: точність 100%, модель безпомилково класифікувала всі 67 зразків стану closed та 54 зразки стану leak3mm. Також відсутні помилки типу False Positive (хибна тривога), що свідчить про те, що модель успішно використала реляційні ознаки (ratios) для ігнорування змін тиску. Відсутність помилок False Negative (пропуск витоку) підтверджує, що ентропійні ознаки дозволили чітко виділити слабкий сигнал витоку 3 мм на фоні шумів.

Результати MLP (рис. 3.2, б) ілюструють нестабільність нейромережі на нових даних, незважаючи на її високі показники при навчанні. Загалом допущено 4 помилки: хибні тривоги (3 випадки) — три зразки герметичної труби були помилково визначені як витік. Це підтверджує схильність MLP до перенавчання на амплітудних характеристиках, збільшення тиску в тестовому записі було сприйнятий мережею як витік. Пропуск витоку (1 випадок) — один зразок витоку 3 мм був класифікований як закрито. Це небезпечний тип помилки для систем безпеки.

Отже, рис. 3.2 демонструє перевагу ансамблевого підходу. При порівняно однакових результатах при проведенні 5ти кратної стратифікованої перехресної валідації для обох нелінійних моделей, модель MLP є нестійкою при зміні умов

					KPM.AKCM - 21.00.00.000 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

експерименту (зсув домену даних) для тестової вибірки, тоді як Random Forest формує жорсткі, інтерпретовані правила на основі ентропії, що забезпечує абсолютну надійність навіть у граничних режимах детектування (малі витoki). Отже, частиною розроблюваної структури інтелектуальної системи буде саме класифікатор Random Forest.

Результати валідації за допомогою розробленого модуля predict.py доводять, що інтеграція ентропійних та реляційних ознак у поєднанні з нелінійним ансамблевим класифікатором Random Forest забезпечує побудову найбільш сійкої системи детектування втоків. Ця конфігурація мінімізує ризики помилкових тривог та пропуску витоків, демонструючи перевагу над нейромережевими та лінійними аналогами.

3.2 Архітектура та конфігурація хмарного середовища AWS

Для забезпечення масштабованості, надійності та можливості опрацювання акустичних даних у реальному часі, розроблений алгоритм детектування витоків інтегрується у хмарну екосистему Amazon Web Services (AWS). Загальна структурна схема запропонованої системи зображена на рис. 3.3 [10].

Запропонована архітектура (рис. 3.3) реалізує конвеєр (pipeline) опрацювання даних і складається з наступних функціональних рівнів:

— рівень збору та зберігання даних: вхідною точкою системи є сервіс Amazon S3, який виступає в ролі масштабованого "озера даних". Сюди надходять акустичні сигнали (формат .wav) з сенсорів, встановлених на трубопроводах. Це забезпечує надійне зберігання історичних даних для подальшого донавчання моделей;

— рівень попереднього опрацювання: для автоматизації операцій використовується безсерверна технологія AWS Lambda. Функції Lambda налаштовані як тригери: вони автоматично активуються при появі нових файлів

					KPM.AKCM - 21.00.00.000 ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

в S3, виконують нормалізацію сигналів та розрахунок інваріантних ознак (ентропії, співвідношень енергій), передаючи підготовлені вектори даних на наступний етап;

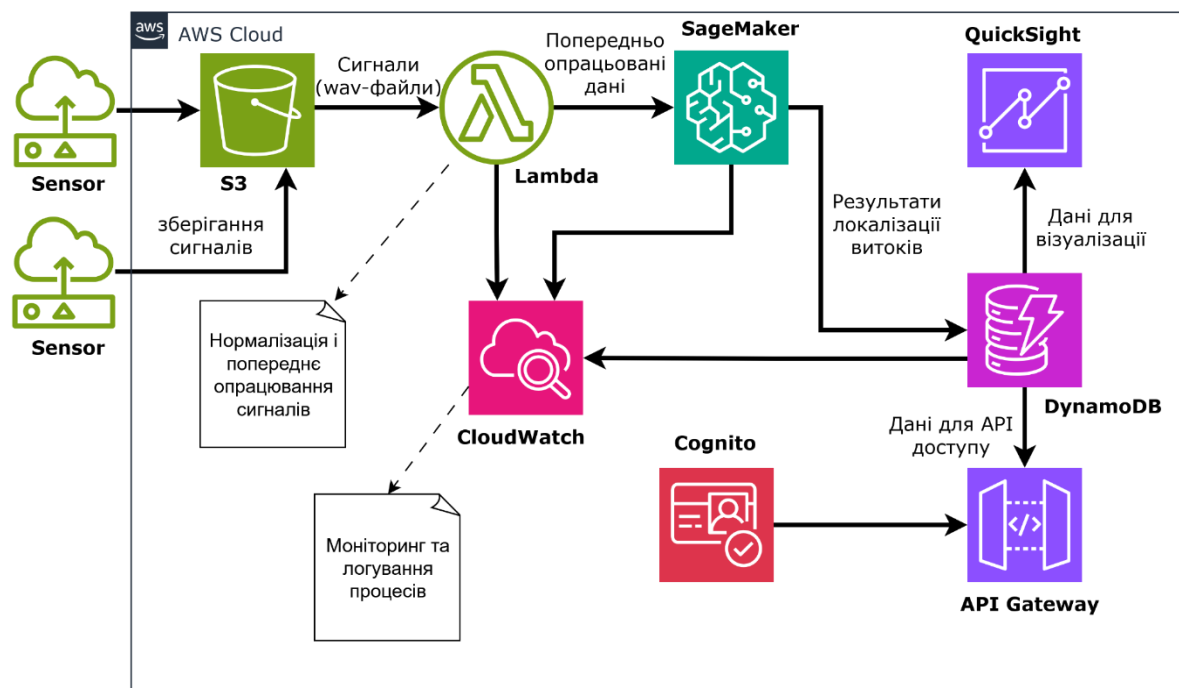


Рисунок 3.3 – Структура інтелектуальної системи детектування витоків на основі хмарних сервісів AWS

— рівень інтелектуального аналізу: Amazon SageMaker забезпечує життєвий цикл моделей машинного навчання — тренування розробленого класифікатора (Random Forest), його валідація та розгортання у вигляді кінцевої точки (endpoint) для отримання прогнозів;

— рівень збереження результатів та візуалізації: результати детектування (мітка класу, ймовірність витоку, координати) зберігаються у базі даних Amazon DynamoDB. Для оперативного моніторингу стану мережі оператором використовується сервіс бізнес-аналітики Amazon QuickSight, який візуалізує статистику інцидентів;

— рівень інтеграції та безпеки: зовнішній доступ до системи для сторонніх додатків реалізується через API Gateway, а контроль доступу та

авторизація користувачів забезпечуються сервісом Amazon Cognito. Моніторинг працездатності всіх компонентів здійснюється через CloudWatch.

Нижче детально розглянуто процес програмної реалізації ключових компонентів системи.

1. Організація сховища даних (Data Lake) на базі Amazon S3. Першим етапом розгортання системи є створення структурованого сховища даних ("Озера даних" — Data Lake), яке забезпечує централізований доступ до навчальних матеріалів та артефактів моделі. Реалізація здійснюється за допомогою сервісу Amazon S3 (Simple Storage Service). Архітектура сховища передбачає поділ на три логічні зони (бакети або префікси):

— Raw Data Zone (/raw-audio): призначена для зберігання вихідних аудіофайлів (.wav), отриманих з акустичних датчиків. Файли класифікуються за каталогами відповідно до міток класів (closed, leak1mm, leak3mm тощо);

— Processed Data Zone (/features): Зберігає результати етапу інженерії ознак — структуровані табличні дані (CSV), що містять розраховані вектори ознак (спектральна ентропія, RMS, ratios) для кожного запису. Це дозволяє уникнути повторного перерахунку ознак при кожному запуску навчання;

— Model Artifacts Zone (/models): призначена для автоматичного зберігання серіалізованих файлів навчених моделей (model.tar.gz), які генерує SageMaker після завершення тренування.

Оскільки розгортання проєкту в хмарі виконано з використанням оточення sandbox курсу AWS Academy Machine Learning Foundations, то інфраструктура збереження об'єктів в AWS S3 була модифікована від обмеження прав користувача test_student, для якого дозволено обмежений перелік дій. Зокрема, сховище структуровано за логічними зонами в межах бакету sagemaker-us-east-1-905418030392 (рис. 3.4). Ключовим каталогом проєкту визначено leak-detection-project/, де зберігаються наступні об'єкти:

— артефакти моделі (model.tar.gz): серіалізовані об'єкти класифікатора Random Forest та засобу масштабування даних StandardScaler, що

					KPM.AКСм - 21.00.00.000 ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

завантажуються за допомогою SDK `sagemaker_session.upload_data` для гарантування прав доступу на рівні IAM-ролей;

— вихідний код інференсу (`sourcedir.tar.gz`): архів, що автоматично формується платформою та містить логіку обробки запитів, включаючи файли `inference.py` та `requirements.txt`.

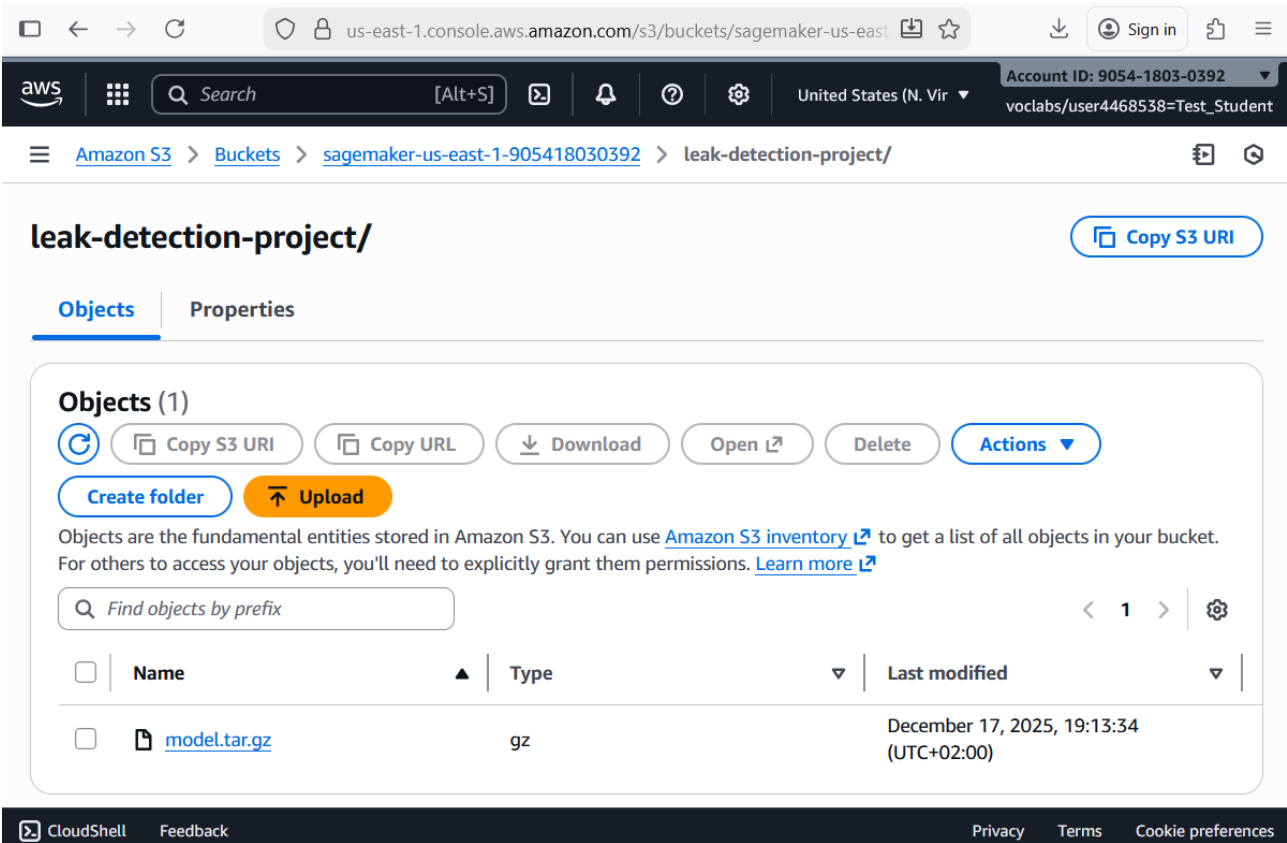


Рисунок 3.4 — Організація сховища даних на базі Amazon S3

2. Вибір обчислювальних ресурсів та середовища розробки. Для розробки та валідації системи використано кероване середовище Amazon SageMaker Notebook Instance, що базується на Docker-контейнері з попередньо налаштованим ядром `conda_tensorflow2_p310`. Конфігурація середовища передбачає використання:

— інстансів розробки — використано сімейство `ml.t2.medium`, що забезпечує баланс між продуктивністю та економічною ефективністю при роботі з табличними даними;

— IAM Execution Role — налаштування політики доступу, що дозволяє інстансу SageMaker виконувати операції читання/запису в S3, взаємодіяти з CloudWatch для логування та розгортати мережеві точки доступу (endpoints) (рис. 3.5).

