

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютеризованих систем управління

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

Зі спеціальності 153 – «Мікро- та наносистемна техніка»

на тему:

«Обробка сенсорного відгуку високопористих газочутливих шарів на основі
ZnO методами машинного навчання»

Зиморой Віталій Віталійович

Завідувач кафедри

_____ проф. П.В. Леонтєв

Керівник

_____ Ю.О. Космінська

«__» _____ 2023 р.

«__» _____ 2023 р.

Суми 2023

Сумський державний університет

(назва вузу)

Факультет ЕЛІТ Кафедра Комп'ютеризованих систем управління

Спеціальність «Мікро- та наносистемна техніка»

Затверджую:

зав.кафедрою _____

“ _____ ” _____ 20__ р.

ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТОВІ

Зиморой Віталій Віталійович

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Обробка сенсорного відгуку високопористих газочутливих шарів на основі ZnO методами машинного навчання.

затверджую наказом по інституту від “15” грудня 2023 р. № 1462-VI.

2. Термін задачі студентом закінченого проекту (роботи) _____

3. Вхідні данні до проекту (роботи) вольт-амперні характеристики фрактально-перколяційних наносистем ZnO/NiO, виміряні в атмосфері метану чи метанолу у різних концентраціях, при зміні напруги від 0 до 5 В.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Аналіз роботи. Постановка задачі дослідження. 2) Теоретичні аспекти вимірювань газових сумішей. 3) Опис методів вимірювання та отримання експериментальних даних. 4) Розробка інформаційного та програмного забезпечення системи машинного навчання.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Рисунок 1.1 Графік зміни поверхневої провідності напівпровідникового зразка; Рисунок 1.2 – 1.4 Три етапи формування фрактально-перколяційної наносистеми; Рисунок 3.1 ВАХ метану та метанолу; Рисунок 4.1 Вивід списку даних датасетів; Рисунок 4.2 Вивід списку даних після нормалізації; Рисунок 4.3 Розподіл даних на навчальний, валідаційний і тестовий набори; Рисунок 4.4 Звіт про класифікацію логістичної регресії; Рисунок 4.5 Звіт про класифікацію мережі Байєса; Рисунок 4.6 Звіт про класифікацію дерева рішень (Decision Tree); Рисунок 4.7 Звіт про результат ROC-AUC Score; Рисунок 4.8 Звіт про результат методу Cross-Validation.; Рисунок 4.9 - 4.11 Learning curves графік логістичної регресії, графік наївного Байєса і дерева рішень (Decision Tree); Рисунок 4.12 Належність даних до двох класів (двох типів газу) у поданні в координатах двох ознак (I1, U1) після навчання моделі; Рисунок 4.13 Звіт роботи моделі XGBoost; Рисунок 4.14 Графік прогнозів проти спостережень (scatter plot); Рисунок 4.15 Результати кластеризації даних газів за допомогою МН; Рисунок 4.16 Звіт тестових прогнозів розбиття метану та метанолу.

6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

| Розділ | Консультант | Підпис, дата | |
|--------|-------------|----------------|------------------|
| | | Завдання видав | Завдання прийняв |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

7. Дата видачі завдання _____

Керівник

(підпис)

Завдання прийняв до виконання

(підпис)

КАЛЕНДАРНИЙ ПЛАН

| № п/п | Назва етапів дипломного проекту (роботи) | Термін виконання проекту (роботи) | Примітка |
|-------|--|-----------------------------------|----------|
| 1. | Аналіз роботи. Постановка задачі дослідження | | |
| 2. | Теоретичні аспекти вимірювань газових сумішей | | |
| 3. | Опис методів вимірювання та отримання експериментальних даних | | |
| 4. | Розробка інформаційного та програмного забезпечення системи машинного навчання | | |
| 5. | Оформлення пояснювальної записки до дипломної роботи | | |

Студент – дипломник

(підпис)

Керівник проекту

РЕФЕРАТ

Об'єктом досліджень у цій магістерській роботі є вимірювальні дані у вигляді вольт-амперних характеристик, одержані від фрактально-перколяційних наносистем на основі ZnO у атмосфері метану або метанолу, які відображають чутливість шарів до різних газів.