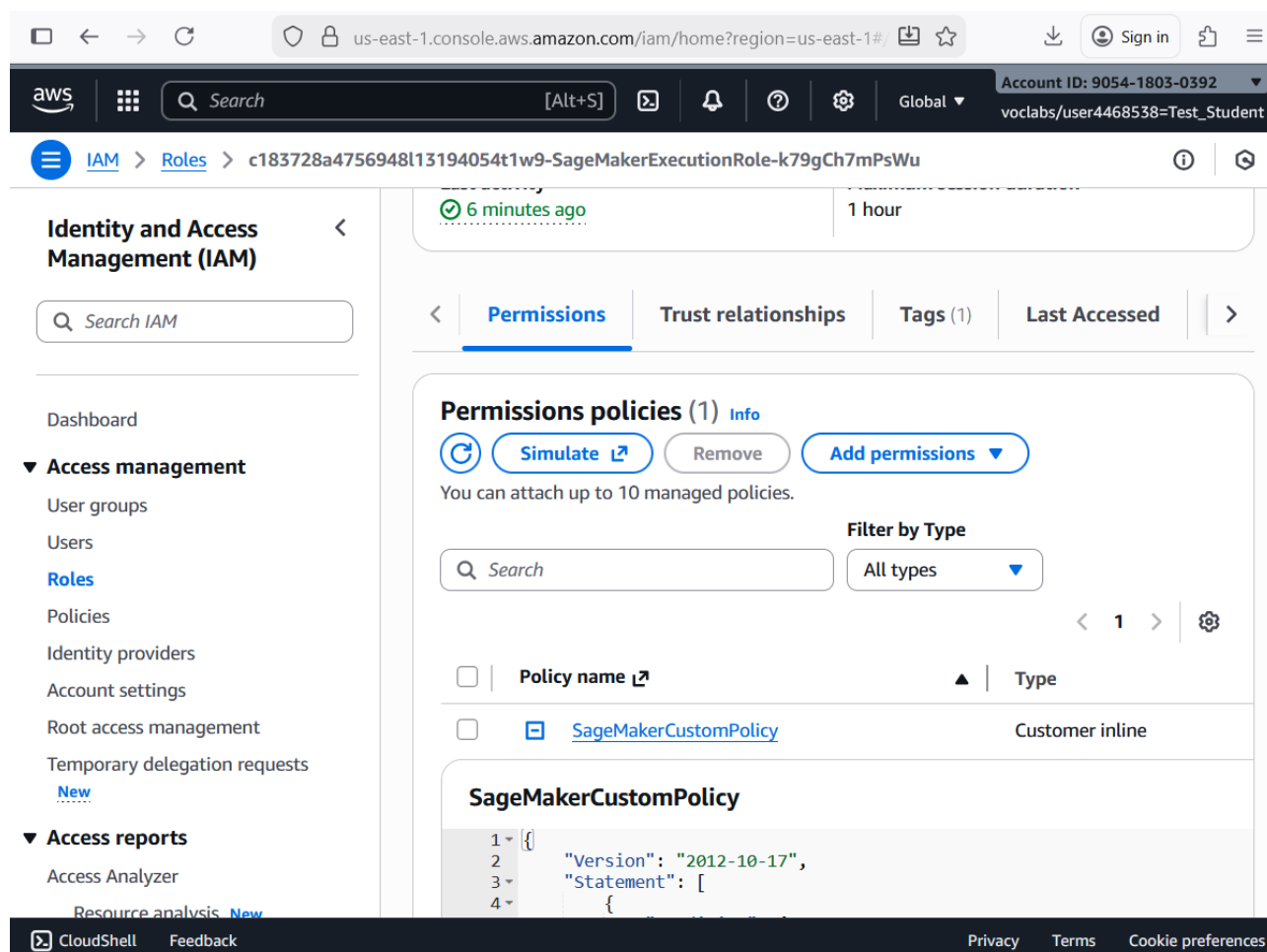


Рисунок 3.5 – Налаштування політики доступу для інстансу SageMaker

Правила політики, показаної на рис. 3.5 наведено нижче у форматі json:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "sagemaker.amazonaws.com"
        }
      },
      "Action": [
        "iam:PassRole",
        "iam:GetRole"
      ]
    }
  ]
}
```

					KPM.AKCM - 21.00.00.000 ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

```

"Resource": [
  "arn:aws:iam::*:role/*SageMakerExecutionRole*",
  "arn:aws:iam::*:role/ForecastRole"
],
"Effect": "Allow"
},
{
  "Action": [
    "sagemaker:CreateModel"
  ],
  "Resource": "arn:aws:sagemaker:us-east-1:905418030392:model/*",
  "Effect": "Allow"
},
{
  "Action": [
    "sagemaker:CreateEndpointConfig",
    "sagemaker:CreateEndpoint",
    "sagemaker:DescribeEndpoint",
    "sagemaker:InvokeEndpoint",
    "sagemaker>DeleteEndpointConfig",
    "sagemaker>DeleteEndpoint"
  ],
  "Resource": "arn:aws:sagemaker:us-east-1:905418030392:endpoint/*",
  "Effect": "Allow"
},
{
  "Action": [
    "sagemaker>CreateTrainingJob",
    "sagemaker:DescribeTrainingJob",
    "sagemaker:StopTrainingJob",
    "sagemaker:ListTags"
  ],
  "Resource": "arn:aws:sagemaker:us-east-1:905418030392:training-job/*",
  "Effect": "Allow"
},
{
  "Action": [
    "sagemaker>CreateHyperParameterTuningJob",
    "sagemaker:ListTrainingJobsForHyperParameterTuningJob"
  ],
  "Resource": "arn:aws:sagemaker:us-east-1:905418030392:hyper-parameter-tuning-job/*",
  "Effect": "Allow"
},
{
  "Action": [
    "sagemaker>CreateTransformJob"
  ],
  "Resource": "arn:aws:sagemaker:us-east-1:905418030392:transform-job/*",
  "Effect": "Allow"
},
{
  "Action": [
    "logs:DescribeLogStreams"
  ],
  "Resource": "arn:aws:logs:us-east-1:905418030392:log-group/*",
  "Effect": "Allow"
},
{
  "Action": [
    "rekognition:CreateCollection",
    "rekognition>DeleteCollection",
    "rekognition:IndexFaces",

```

					KPM.AKCM - 21.00.00.000 ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		