Мета роботи: розробка програмного інструменту на основі машинного навчання для ефективного розпізнавання типу газу в середовищі та його концентрації газу за вольтамперними характеристиками фрактально-перколяційних наносистем, що ґрунтується на їх структурних та морфологічних особливостях.

У основу роботи покладене спрощене подання газового середовища як одного типу газу із двох можливих. Це дозволяє оцінити принципову можливість застосування методів машинного навчання до покращення селективності газових сенсорів за рахунок комп'ютерної обробки даних вимірювань. Запропонований підхід може бути застосований для ширшого набору газів та газових сумішей за умови подальшого вдосконалення моделі.

Робота складається з вступу, теоретичних аспектів роботи, таких як огляд сучасних підходів до розпізнавання газових сумішей, опис фрактально-перколяційних наносистем та їхніх електрофізичних властивостей, розгляд методів машинного навчання а також їх можливого застосування у сфері сенсорних технологій, розробки програмного інструменту та викладення правил техніки безпеки при роботі за комп'ютером, та висновків.

Робота викладена на 83 сторінок , 21 рисуноків, 3 формули, 1 таблиця, 37 літературних джерел, 1 додаток.

КЛЮЧОВІ СЛОВА: ФРАКТАЛЬНО-ПЕРКОЛЯЦІЙНІ НАНОСИСТЕМИ, ГАЗОВІ СЕНСОРИ, СЕЛЕКТИВНІСТЬ, МАШИННЕ НАВЧАННЯ, ВОЛЬТАМПЕРНІ ХАРАКТЕРИСТИКИ.

ЗМІСТ

| | |
|---|-----------|
| РЕФЕРАТ | 4 |
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ | 7 |
| ВСТУП | 8 |
| 1 ТЕХНОЛОГІЇ ГАЗОВИХ СЕНСОРІВ: ВІД ОСНОВНИХ ЕЛЕМЕНТІВ ДО ФРАКТАЛЬНО-ПЕРКОЛЯЦІЙНИХ НАНОСИСТЕМ НА ОСНОВІ ZNO/NIO | 9 |
| 1.1 Технологія та класифікація сенсорів. | 9 |
| 1.2 Газові сенсорні на основі напівпровідникових оксидів. | 16 |
| 1.3 Основні елементи та фізичні властивості газових сенсорів. | 17 |
| 1.4 Сенсорні, які реагують на гази за допомогою ZnO. | 20 |
| 1.5 Сенсорні властивості фрактально-перколяційної наносистеми на основі ZnO/NiO. | 22 |
| 2 МАШИННЕ НАВЧАННЯ У АНАЛІЗІ ДАНИХ ГАЗОВИХ СЕНСОРІВ | 27 |
| 2.1 Основні принципи машинного навчання. | 27 |
| 2.2 Машинне навчання для аналізу даних у газових сенсорах. | 32 |
| 3 РОЗРОБКА ПРОГРАМНОГО ІНСТРУМЕНТУ | 38 |
| 3.1 Вибір та обґрунтування мови програмування (Python). | 38 |
| 3.2 Алгоритми обробки та аналіз вольт-амперних характеристик. | 39 |
| 4 МЕТОДОЛОГІЯ ТА РЕЗУЛЬТАТИ МН В АНАЛІЗІ ВАХ ВИСОКОПОРИСТИХ СИСТЕМ ZNO | 45 |
| 4.1 Використання машинного навчання для розпізнавання газів та їх концентрації..... | 45 |
| 4.1.1 Набір даних для аналізу. | 45 |
| 4.1.2 Методи класифікації у задачі газового аналізу..... | 45 |
| 4.1.3 Передбачення концентрації газу. | 49 |
| 4.1.4 Методи кластеризації у задачі газового аналізу. | 50 |

| | | |
|----------|--|-----------|
| 4.2 | Результати роботи програмного інструменту на різних газових сумішах. | 51 |
| 4.2.1 | Результати машинного навчання методами класифікації. | 51 |
| 4.2.2 | Результати машинного навчання по передбаченню концентрації газу. . | 61 |
| 4.2.3 | Результати машинного навчання методом кластеризації. | 64 |
| 5 | ПРАВИЛА ТЕХНІКИ БЕЗПЕКИ ПРИ РОБОТІ ЗА КОМП'ЮТЕРОМ .. | 67 |
| | ВИСНОВКИ | 69 |
| | СПИСОК ЛІТЕРАТУРИ | 71 |
| | ДОДАТОК А..... | 75 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

ІТО – оксид індію та олова.