```

        "rekognition:ListFaces",
        "rekognition:SearchFacesByImage"
    ],
    "Resource": "arn:aws:rekognition:us-east-1:905418030392:collection/Collection",
    "Effect": "Allow"
},
{
    "Action": [
        "sagemaker:Describe*"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Action": [
        "cloudwatch:GetMetricStatistics"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Action": [
        "s3:CreateBucket",
        "s3:PutObject",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:DeleteObject"
    ],
    "Resource": [
        "arn:aws:s3:::sagemaker*",
        "arn:aws:s3:::sagemaker/*"
    ],
    "Effect": "Allow"
},
{
    "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:ListBucket"
    ],
    "Resource": [
        "arn:aws:s3:::aws-tc-largeobjects",
        "arn:aws:s3:::aws-tc-largeobjects/CUR-TF-200-ACMLFO-1/*",
        "arn:aws:s3:::aws-tc-largeobjects/CUR-TF-200-ACAIML-1/*",
        "arn:aws:s3:::c183728a4756948113194054t1w905418030-sandboxbucket-etew6oowksv3",
        "arn:aws:s3:::c183728a4756948113194054t1w905418030-sandboxbucket-etew6oowksv3/*"
    ],
    "Effect": "Allow"
}
]
}

```

3. Конфігурація контейнеризованого середовища інференсу. Для реалізації онлайн-моніторингу (Online Inference) обрано архітектуру на основі SageMaker SKLearnModel. Конфігурація передбачає використання спеціалізованого Docker-образу з підтримкою Scikit-learn версії 1.2-1. На відміну від локального середовища, хмарна конфігурація вимагає чіткої відповідності

					KPM.AКСм - 21.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

версій Python (py3) та бібліотек для уникнення помилок десеріалізації об'єктів joblib.

4. Система моніторингу та логування CloudWatch забезпечує:

- відстеження метрик продуктивності ендпоінтів (Invocations, ModelLatency);
- логування потоків виводу Python-скриптів, що є критичним для налагодження логіки inference.py та діагностики помилок типу Internal Server Error (500).

Приклад моніторингу активності розгорнутого ендпоінта сервісом CloudWatch наведено на рис. 3.6.

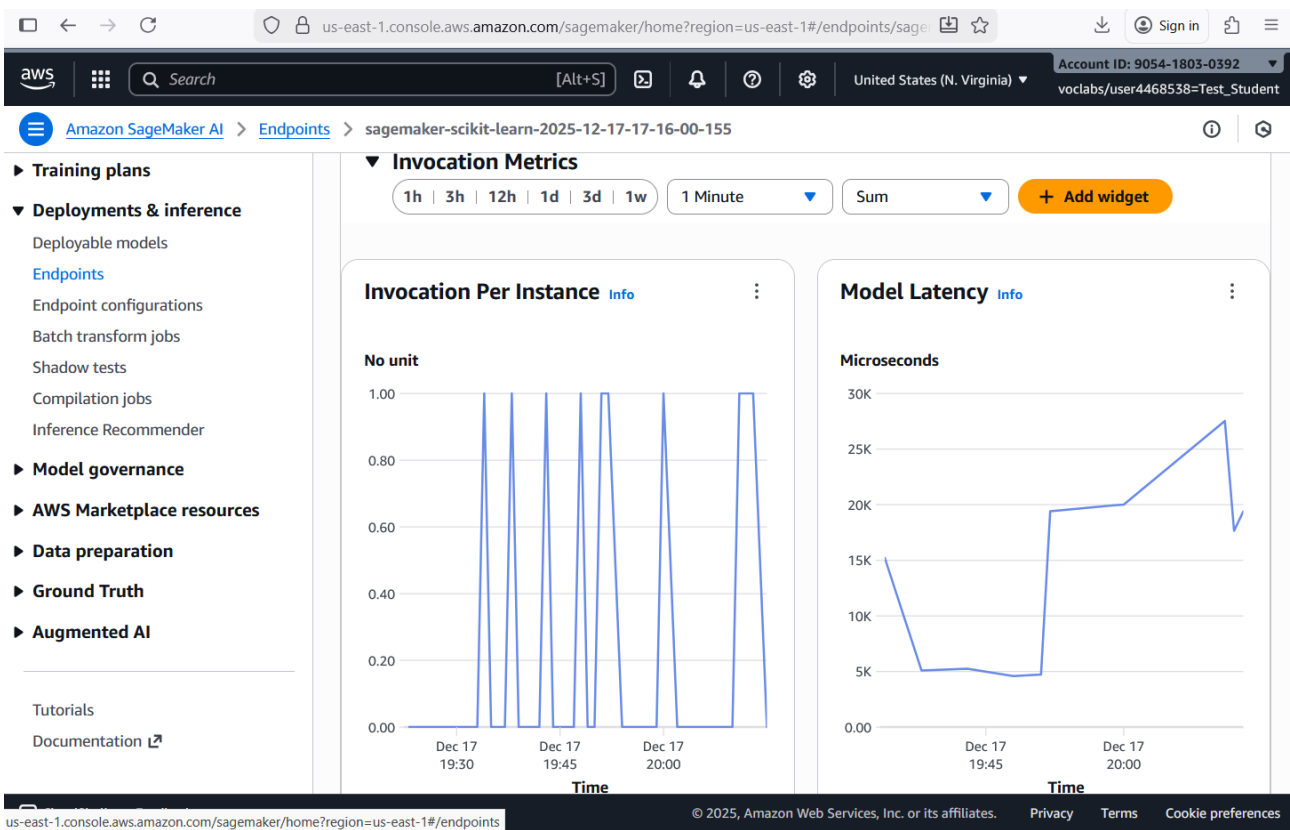


Рисунок 3.6 – Моніторингу активності розгорнутого ендпоінта сервісом CloudWatch

Розроблена архітектура дозволяє реалізувати гнучкий процес деплою, де обчислювальні ресурси для навчання та хостингу моделі відокремлені, що

забезпечує стабільну роботу системи при інтенсивному потоці вхідних даних з давачів у вигляді записів формату wav.

5. Валідація та тестування хмарного рішення. Після розгортання проводиться перевірка працездатності системи безпосередньо з Jupyter Notebook за допомогою клієнта boto3. На вхід подаються контрольні вектори ознак з тестової вибірки, і отримані прогнози порівнюються з еталонними мітками для підтвердження того, що хмарна реалізація моделі працює ідентично до локального прототипу (описано в п. 3.4).

3.3 Програмна реалізація підсистеми машинного навчання в середовищі Amazon SageMaker

Програмна реалізація інтелектуальної підсистеми розпочинається з розгортання спеціалізованого обчислювального середовища, яке забезпечує виконання ресурсомістких операцій аналізу даних та хостингу моделей. На рис. 3.7 наведено інтерфейс консолі керування Amazon SageMaker AI, де показано успішне створення та запуск інстансу ноутбука (Notebook Instance).

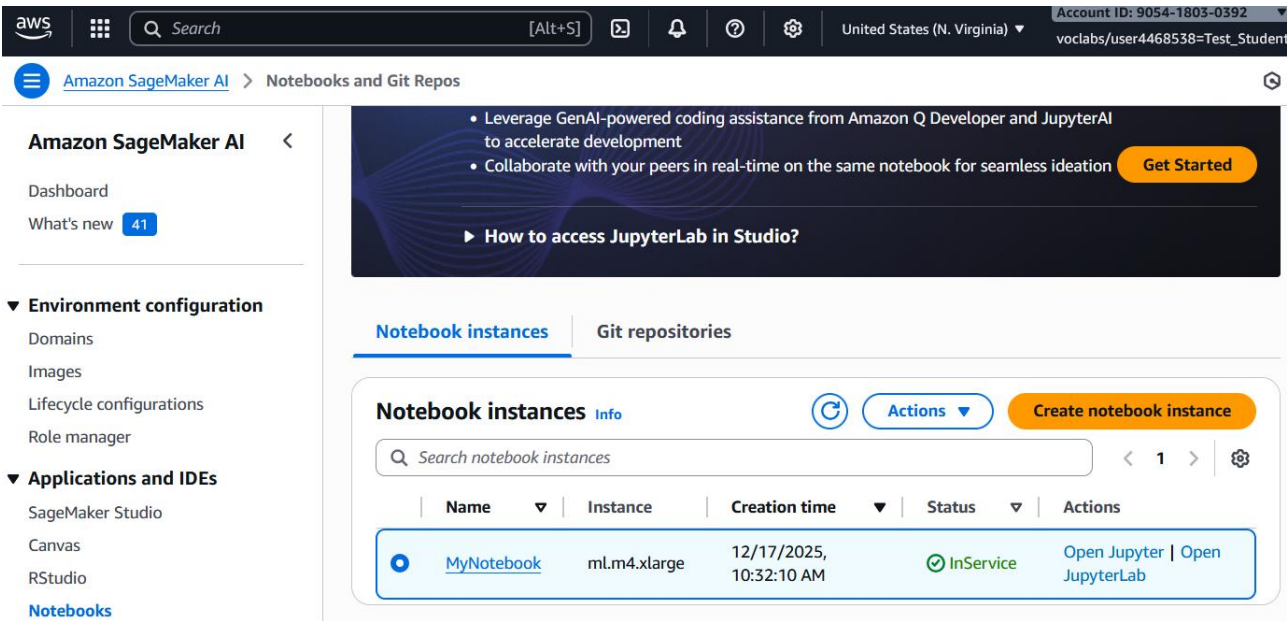


Рисунок 3.7 – Активний екземпляр середовища розробки в Amazon SageMaker

Перехід до середовища розробки відбувається за посиланням “Open JupyterLab” (рис. 3.7). Приклад ініціалізування доступу до бакету, який створено за замовчуванням в оточенні sandbox, з jupyter notebook, наведено на рис. 3.8 — 3.9.

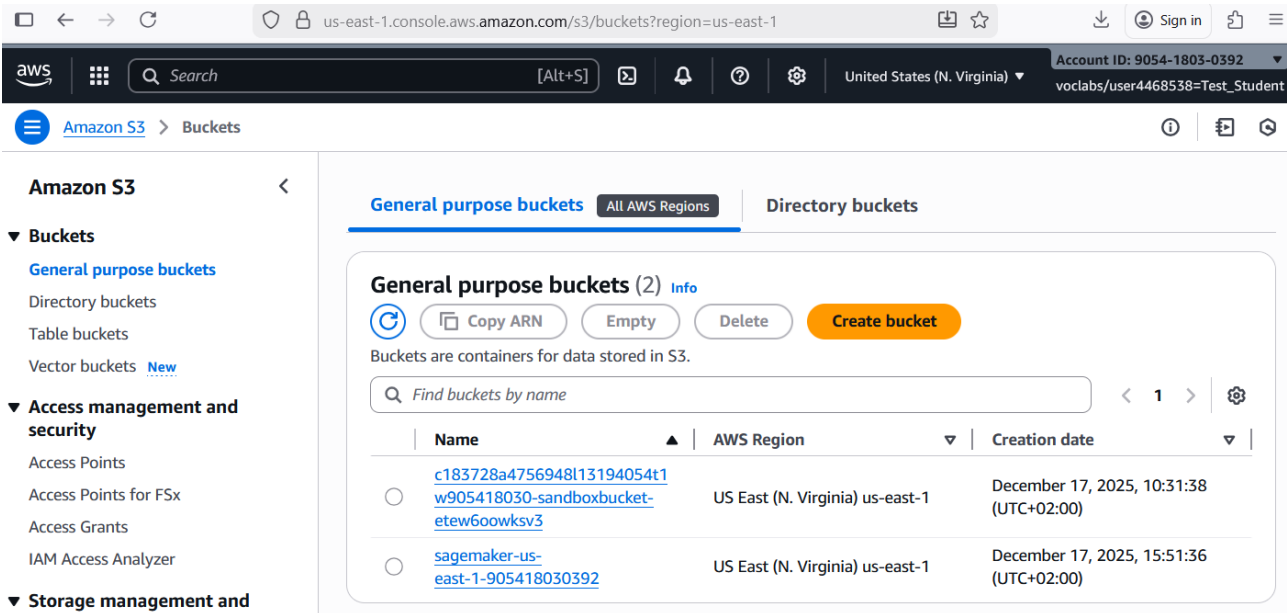


Рисунок 3.8 – Об’єктні сховища даних проєкту

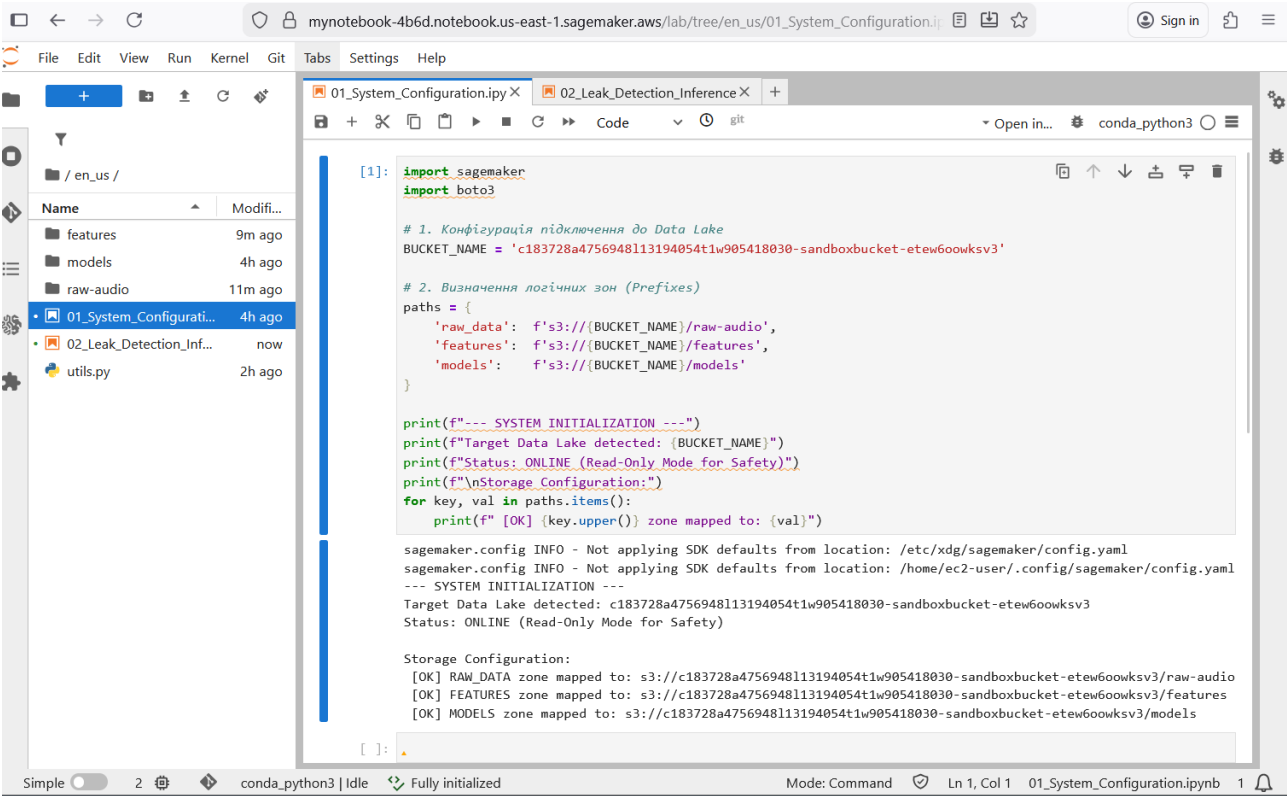


Рисунок 3.9 – Результат програмної ініціалізації параметрів доступу до даних

Після активації інстансу ноутбука запускається інтерактивне середовище розробки для виконання сценарію 02_Leak_detection_inference.ipynb. Даний блокнот виконує роль керуючого інтерфейсу для автоматизації процесу підготовки та розгортання моделі. Програмна логіка блокнота реалізує етап підготовки вхідних даних для інференсу. Програмний код наведено в додатку В, а результат роботи показано на рис. 3.10—3.12.

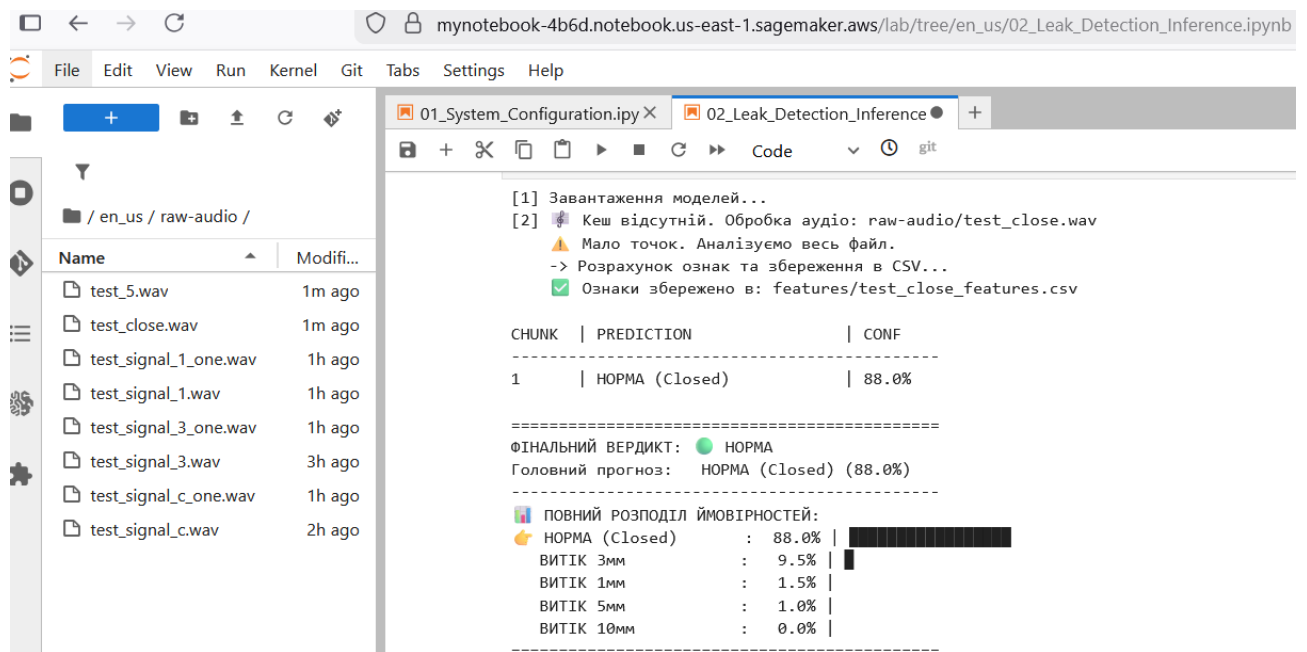


Рисунок 3.10 – Результат без застосуванням кешу при тестуванні діагностичного сигналу за відсутності витoku

Особливістю цього скрипта є впровадження механізмів кешування та серіалізації ознак, що дозволяє уникнути повторних обчислювальних витрат. Алгоритм програми передбачає перевірку наявності вже розрахованих ознак (рис. 3.10—3.11). У коді використовується логіка збереження результатів інженерії ознак у локальні файли (рис. 3.12): замість повного перерахунку спектральних характеристик для всього датасету при кожній ітерації, система завантажує готові вектори ознак, наявність готових до навчання та тестування масивів, гарантує ідентичність даних на етапі тестування працездатності розгорнутого ендпоінта (17 розрахованих ознак).

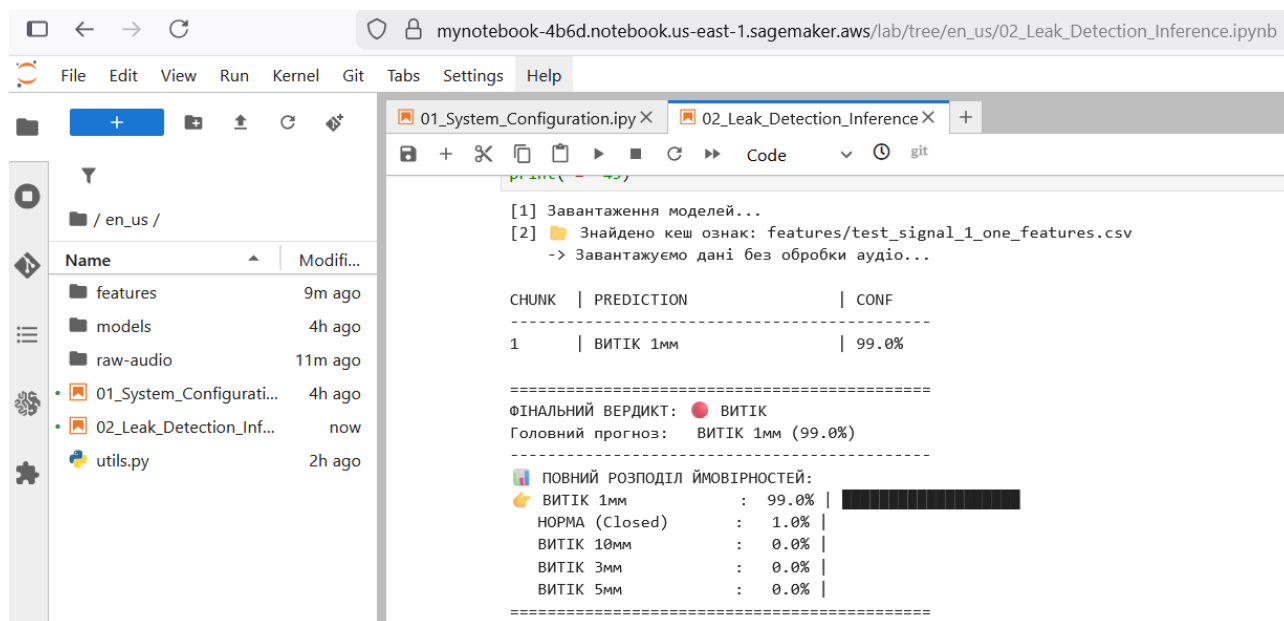


Рисунок 3.11 – Результат із застосуванням кешу при тестуванні діагностичного сигналу за наявності витoku 1 мм

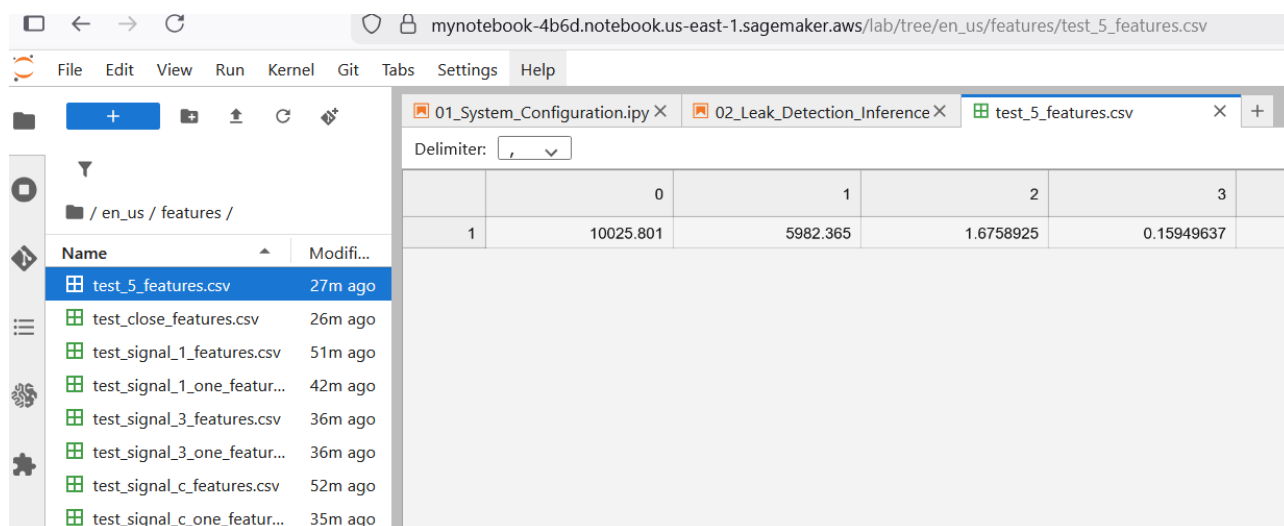


Рисунок 3.12 – Збережені результати інженерії ознак у файлах формату .csv

Наступним етапом є створення кастомного контейнеризованого рішення, що забезпечує стабільне завантаження моделі та виконання прогнозів. Основою підсистеми є навчена модель Random Forest та об'єкт попередньої обробки StandardScaler (програмний код навчання моделей наведено в додатку А). Особливістю програмної реалізації є пакування артефактів у стиснутий архів model.tar.gz:

					KPM.AКСм - 21.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