ВАХ – вольт-амперна характеристика.

МН – машинне навчання

НМ – нейронні мережі

SVM – метод опорних векторів

KNN – метод k-найближчих сусідів

ANN – штучна нейронна мережа (Artificial Neural Networks)

ВСТУП

Технологія виявлення газів в сучасному світі володіє великою вагою та використовується в різних галузях, починаючи від екологічного моніторингу й закінчуючи промисловою безпекою. Уміння точно визначати та вимірювати гази виявляється ключовим для забезпечення громадського здоров'я, збереження екологічної стійкості та забезпечення безпеки на робочому місці. Останнім часом високотехнологічні матеріали та інноваційні підходи в галузі сенсорики призвели до значних досягнень у виявленні газів.

Роль технології виявлення газів визначається її впливом на екологічний стан робочого середовища та безпеку промислових об'єктів. Сучасні вимоги до якості повітря, а також стандарти щодо викидів газів роблять актуальним розвиток нових методів та засобів для точного і надійного виявлення газових забруднень.

Дослідження в області сенсорики газів та застосування програмного забезпечення для обробки та аналізу отриманих даних відкриває нові перспективи в галузі моніторингу робочого середовища та промислової безпеки. Розробка ефективних алгоритмів машинного навчання дозволить досягти вищого рівня автоматизації та точності виявлення газових компонентів, що має велике значення для подальших покращень у цій галузі.

Незважаючи на існуючі досягнення в галузі сенсорики газів, проблеми селективності та чутливості залишаються актуальними. Метою нашого дослідження є вдосконалення цих параметрів шляхом застосування методів машинного навчання.

У цій магістерській роботі я сподіваюся зробити внесок у розвиток галузі сенсорики газів та розширити можливості виявлення та аналізу різного роду газу за допомогою передових технологій машинного навчання.

1 ТЕХНОЛОГІЇ ГАЗОВИХ СЕНСОРІВ: ВІД ОСНОВНИХ ЕЛЕМЕНТІВ ДО ФРАКТАЛЬНО-ПЕРКОЛЯЦІЙНИХ НАНОСИСТЕМ НА ОСНОВІ ZnO/NiO

1.1 Технологія та класифікація сенсорів.

Сенсор – це звичайний пристрій, що перетворює вимірювану величину в вихідний сигнал для передачі та використання в системах керування. Він є ключовим елементом технічних систем для вимірювання, сигналізації, регулювання та управління процесами. Сенсори здатні перетворювати різні параметри (температуру, тиск, концентрацію, напругу тощо) у зручний для обробки сигнал.

Розробка сенсорів є завданням, що вимагає співпраці фахівців з різних галузей, таких як хімія, фізика, біологія, інженерія та екологія. Основні вимоги до сучасних сенсорів включають селективність, високочутливість, надійність, стабільність та точність.

Сфера застосування сенсорів дуже широка і охоплює медицину, екологію, безпеку, промисловість, військово-промисловий комплекс та побутову техніку. Використання сенсорів необхідне для контролю летких продуктів горіння, якості харчових продуктів, діагностики захворювань, моніторингу робочого середовища на наявність токсичних газів та дослідження клітин живих організмів.

Походження історії хімічних сенсорів можна відслідкувати до закінчення XIX та початку XX століття. У цей період виникли перші пристрої, які можна розглядати як прототипи сучасних хімічних сенсорів. Наприклад, у 1880 році був створений катарометр для визначення вмісту водню в водяній парі, а в 1888 році з'явилися металеві електроди Нернста, а в 1906 році - скляний електрод Кремера.