```
import tarfile
import os

with tarfile.open('model.tar.gz', 'w:gz') as tar:
    tar.add('models/model_forest.pkl', arcname='model_forest.pkl')
    tar.add('models/scaler_forest.pkl', arcname='scaler_forest.pkl')
```

```
print("✅ Артефакт model.tar.gz створено.")
```

Це забезпечує цілісність даних при їх передачі до S3-бакету sagemaker-us-east-1-905418030392/leak-detection-project/ (рис. 3.4). Програмна ініціалізація завантаження виконується за допомогою SDK-методу `sagemaker_session.upload_data`, що дозволяє автоматично делегувати права доступу через IAM-ролі та уникнути помилок типу «Access denied», оскільки `sandbox` розгорнуто в режимі `test_student`:

```
import sagemaker

sagemaker_session = sagemaker.Session()
bucket = sagemaker_session.default_bucket()
prefix = 'leak-detection-project'

model_artifact = sagemaker_session.upload_data(
    path='model.tar.gz',
    bucket=bucket,
    key_prefix=prefix
)

print(f"✅ Файл успішно завантажено сюди: {model_artifact}")
```

Для підготовки середовища розгортання створено папку `deploy_source`, куди поміщені файли `inference.py` та `requirements.txt`. Така структура гарантує, що при виконанні команди `.deploy()` SageMaker отримає повний пакет інструкцій для роботи з кешованими моделями.

Файл `requirements.txt` виконує роль специфікації програмного оточення, яка гарантує відтворюваність результатів моделювання у хмарному контейнері. Код файлу `requirements.txt` містить залежності, необхідні для реалізації прогнозування:

```
numpy
scikit-learn==1.2.2
joblib
```

Бібліотека `numpy` забезпечує перетворення вхідних JSON-структур у числові вектори. Модель `Random Forest` та засоби масштабування ознак (`StandardScaler`) реалізовано на `scikit-learn==1.2.2`. Вказування версії 1.2.2 виконано для забезпечення ідентичності методів опрацювання даних у

					KPM.AКСм - 21.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

локальному середовищі навчання та хмарному середовищі розгортання, з метою запобігання помилкам десеріалізації об'єктів. Модуль `joblib` — спеціалізований інструмент для збереження та завантаження (серіалізації) об'єктів Python. В архітектурі системи він відповідає за відновлення стану навченої моделі та масштабувальника з бінарних файлів, завантажених з S3-бакета. Така конфігурація дозволяє мінімізувати обсяг обчислювального образу, включаючи лише необхідні компоненти для математичного розрахунку прогнозу, що позитивно впливає на швидкість запуску інстансу та знижує затримки при опрацюванні вхідних запитів.

Логіка функціонування реалізована у скрипті `inference.py`, який керує життєвим циклом запиту всередині Docker-контейнера. Програмний код наведено нижче. Він структурований за чотирма функціональними блоками.

```
import joblib
import os
import numpy as np
import json
import warnings

warnings.filterwarnings("ignore")

def model_fn(model_dir):
    print(f"Loading model from: {model_dir}")
    try:
        model = joblib.load(os.path.join(model_dir, "model_forest.pkl"))
        scaler = joblib.load(os.path.join(model_dir, "scaler_forest.pkl"))
        return {"model": model, "scaler": scaler}
    except Exception as e:
        print(f"❌ CRITICAL ERROR LOADING MODEL: {e}")
        raise e

def input_fn(request_body, request_content_type):
    if request_content_type == 'application/json':
        input_data = json.loads(request_body)
        return np.array(input_data)
    else:
        raise ValueError(f"Unsupported content type: {request_content_type}")

def predict_fn(input_data, model_artifacts):
    model = model_artifacts["model"]
    scaler = model_artifacts["scaler"]

    if input_data.ndim == 1:
        input_data = input_data.reshape(1, -1)

    scaled_data = scaler.transform(input_data)
    return model.predict_proba(scaled_data)

def output_fn(prediction, response_content_type):
```

					KPM.AКСм - 21.00.00.000 ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		


```

if response_content_type == 'application/json':
    return json.dumps(prediction.tolist())
raise ValueError(f"Unsupported response type: {response_content_type}")

```

Функція завантаження моделі `model_fn` забезпечує десеріалізацію моделі та масштабувальника з кореневого каталогу. Реалізовано опрацювання винятків для виявлення критичних помилок завантаження. Функція опрацювання вхідних даних `input_fn` відповідає за десеріалізацію вхідних JSON-об'єктів. Реалізовано перевірку Content-Type та конвертацію вхідного вектору у формат `numpy.array` для забезпечення сумісності з алгоритмами Scikit-learn. Функція прогнозування `predict_fn` виконує трансформацію даних за допомогою `StandardScaler` та розрахунок ймовірностей приналежності до класів витоків через метод `predict_proba`. Програмно реалізовано автоматичну зміну розмірності вхідного масиву (`reshape(1, -1)`), що гарантує стабільну роботу при поодиноких запитах. Функція форматування виходу `output_fn` серіалізує отриманий вектор ймовірностей у формат JSON, що дозволяє інтегрувати підсистему з іншими сервісами моніторингу.

Такий підхід до програмної реалізації забезпечує модульність системи: логіка обробки даних відокремлена від даних моделі, що дозволяє проводити оновлення алгоритмів детектування витоків без необхідності перебудови хмарної інфраструктури.

3.4 Розгортання інтелектуального сервісу моніторингу

Процес активації сервісу ініціюється методом `.deploy()` об'єкта `SKLearnModel`:

```

from sagemaker.sklearn.model import SKLearnModel

sklearn_model = SKLearnModel(
    model_data=model_artifact,
    role=sagemaker.get_execution_role(),
    entry_point='inference.py',
    source_dir='deploy_source',
    framework_version='1.2-1',
    py_version='py3'
)

```

					KPM.АКСм - 21.00.00.000 ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