У той час термін "сенсор" визначав портативні пристрої для аналізу хімічного складу навколишнього середовища. Основна структура сенсора

включала чутливий елемент, а також невід'ємно важливий перетворювач, які взаємодіяли для отримання вихідного сигналу.

Класифікація сенсорів включає в себе різні параметри, такі як вид вихідних величин, характер вихідного сигналу, середовище передачі сигналів, кількість вхідних величин, технологія виготовлення та інші. Наприклад, сенсори можна поділити на активні і пасивні, дискретні, аналогові, цифрові, провідні та бездротові залежно від виду вихідних величин та характеру сигналу.

З розвитком технологій було створено багато нових типів сенсорів, таких як електрохімічні, потенціометричні, оптичні, напівпровідникові і багато інших [4]. Класифікація сенсорів враховує вимірювані параметри і принцип їхньої дії.

Сенсори можна класифікувати за природою процесів, бо принцип дії лежить в їх основі і ділиться на три основні типи: фізичні, хімічні та біологічні. Цей умовний поділ обумовлений тим, що у більшості випадків одночасно можуть відбуватися як фізичні, так і хімічні процеси. Деякі ключові типи сенсорів, які мають важливе застосування на практиці, включають ультразвукові сенсори, інфрачервоні датчики, п'єзоелектричні сенсори, магнітоелектричні датчики, біосенсори, а також хімічні та газові сенсори.

Кондуктометричні сенсори, також відомі як резистивні або хеморезистивні, мають просту структуру та базуються на тому, що наявність або відсутність певних хімічних елементів, що контактують з пристроєм, можуть змінювати електропровідність [5]. Основним принципом функціонування цих датчиків є зміна опору під впливом різних хімічних процесів, таких як адсорбція, хімічні реакції, дифузія або каталіз які відбуваються на поверхні або в об'ємі чутливого шару.

В свою чергу адсорбція - це процес, при якому молекули однієї речовини (адсорбат) міцно прикріплюються до поверхні іншої речовини (адсорбенту). У випадку напівпровідникових газових сенсорів адсорбатом є молекули газу, а адсорбентом - поверхня напівпровідника.

Процес адсорбції газів на поверхню напівпровідника відбувається у кілька стадій:

1. Дифузія. Молекула газу дифундує з газової фази до поверхні напівпровідника.

2. Адгезія. Молекула газу прилипає до поверхні напівпровідника завдяки різниці потенціалів між ними.

3. Абсорбція. Молекула газу поглинається поверхнею напівпровідника, що призводить до зміни її стану.

Існує два основних типи адсорбції газів на поверхні напівпровідника. Розглянемо їх нижче.

Фізична адсорбція. Даний вид адсорбції може відбуватися за рахунок слабких сил Ван-дер-Ваальса. Від 0,01 до 0,1 еВ саме стільки становить енергія, що вивільняється при фізичній адсорбції молекули газу на поверхні напівпровідника.

Хімічна адсорбція. Цей тип адсорбції відбувається за рахунок ковалентних зв'язків. Від 1 до 10 еВ саме стільки становить енергія, що вивільняється при хімічній адсорбції молекули газу на поверхні напівпровідника.

Важливо зауважити що адсорбція газів на поверхню напівпровідника має деякі наслідки і може призвести до зміни електрофізичних властивостей. В свою чергу ці зміни можуть бути використані для визначення концентрації газу в навколишньому середовищі.

Важливим фактором, що визначає чутливість напівпровідникового газового сенсора до певного газу, є тип адсорбції, що відбувається. Фізична адсорбція зазвичай призводить до незначних змін електрофізичних властивостей напівпровідника, тому сенсори, в яких відбувається фізична адсорбція, мають низьку чутливість. Хімічна адсорбція, з іншого боку, зазвичай призводить до таких самих змін але більш істотних тому сенсори, в яких відбувається хімічна адсорбція, мають більш високу чутливість.