```

predictor = sklearn_model.deploy(
    instance_type='ml.t2.medium',
    initial_instance_count=1
)

```

Для забезпечення оптимального співвідношення між часом відгуку та витратами на хмарні ресурси обрано інстанс типу `ml.t2.medium`. В результаті виконання процедури в консолі Amazon SageMaker було сформовано точку доступу, що перейшла у статус `InService` (рис. 3.13). Даний статус підтверджує успішне розгортання Docker-контейнера, завантаження артефактів моделі та готовність системи до опрацювання вхідних HTTP-запитів.

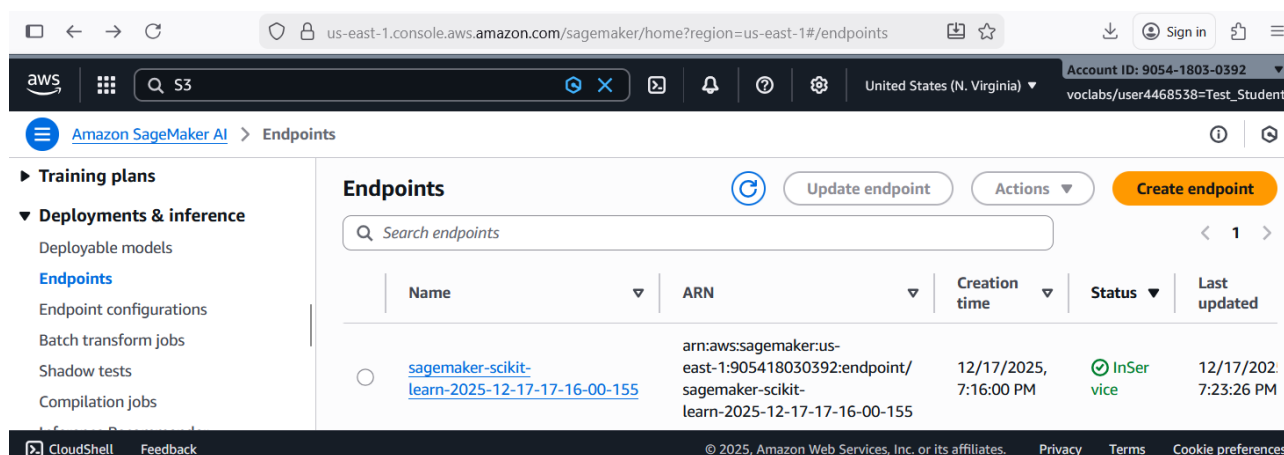


Рисунок 3.13 – Розгорнутий endpoint моделі для використання

Для тестування працездатності моделі з використанням розгорнутого ендпоінта виконано запити, код для формування яких показано на рис. 3.14.

Для інтерпретування одержаних результатів у середовищі Jupyter Notebook сформовано блок коду, який візуалізує впевненості моделі у прийнятому рішенні. Він повертає розподіл ймовірностей за п'ятьма категоріями станів трубопроводу. Візуальне інтерпретування одержаних результатів на рис. 3.14 показано на рис. 3.15.

Високе значення ймовірності (88%) для класу «Норма» свідчить про коректну роботу моделі та відсутність помилкових спрацювань при аналізі контрольних зразків.

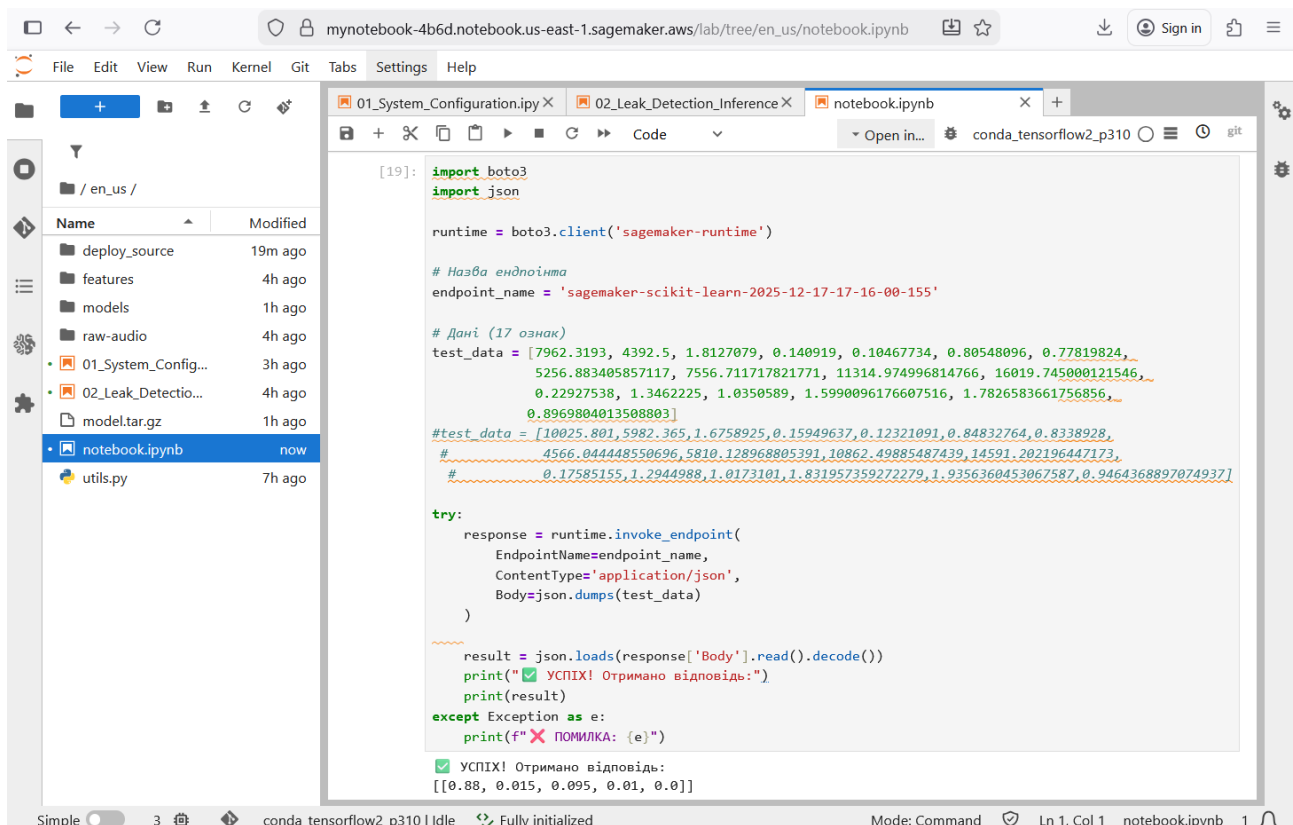


Рисунок 3.14 – Тестування працездатності розгорнутої точки доступу до моделі

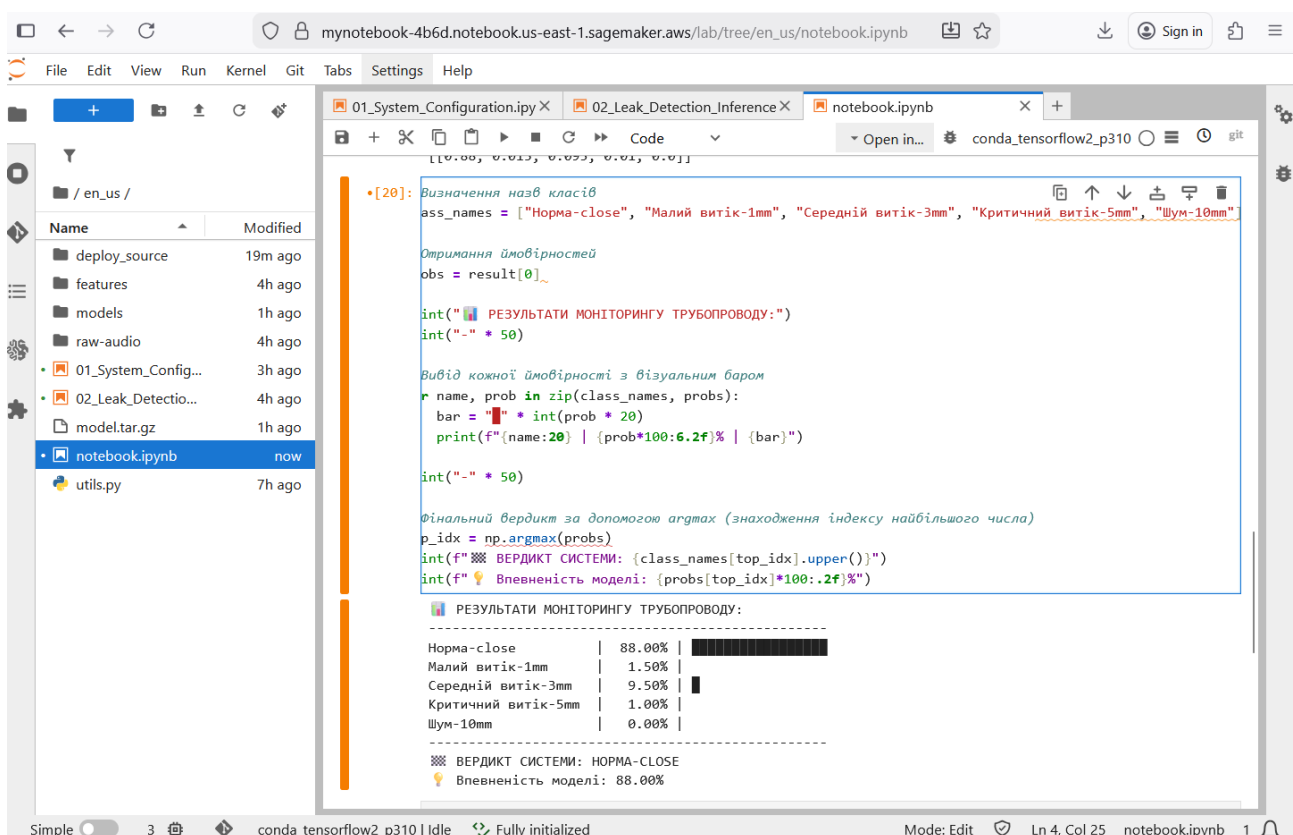


Рисунок 3.15 – Інтерпретування результатів

Аналіз експлуатаційних характеристик розгорнутої точки доступу до моделі проведено за допомогою метрик Amazon CloudWatch (рис.3.6). Динаміку роботи сервісу можна відстежувати з допомогою графіків Invocations (виклики) та ModelLatency (затримка моделі). Наявність активності на графіку викликів підтверджує успішну доставку тестових пакетів даних від клієнта (Jupyter Notebook) до моделі. Показник ModelLatency вимірює час виконання функцій `predict_fn` та `transform` у скрипті `inference.py`. Стабільність цього графіка свідчить про відсутність перевантаження процесора при виконанні математичних операцій.

Розгорнута система демонструє високу діагностичну здатність та стабільні динамічні характеристики, що підтверджує ефективність обраної архітектури на базі Amazon SageMaker для задач детектування витоків в режимі моніторингу трубопроводу.

З метою економії ресурсів після закінчення тестування працездатності розгорнутої системи необхідно видалити ендпоінт (рис. 3.16).

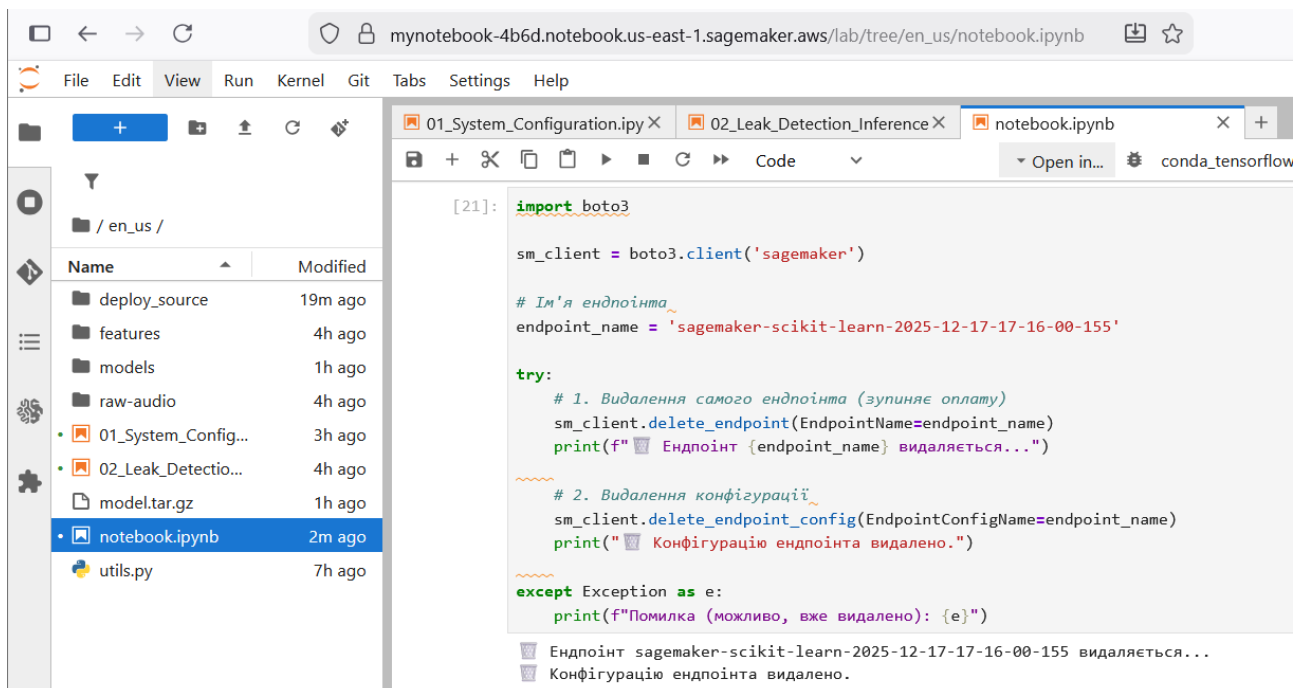


Рисунок 3.16 – Видалення ендпоінту

					KPM.AКСм - 21.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		68

Отже, в розділі представлено розроблення та розгортання хмарної архітектури моніторингу на базі сервісів AWS, що включає структуроване сховище артефактів Amazon S3, керовані інстанси розробки ml.m4.xlarge та спеціалізовані контейнери для інференсу. Програмно реалізовано підсистему інференсу у вигляді кастомного скрипта inference.py, що здійснює автоматизоване завантаження моделі Random Forest та об'єкта стандартизації StandardScaler за допомогою бібліотеки joblib. Оптимізовано процес підготовки даних шляхом впровадження механізмів кешування розрахованих ознак у середовищі SageMaker Notebook. Експериментально підтверджено працездатність розгорнутого ендпоїнта на базі інстансу ml.t2.medium. Тестування системи показало діагностичну здатність розробки. Технічна валідація за допомогою метрик CloudWatch продемонструвала високу швидкість відгуку системи та стабільність опрацювання вхідних запитів. Це дозволяє стверджувати, що запропоноване хмарне рішення є придатним для інтеграції в автоматизовані системи моніторингу стану трубопроводів у режимі реального часу.

Результати розроблення хмарної архітектури, наведені в п. 3.2—3.4 було представлено на 6й Міжнародній науково-практичній конференції «Global Directions in Scientific Research and Technological Development» [18].

					KPM.AКСм - 21.00.00.000 ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У кваліфікаційній роботі представлено розроблення інтелектуальної системи детектування витоків у трубопроводах з використанням AWS SageMaker. В роботі обґрунтовано вибір активного акустичного методу для детектування витоків з трубопроводів. Розроблено метод формування комбінованого вектора ознак для навчання моделей, що включає ентропійні та реляційні складові. Такий підхід дозволив підвищити інваріантність системи до змін робочого тиску та несучої частоти тестового сигналу, забезпечивши стабільність класифікації при змінних умовах експлуатації трубопроводу. Показано, що оптимальною архітектурою класифікатора для обраного підходу є класифікатор на основі ансамблевої моделі Random Forest. Проведені порівняльні дослідження підтвердили, що дана модель демонструє найкращий результат при тестуванні її на незалежній вибірці у порівнянні з іншими класифікаторами – перцептронною структурою та MLP.

Програмно реалізовано інтелектуальну систему з використанням середовища Amazon SageMaker. Продемонстровано етапи підготовки даних у Amazon S3, використання механізму кешування розрахованих ознак, розгортання ендпоінта та його тестування. Експериментальна апробація розгорнутої системи підтвердила її ефективність. Сервіс продемонстрував стійку роботу та низьку затримку формування відповіді-прогнозу.

Практична цінність роботи полягає у розробленні хмарного рішення, яке завдяки використанню сервісів AWS забезпечує автоматичне детектування стану трубопроводу з можливістю візуалізації аналітичних звітів для оперативного прийняття рішень. Результати роботи можуть бути використані при проектуванні та модернізації систем автоматизованого контролю трубопровідних систем для підвищення їх надійності та екологічної безпеки.

					KPM.AКСм - 21.00.00.000 ПЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ПОСИЛАНЬ НА ДЖЕРЕЛА

1. Безгачнюк Ю., Штаєр Л. Огляд сучасного стану методів контролю витоків з трубопроводів. International Science Journal of Engineering & Agriculture. 2024. Vol. 3, No. 3. P. 43–50. DOI: 10.46299/j.isjea.20240303.04.
2. Заміховський Л.М., Штаєр Л.О. Контроль витоків з магістральних та технологічних трубопроводів. Івано-Франківськ: ІФНТУНГ, 2013. 224 р. (монографія).
3. Romaniuk V. INTEGRATED APPROACH TO THE DIAGNOSTICS OF UKRAINE'S GAS TRANSMISSION SYSTEM. Sworld-Us Conference proceedings. 2022. No. usc31-00. P. 47–54. DOI: 10.30888/2709-2267.2025-31-00-003.
4. Ullah N., Ahmed Z., Kim J.-M. Pipeline Leakage Detection Using Acoustic Emission and Machine Learning Algorithms. Sensors. 2023. Vol. 23, No. 6. P. 3226. DOI: 10.3390/s23063226.
5. Aljameel S.S., Alabbad D.A., Alomari D., Alzannan R., Alismail S., Alkhudir A., Aljubran F., Nikolskaya E., Rahman A. Oil and Gas Pipelines Leakage Detection Approaches: A Systematic Review of Literature. International Journal of Safety and Security Engineering. 2024. Vol. 14, No. 3. P. 773–786. DOI: 10.18280/ijssse.140310.
6. Leak Detection. wikipedia. URL: https://en.wikipedia.org/wiki/Leak_detection#cite_note-2.
7. Mishchuk Y., Mishchuk D. IoT-Based Industrial Automation Systems. Girnichi, budivelni, dorozhni ta meliorativni mashini. 2020. No. 96. P. 42–50. DOI: 10.32347/gbdmm2020.96.0501.
8. Verma A., Garg S. Optimization of Enhanced Pipeline Leak Detection Using Advanced Dense Hebbian Neural Network Architectures for Improved Accuracy and Efficiency. International Journal on Engineering Artificial Intelligence Management, Decision Support, and Policies. 2025. Vol. 2, No. 1. P. 55–67. DOI: 10.63503/j.ijaimd.2025.41.

					KPM.AKCM - 21.00.00.000 ПЗ	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дата		

9. Liang H., Gao Y., Li H., Huang S., Chen M., Wang B. Pipeline Leakage Detection Based on Secondary Phase Transform Cross-Correlation. Sensors. 2023. Vol. 23, No. 3. P. 1572. DOI: 10.3390/s23031572.

10. Штаєр Л.О., Безгачнюк Ю.В., Белей О.І. Використання сервісів хмарних обчислень для опрацювання діагностичних сигналів в акустичному методі контролю витоків з трубопроводів. Інформаційні технології в освіті, техніці та промисловості: матеріали Всеукр. наук.-практ. конф. Івано-Франківськ: ІФНТУНГ, 2025. Р. 152–154. URL: <https://drive.google.com/file/d/1sim7SKa62RzaS5XZrTGt248h5VAFf1WQ/view>.

11. Заміховський Л.М., Ровінський В.А., Штаєр Л.О. Спосіб локалізації місця витoku речовини з трубопроводу та система для його здійснення. Пат. 83290 Україна, МПК G 01 N 29/04, G 01 M 3/24. № a200610331; заявл. 28.09.06; опубл. 25.06.2008, Бюл. № 12.

12. El-Zahab S., Abdelkader E.M., Fares A., Zayed T. Comparative Analysis of Machine Learning Techniques in Enhancing Acoustic Noise Loggers' Leak Detection. Water. 2025. Vol. 17, No. 16. P. 2427. DOI: 10.3390/w17162427.

13. Hong Z., Zhao D., Dong L., Liu S., Qiu F., Jin Y. Improving Pipeline Leak Detection Accuracy by Constructing a Multistate Leakage Sounds Training Set. Journal of Pipeline Systems Engineering and Practice. 2025. Vol. 16, No. 4. P. 04025056. DOI: 10.1061/JPSEA2.PSENG-1864.

14. Li Y., Wang X., Wu J. Fault Diagnosis of Rolling Bearing Based on Permutation Entropy and Extreme Learning Machine. 2016 Chinese Control and Decision Conference (CCDC): 2016 Chinese Control and Decision Conference (CCDC). Yinchuan, China: IEEE, 2016. P. 2966–2971. DOI: 10.1109/CCDC.2016.7531490.

15. Zhang B., Li W., Li X.-L., Ng S.-K. Intelligent Fault Diagnosis Under Varying Working Conditions Based on Domain Adaptive Convolutional Neural Networks. IEEE Access. 2018. Vol. 6. P. 66367–66384. DOI: 10.1109/ACCESS.2018.2878491.

					KPM.АКСм - 21.00.00.000 ПЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		

16. Amazon SageMaker AI Documentation. AWS Documentation. URL: <https://docs.aws.amazon.com/sagemaker/>.

17. Марчук Д.К., Граф М.С. МЕТОДИ ОЦІНКИ ЕФЕКТИВНОСТІ МОДЕЛЕЙ ВІЯВЛЕННЯ ОБ'ЄКТІВ У КОМП'ЮТЕРНОМУ ЗОРІ. Вісник Херсонського національного технічного університету. 2023. 2(85). Р. 181–186. DOI: 10.35546/kntu2078-4481.2023.2.25.

18. Безгачнюк Ю., Штаєр Л., Белей О., Титиш Н. Розроблення інтелектуальної системи детектування витоків з трубопроводів з використанням AWS SageMaker. Global Directions in Scientific Research and Technological Development: Collection of Scientific Papers with the Proceedings of the 6th International Scientific and Practical Conference (December 22-24, 2025), Valencia, Spain, European Open Science Space. URL: <https://www.eoss-conf.com/>

					КРМ.АКСм - 21.00.00.000 ПЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток А Програмний код модуля train.py

```
# --- train.py ---

import glob
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.metrics import classification_report
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import Perceptron
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import StratifiedKFold, cross_val_score
from sklearn.metrics import classification_report, confusion_matrix
import joblib
import seaborn as sns
import numpy as np

import utils

USE_ENTROPY_FEATURES = True
USE_RATIO_FEATURES = True
CLASSIFIER_TYPE = "FOREST"
N_CHUNKS_PER_FILE = 20

print(f"🔔 СТАТУС: Навчання з ентропією: {USE_ENTROPY_FEATURES}")
print(f"🔔 СТАТУС: Використання співвідношень (ratios): {USE_RATIO_FEATURES}")
print(f"🔔 СТАТУС: Класифікатор: {CLASSIFIER_TYPE}")

# === ОБРОБКА ВСІХ ФАЙЛІВ ===
test_signal_file = "data1/!_t_gaus_Barker5_960_0_1s_20.wav"
print(f"🔍 Знаходження позицій початку корисних блоків...")
offset_for_starts = 0
if "480" in test_signal_file:
    offset_for_starts = 21 * 192000

starts = utils.find_signal_starts(test_signal_file,
                                  utils.SR_EXPECTED,
                                  offset_samples=offset_for_starts)

print(starts)
print(f"Знайдено {len(starts)} блоків.")

print(f"📁 Збір файлів для навчання...")

path1 = "data_t_gaus_Barker5_960_0_1s_20/*.wav"
path2 = "data_t_pryamok_Barker5_480_0_1s_20/*.wav"
path3 = "data_for_test/*.wav"

file_list_gaus = glob.glob(path1)
file_list_pryamok = glob.glob(path2)
file_list_pressure = glob.glob(path3)

file_list = file_list_gaus + file_list_pryamok + file_list_pressure

print(f"Знайдено {len(file_list)} файлів: {len(file_list_gaus)} (gaus) + {len(file_list_pryamok)} (pryamok) + {len(file_list_pressure)} (pressure).")

label_map = {
    "closed": 0,
    "leak1mm": 1,
```

```

        "leak3mm": 2,
        "leak5mm": 3,
        "leak10mm": 4
    }

    print(f"Використовуються класи: {list(label_map.keys())}")
    X, y = [], []

    sorted_label_keys = sorted(label_map.keys(), key=len, reverse=True)

    print(f"Використовуються класи: {list(label_map.keys())}")
    print(f"Порядок перевірки міток: {sorted_label_keys}")

    for filepath in file_list:
        label_str = None

        for k in sorted_label_keys:
            if k in filepath:
                label_str = k
                break

        if label_str is None:
            print(f"УВАГА: Не вдалося визначити мітку для файлу {filepath}. Файл пропущено.")
            continue
        current_offset = 0
        if "480" in filepath:
            current_offset = 21 * 192000
        chunks = utils.get_chunks_with_starts(filepath,
                                                starts,
                                                offset_samples=current_offset,
                                                n_chunks=N_CHUNKS_PER_FILE)

        for chunk in chunks:
            X.append(utils.extract_features(chunk,
                                            use_entropy=USE_ENTROPY_FEATURES,
                                            use_ratios=USE_RATIO_FEATURES))
            y.append(label_map[label_str])

    X = utils.np.array(X)
    y = utils.np.array(y)

    # === МАСШТАБУВАННЯ ===
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)

    # === МОДЕЛЬ ===
    if CLASSIFIER_TYPE == "PERCEPTRON":
        clf = Perceptron(random_state=42,
                          max_iter=1000,
                          tol=1e-3,
                          class_weight='balanced')

    elif CLASSIFIER_TYPE == "MLP":
        clf = MLPClassifier(random_state=42,
                            max_iter=1500,
                            hidden_layer_sizes=(50,), alpha=0.001)

    else:
        clf = RandomForestClassifier(n_estimators=200,
                                    random_state=42,
                                    class_weight='balanced')

```

```

print(f"Навчання моделі типу: {CLASSIFIER_TYPE}...")
clf.fit(X_scaled, y)
y_pred = clf.predict(X_scaled)

if CLASSIFIER_TYPE == "FOREST":
    base_feature_names = [
        "energy_ch1", "energy_ch2", "energy_ratio",
        "rms1", "rms2", "peak1", "peak2",
        "spec_centroid1", "spec_centroid2",
        "spec_bw1", "spec_bw2",
        "norm_corr"
    ]

    if USE_RATIO_FEATURES:
        base_feature_names.extend(["rms_ratio", "peak_ratio"])

    if USE_ENTROPY_FEATURES:
        base_feature_names.extend(["entropy1", "entropy2"])
    if USE_RATIO_FEATURES:
        base_feature_names.append("entropy_ratio")

    feature_names = base_feature_names

    importances = clf.feature_importances_
    indices = utils.argsort(importances)[-1]

    plt.figure(figsize=(10, 6))
    plt.title("Важливість ознак у RandomForest")

    plt.bar(range(len(importances)), importances[indices], align="center")
    plt.xticks(range(len(importances)), [feature_names[i] for i in indices], rotation=45, fontsize=14)
    plt.ylabel("Важливість", fontsize=14)
    plt.tight_layout()

    print("Важливість ознак (RandomForest):")
    for i in indices:
        print(f"{feature_names[i]}: {importances[i]:.4f}")

else:
    print(f"Аналіз важливості ознак недоступний для {CLASSIFIER_TYPE}")

# === PCA ВІЗУАЛІЗАЦІЯ ===
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)

plt.figure(figsize=(8,6))
for label_str, label_id in label_map.items():
    mask = y == label_id
    plt.scatter(X_pca[mask,0], X_pca[mask,1], label=label_str, alpha=0.7)
plt.legend()
plt.xlabel("PCA 1")
plt.ylabel("PCA 2")
plt.title("PCA розподіл класів")
plt.grid(True)

# 5-фолдова стратифікована крос-валідація
skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

scores = cross_val_score(clf, X_scaled, y, cv=skf)
print(f"Середня точність: {scores.mean():.4f} ± {scores.std():.4f}")

y_true_all = []

```

```

y_pred_all = []

for train_idx, test_idx in skf.split(X_scaled, y):
    X_train, X_test = X_scaled[train_idx], X_scaled[test_idx]
    y_train, y_test = y[train_idx], y[test_idx]
    clf.fit(X_train, y_train)
    y_pred = clf.predict(X_test)
    y_true_all.extend(y_test)
    y_pred_all.extend(y_pred)

print(classification_report(y_true_all, y_pred_all, target_names=label_map.keys()))
print("Матриця плутанини:")
print(confusion_matrix(y_true_all, y_pred_all))

print("Матриця плутанини (крос-валідація):")
cm = confusion_matrix(y_true_all, y_pred_all)
print(cm)

plt.figure(figsize=(10, 8))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=label_map.keys(),
            yticklabels=label_map.keys(),
            annot_kws={"size": 16})

plt.title(f"Рисунок 5(A). Матриця плутанини (Крос-валідація)\nКласифікатор:
{CLASSIFIER_TYPE.upper()}", fontsize=16)
plt.ylabel("Істина", fontsize=16)
plt.xlabel("Прогноз", fontsize=16)
plt.tight_layout()
config = {
    'use_entropy': USE_ENTROPY_FEATURES,
    'use_ratios': USE_RATIO_FEATURES,
    'classifier_type': CLASSIFIER_TYPE
}

MODEL_NAME = "forest"
if CLASSIFIER_TYPE == "FOREST":
    MODEL_NAME = "forest"
elif CLASSIFIER_TYPE == "MLP":
    MODEL_NAME = "mlp"
elif CLASSIFIER_TYPE == "PERCEPTRON":
    MODEL_NAME = "perceptron"

joblib.dump(clf, f"model_{MODEL_NAME}.pkl")
joblib.dump scaler, f"scaler_{MODEL_NAME}.pkl")
joblib.dump(config, f"config_{MODEL_NAME}.pkl")

print(f"✅ Модель ({MODEL_NAME}), масштабувальник та конфіг збережені.")
plt.show()

```

Додаток Б Програмний код модуля predict.py

```
import glob
import joblib
import utils
from sklearn.metrics import classification_report, confusion_matrix, f1_score
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

label_map = {0: "closed", 1: "leak1mm", 2: "leak3mm", 3: "leak5mm", 4: "leak10mm"}
label_map_inv = {v: k for k, v in label_map.items()}

N_CHUNKS_PER_FILE = 80
OFFSET_1_SEC = 1 * utils.SR_EXPECTED + 20 * 2.1 * 192000

MODEL_NAMES = ["forest", "mlp", "perceptron"]

plot_results = {
    'model_names': [],
    'f1_macro_scores': [],
    'confusion_matrices': [],
    'cm_labels': []
}

test_signal_file = "data_for_test/test/t_Barker5_480Hz_0_1_80sht_zsuv1s.wav"

print(f"🔍 Знаходження позицій початку з {test_signal_file} (зсув: 1с)...")
starts = utils.find_signal_starts(test_signal_file,
                                   utils.SR_EXPECTED,
                                   offset_samples=OFFSET_1_SEC)
print(f"Знайдено {len(starts)} блоків для розмітки.")