Ось деякі приклади того, як адсорбція газів на поверхню напівпровідника може призвести до зміни його електрофізичних властивостей:

Зміна електропровідності напівпровідника, саме до цього може призвести адсорбція газів. Такий ефект може бути викликаний зміною кількості або рухливості носіїв заряду. (рис. 1.1).

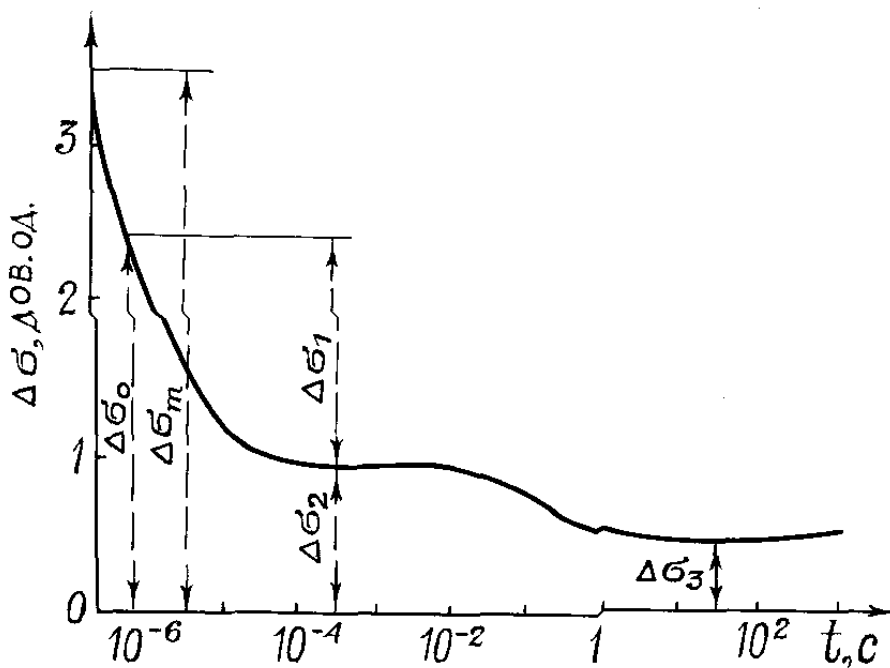


Рисунок 1.1 — Графік зміни поверхневої провідності напівпровідникового зразка.

Також можлива зміна потенціалу контакту. Це пояснюється зміною потенціалу контакту між напівпровідником і електродом. Це може бути пов'язано зі зміною розподілу зарядів на межі розділу між напівпровідником і електродом.

Зміна оптичного поглинання. Адсорбція газів може призвести до зміни оптичного поглинання напівпровідника. Це може бути використано для визначення концентрації газу за допомогою фотометричних методів.

Ультразвукові датчики працюють на основі ехолокації, визначаючи відстань до об'єкта за рахунок вимірювання часу між відправленням та отриманням звукового імпульсу. Ці сенсори застосовуються для виявлення різноманітних об'єктів та вимірювання відстаней.

Інфрачервоні датчики працюють у спектральній області інфрачервоного випромінювання і використовуються, наприклад, для виявлення руху та автоматичного управління освітленням.

П'єзоелектричні сенсори використовують п'єзоелектричний ефект у кристалах або кераміці для перетворення механічної енергії в електричну та навпаки.

Магнітоелектричні датчики реагують на зміни електричних властивостей під впливом магнітного поля і використовуються, наприклад, у системах навігації та безпеки.

Біосенсори включають в себе біологічний матеріал і використовуються для вивчення міжмолекулярних взаємодій та контролю процесів формування тонких плівок.

Газові сенсори реагують на присутність різних газів у навколишньому середовищі і використовуються для виявлення та моніторингу газів.

Зараз для визначення газів широко використовуються різні типи газових сенсорів, включаючи каталітичні, теплопровідні, електрохімічні, резистивні, механічні, оптичні, акустичні та інші. Кожен з цих типів сенсорів працює за своїм принципом, який базується на конкретних процесах, визначаючи їхню чутливість.