test_folder_path = "data_for_test/*.wav"
file_list = glob.glob(test_folder_path)
print(f"📁 Знайдено {len(file_list)} файлів для тестування у '{test_folder_path}'")

sorted_label_keys = sorted(label_map_inv.keys(), key=len, reverse=True)

for model_name in MODEL_NAMES:
    print(f"\n=====")
    print(f"🧪 ТЕСТУВАННЯ МОДЕЛІ: {model_name.upper()}")
    print(f"=====")

    try:
        config = joblib.load(f"config_{model_name}.pkl")
        use_entropy = config.get('use_entropy', False)
        use_ratios = config.get('use_ratios', False)

        clf = joblib.load(f"model_{model_name}.pkl")
        scaler = joblib.load(f"scaler_{model_name}.pkl")
        print(f" (Використовує ентропію: {use_entropy}, Співвідношення: {use_ratios})")
    except Exception as e:
        print(f"❌ ПОМИЛКА: Модель '{model_name}' не знайдено або конфіг пошкоджено: {e}")
        continue

    all_true_labels = []
    all_pred_labels = []
```

```

for filepath in file_list:
    if filepath.endswith(test_signal_file):
        continue

    true_label_str = None
    for k in sorted_label_keys:
        if k in filepath:
            true_label_str = k
            break

    if true_label_str is None:
        print(f" ! Пропускаю файл (не вдалося визначити мітку): {filepath}")
        continue

    if true_label_str not in label_map_inv:
        print(f" ! Пропускаю файл (мітка '{true_label_str}' невідома моделі).")
        continue

    true_label_id = label_map_inv[true_label_str]

    chunks = utils.get_chunks_with_starts(filepath,
                                           starts,
                                           offset_samples=OFFSET_1_SEC,
                                           n_chunks=N_CHUNKS_PER_FILE)

    X_new = [utils.extract_features(chunk,
                                     use_entropy=use_entropy,
                                     use_ratios=use_ratios) for chunk in chunks]

    if not X_new:
        print(f" ! Не знайдено чанків для {filepath}")
        continue

    X_new_scaled = scaler.transform(X_new)
    preds = clf.predict(X_new_scaled)

    all_pred_labels.extend(preds)
    all_true_labels.extend([true_label_id] * len(preds))

    if not all_true_labels:
        print(" ❌ Немає даних для тестування.")
        continue

    unique_labels_in_test = np.unique(all_true_labels).astype(int)
    target_names_for_report = [label_map[label_id] for label_id in unique_labels_in_test]

    report_str = classification_report(all_true_labels,
                                       all_pred_labels,
                                       labels=unique_labels_in_test,
                                       target_names=target_names_for_report,
                                       zero_division=0)

    matrix = confusion_matrix(all_true_labels,
                              all_pred_labels,
                              labels=unique_labels_in_test)

    f1_macro = f1_score(all_true_labels,
                        all_pred_labels,
                        labels=unique_labels_in_test,
                        average='macro',
                        zero_division=0)

```

```

plot_results['model_names'].append(model_name.upper()) # FOREST, MLP, PERCEPTRON
plot_results['f1_macro_scores'].append(f1_macro)
plot_results['confusion_matrices'].append(matrix)
plot_results['cm_labels'].append(target_names_for_report)

print(f"\n--- ФІНАЛЬНИЙ ЗВІТ для {model_name.upper()} ---")
print(f"(Звіт згенеровано лише для класів, знайдених у тестах: {target_names_for_report})")
print("\nМатриця плутанини:")
print(f"(Стовпці та рядки відповідають: {target_names_for_report})")
print(matrix)
print("\nЗвіт по класах:")
print(report_str)
print(f"\n -> F1-Макро (по тестових класах): {f1_macro:.4f}")

print("\nВсі моделі протестовано.")

print("\n" + "=" * 40)
print("=" * 40)

plt.rcParams.update({'font.size': 12})

try:
    model_names_fig4 = plot_results['model_names']
    f1_scores_fig4 = plot_results['f1_macro_scores']

    plt.figure(figsize=(10, 6))
    colors = ['#1f77b4', '#ff7f0e', '#2ca02c']
    bars = plt.bar(model_names_fig4, f1_scores_fig4, color=colors[:len(model_names_fig4)])

    plt.title("Рисунок 4. Порівняння F1-міри (Macro Avg) класифікаторів на \"сліпому\" тестовому наборі", fontsize=16,
              pad=20)
    plt.ylabel("F1-міра (Macro Avg)", fontsize=14)
    plt.xlabel("Архітектура класифікатора", fontsize=14)
    plt.ylim(0, 1.05)
    plt.grid(axis='y', linestyle='--', alpha=0.7)

    for bar in bars:
        yval = bar.get_height()
        plt.text(bar.get_x() + bar.get_width() / 2.0, yval + 0.01, f'{yval:.2f}',
                 ha='center', va='bottom', fontsize=12)

    plt.tight_layout()
    fig4_path = 'figure_4_blind_test_comparison.png'
    plt.savefig(fig4_path, dpi=300)
    print(f"✅ Графік '{fig4_path}' успішно збережено.")

except Exception as e:
    print(f"❌ Не вдалося згенерувати Рисунок 4: {e}")

try:
    idx_forest = plot_results['model_names'].index('FOREST')
    idx_mlp = plot_results['model_names'].index('MLP')

    cm_forest = plot_results['confusion_matrices'][idx_forest]
    cm_mlp = plot_results['confusion_matrices'][idx_mlp]
    cm_labels = plot_results['cm_labels'][idx_forest]

    fig, axes = plt.subplots(1, 2, figsize=(14, 6))

```



```

sns.heatmap(cm_forest, annot=True, fmt='d', cmap='Blues',
             xticklabels=cm_labels, yticklabels=cm_labels, ax=axes[0],
             annot_kws={"size": 16})
axes[0].set_title("RandomForest", fontsize=16)
axes[0].set_ylabel("Істина", fontsize=14)
axes[0].set_xlabel("Прогноз \n а)", fontsize=14)

sns.heatmap(cm_mlp, annot=True, fmt='d', cmap='Oranges',
            xticklabels=cm_labels, yticklabels=cm_labels, ax=axes[1],
            annot_kws={"size": 16})
axes[1].set_title("MLP", fontsize=16)
axes[1].set_ylabel("Істина", fontsize=14)
axes[1].set_xlabel("Прогноз \n б)", fontsize=14)

fig.suptitle("Рисунок 5. Матриці плутанини для (А) RandomForest та (Б) MLP на \"сліпому\"
тестовому наборі",
             fontsize=18, y=1.03)

plt.tight_layout()
fig5_path = 'figure_5_confusion_matrices.png'
plt.savefig(fig5_path, dpi=300, bbox_inches='tight')
print(f"✅ Графік '{fig5_path}' успішно збережено.")

except Exception as e:
    print(f"❌ Не вдалося згенерувати Рисунок 5 (можливо, не знайдено 'FOREST' або 'MLP'):
{e}")

plt.show()

print("\n🎉 Оцінку завершено, графіки згенеровано.")

```

Додаток В Програмний код ноутбука 02_Leak_detection_inference.ipynb

```
import os, joblib, numpy as np, warnings, librosa
import pandas as pd

try: import utils
except ImportError: raise SystemExit(" ❌ utils.py відсутній.")
warnings.filterwarnings("ignore")

# --- КОНФІГУРАЦІЯ ---
MODEL, SCALER = 'models/model_forest.pkl', 'models/scaler_forest.pkl'
SIGNAL = 'raw-audio/test_signal_1_one.wav'
# збереження ознак
FEATURES_DIR = 'features'
CSV_NAME = os.path.splitext(os.path.basename(SIGNAL))[0] + '_features.csv'
CSV_PATH = os.path.join(FEATURES_DIR, CSV_NAME)

CHUNKS = 5
LABELS = {0: "НОРМА (Closed)", 1: "ВИТІК 1мм", 2: "ВИТІК 3мм", 3: "ВИТІК 5мм", 4: "ВИТІК
10мм"}
CONFIG = {'use_entropy': True, 'use_ratios': True}

# --- ЗАВАНТАЖЕННЯ МОДЕЛЕЙ ---
print(f"[1] Завантаження моделей...")
model, scaler = joblib.load(MODEL), joblib.load(SCALER)

# --- ЛОГІКА DATA ZONE (КЕШУВАННЯ) ---
features_data = []

if os.path.exists(CSV_PATH):
    print(f"[2] 📁 Знайдено кеш ознак: {CSV_PATH}")
    print(" -> Завантажуємо дані без обробки аудіо...")
    df = pd.read_csv(CSV_PATH)
    features_data = df.values.tolist()
else:
    print(f"[2] 🔊 Кеш відсутній. Обробка аудіо: {SIGNAL}")

    try:
        starts = utils.find_signal_starts(SIGNAL, utils.SR_EXPECTED)
        if len(starts) < 2:
            print(" ⚠️ Аналізуємо весь файл.")
            y, _ = librosa.load(SIGNAL, sr=utils.SR_EXPECTED, mono=False)
            chunks = [np.vstack([y, y])] if y.ndim == 1 else [y]
        else:
            chunks = utils.get_chunks_with_starts(SIGNAL, starts, n_chunks=CHUNKS)
    except Exception as e:
        print(f" ❌ Помилка обробки: {e}"); chunks = []

    if not chunks: raise SystemExit(" ❌ Немає даних для аналізу.")

# --- Екстракція ознак ---
print(" -> Розрахунок ознак та збереження в CSV...")
for chunk in chunks:
    feats = utils.extract_features(chunk, **CONFIG)
    features_data.append(feats)

# --- Збереження в Data Zone ---
if not os.path.exists(FEATURES_DIR):
    os.makedirs(FEATURES_DIR)
```

```

df_save = pd.DataFrame(features_data)
df_save.to_csv(CSV_PATH, index=False)
print(f"✅ Ознаки збережено в: {CSV_PATH}")

# --- АНАЛІЗ (Інференс) ---
print(f"\n{'CHUNK':<6} | {'PREDICTION':<25} | {'CONF'}\n" + "-"*45)
probs_sum = np.zeros(len(LABELS))

# Робота з features_data
for i, feats in enumerate(features_data):
    # Масштабування
    feats_scaled = scaler.transform(np.array(feats).reshape(1, -1))

    # Прогноз
    probs = model.predict_proba(feats_scaled)[0]
    pred = np.argmax(probs)

    probs_sum += probs
    print(f"{i+1:<6} | {LABELS.get(pred):<25} | {probs[pred]:.1%}")

# --- ПІДСУМОК ---
avg_probs = probs_sum / len(features_data)
final_idx = np.argmax(avg_probs)
status = "🟢 НОРМА" if final_idx == 0 else "🔴 ВІТІК"

print("\n" + "-"*45)
print(f"ФІНАЛЬНИЙ ВЕРДИКТ: {status}")
print(f"Головний прогноз: {LABELS[final_idx]} ({avg_probs[final_idx]:.1%}")
print("-" * 45)
print("📊 ПОВНИЙ РОЗПОДІЛ ЙМОВІРНОСТЕЙ:")

sorted_indices = np.argsort(avg_probs)[::-1]

for idx in sorted_indices:
    label = LABELS[idx]
    prob = avg_probs[idx]

    marker = "👉" if idx == final_idx else " "
    bar_len = int(prob * 20)
    bar = "█" * bar_len

    print(f"{marker} {label:<20} : {prob:>6.1%} | {bar}")

print("-" * 45)

```

БІБЛІОГРАФІЧНА ДОВІДКА

Тема кваліфікаційної роботи магістра – «Інтелектуальна система детектування витоків у трубопроводах з використанням AWS SageMaker»

Обсяг пояснювальної записки в аркушах – 84.

Перелік креслень графічної частини:

КМР.АКСм–21.00.00.000 С1 – Контекстна діаграма процедури навчання моделей та тестування. Схема структурна (1 аркуш)

КМР.АКСм–21.00.00.000 – Порівняльний аналіз якості. Робочі вікна (1 аркуш)

КМР.АКСм–21.00.00.000 С1– Архітектура системи збору, опрацювання та аналізу діагностичних сигналів з використанням сервісів AWS. Схема структурна (1 аркуш)

КМР.АКСм–21.00.00.000 – Мережева точка доступу до моделі. Робочі вікна (1 аркуш)

Дата закінчення магістерської роботи « 22 » грудня 2025р.

Студент-магістр_____Титиш Н. В.