Наприклад, каталітичні сенсори вимірюють кількість виділеної теплоти під час каталітичної реакції окиснення газу, що аналізується. Теплопровідні сенсори використовують різну швидкість втрати тепла нагрітим чутливим елементом у середовищі різних газів з різною теплопровідністю. Електрохімічні сенсори реагують на газ, змінюючи свої електрохімічні властивості на електродах під час окисно-відновних реакцій.

Механічні сенсори використовують зміну об'єму металу при адсорбції газів, а акустичні сенсори детектують зміну частоти акустичних хвиль при адсорбції газів. Оптичні сенсори вимірюють зміну оптичних властивостей матеріалу через взаємодію з газом.

Зокрема, п'єзокварцеві, іон-провідні, оптичні та ємнісні газові сенсори набули широкого застосування. П'єзокварцеві сенсори використовують зміну частоти коливань п'єзоелектричного елемента при адсорбції газу [6]. Іон-провідні сенсори базуються на утворенні іонів у газочутливому шарі під час окисно-відновних або кислотно-основних перетворень аналізованого газу. Оптичні сенсори реєструють зміну світлопропускання чутливого шару при взаємодії з газом, і ємнісні сенсори визначають зміну електричної ємності чутливого шару при наявності певного газу.

Різні типи сенсорів мають свої сильні та слабкі сторони, тому їхні характеристики порівнюють за різними параметрами, щоб визначити, який тип сенсора є найбільш ефективним для конкретного застосування.

Сенсори мають ряд основних характеристик, визначаючи їхню ефективність. Ці характеристики включають чутливість, поріг чутливості, швидкодію (час відгуку і релаксації), стабільність, селективність, робочу температуру та номінальну статичну характеристику перетворення [7].

Чутливість сенсора визначається зміною величини вихідного сигналу (або електричного опору для резистивних сенсорів) відносно зміни концентрації аналізованого газу. Наприклад, для адсорбційно-напівпровідникового сенсора чутливість визначається його здатністю змінювати електричну провідність у присутності газу.

Поріг чутливості вказує на мінімальну концентрацію газу, на яку реагує сенсор. Швидкодія сенсора характеризується часом відгуку та релаксації, а величина відгуку вимірюється при зміні концентрації газу.

Селективність визначається здатністю сенсора реагувати лише на певний газ і не реагувати на інші. Стабільність показує, наскільки сталими залишаються параметри сенсора протягом тривалого часу експлуатації [8].

Робоча температура вказує на температуру чутливого шару сенсора під час вимірювань, і її зменшення може призвести до поліпшення енергоефективності сенсорів.

Номінальна статична характеристика перетворення описує залежність між значеннями величин сигналів сенсора у статичному режимі від концентрації газу.

Важливо враховувати, що поліпшення однієї характеристики може призвести до погіршення інших, і враховувати ці компроміси при виборі сенсора для конкретного застосування. Резистивні сенсори, зокрема, є перспективними, оскільки вони комбінують високі показники багатьох характеристик, хоча мають свої обмеження.

Ключовими аспектами, які вирішують питання щодо властивостей хімічних сенсорів, зокрема резистивних, є функція рецептора і передача інформації. Рецепторна функція сенсора залежить від окисно-відновних та кислотних властивостей поверхні, присутності специфічних промоторів реакцій, а також адсорбції кисню та води. Функція передачі інформації, у свою чергу, визначається висотою бар'єра Шоттки і залежить від рухливості носіїв струму, розміру зерен чутливого матеріалу та хімічних домішок [9].

Оскільки при аналізі повітряно-газових сумішей, зокрема в промислових зонах чи для виявлення витоків газів з різних ємностей, існує велика ймовірність присутності домішок, таких як CO, SO₂, H₂S, які можуть легко впливати на металеві резистори, більш обґрунтованим є використання адсорбційно-напівпровідникових сенсорів на основі оксидних матеріалів. Зазначений тип сенсорів змінює свої властивості під впливом робочого середовища. Принцип їхньої дії полягає в зміні електричного опору (провідності) при проходженні газОВО-повітряної суміші через їхню поверхню. Зміна опору відбувається через

хемосорбцію кисню на поверхні, викликану взаємодією цього кисню з аналізованими газами. Чутливість сенсора залежить від ступеня зміни опору в присутності аналізованого газу, що визначає його ефективність. Перші адсорбційно-напівпровідникові сенсори були розроблені в Японії, яка залишається лідером у виробництві таких сенсорів.

1.2 Газові сенсори на основі напівпровідникових оксидів.

У сучасний період розвитку водневої енергетики, транспорту та інших галузей виробництва актуальним є питання глобального забруднення. З цієї причини важливим стає контроль за вмістом токсичних і вибухонебезпечних речовин у навколишньому середовищі. Для досягнення цієї мети використовують газоаналітичні прилади, оснащені хімічними газовими сенсорами. Ці сенсори використовуються в різних галузях, включаючи промисловість, медицину та контроль якості продуктів [1].

Напівпровідникові оксиди, такі як SnO_2 , ZnO , NiO , TiO_2 , WO_3 та In_2O_3 , представляють значний інтерес для створення газових сенсорів. Електропровідність цих напівпровідників є дуже чутливою до змін на поверхні в діапазоні температур від 300 до 800 К [2]. Це обумовлено окисно-відновними реакціями на поверхні оксидів, що відбуваються в цьому температурному інтервалі. Поверхня оксидів має високі адсорбційні властивості через наявність вільних електронів у зоні провідності, поверхневих і об'ємних кисневих вакансій, а також активного хемосорбованого кисню.

Газові сенсори на основі напівпровідникових оксидів працюють за принципом зміни концентрації носіїв заряду на поверхні через хімічні реакції з газами. Останнім часом активно вивчаються нанорозмірні матеріали для поліпшення чутливості і динамічних характеристик сенсорів [3]. Це включає синтез та дослідження фізико-хімічних властивостей наноматеріалів, їхню морфологію та структуру.

Одним із важливих напрямків досліджень є вивчення механізму та кінетики реакцій на поверхні газочутливого шару сенсора для створення високочутливих газових сенсорів. При цьому важливо враховувати як фізико-хімічні властивості поверхні, так і характеристики нанорозмірних матеріалів.

Отримання детальної інформації про морфологію, структурно-адсорбційні властивості, фізико-хімічні характеристики та red-ox властивості поверхні є ключовим для цілеспрямованого синтезу сенсорних напівпровідникових матеріалів із заданими властивостями. Такий підхід є важливим для розвитку високочутливих газових сенсорів у сучасних технологіях.

1.3 Основні елементи та фізичні властивості газових сенсорів.

Газочутливий матеріал є вирішальним компонентом газових сенсорів, оскільки він визначає їхню чутливість та вибірковість. Розглянемо різні типи газочутливих матеріалів та їхні фізичні властивості.

1. Оксиди металів (NiO , SnO_2 , Fe_2O_3 , ZnO) мають високу чутливість до різних газів, що робить їх цінними матеріалами для створення газових сенсорів [10].

2. Напівпровідники, такі як кремній (Si) та германій (Ge), також проявляють газочутливість, змінюючи свою електропровідність при взаємодії з газами [11].

3. Полімери або полімерні газочутливі матеріали, які реагують на зміни масозмінності чи оптичних властивостей під впливом газів, особливо ефективні для виявлення летючих органічних сполук [12].

4. Наноматеріали, такі як нанотрубки вуглецю чи наночастинки металів, може підвищити чутливість газових сенсорів завдяки їхнім унікальним електронним та поверхневим властивостям [13].

Такі газочутливі матеріали можна використовувати для створення сенсорів з різною вибірковістю та чутливістю до різних газів, що робить їх ефективними у різноманітних застосуваннях.

У газових сенсорах, споживач потужності (часто називається нагрівачем) та трансдюсер - це ключові компоненти, які впливають на продуктивність сенсора. Давайте детальніше розглянемо ці елементи.

Споживач потужності або нагрівач є важливою частиною газових сенсорів, особливо тих, які використовують напівпровідникові оксиди металів (MOS) або інші матеріали, чутливі до температури. Основні функції нагрівача включають:

1. Оптимізація температури. Газочутливі матеріали часто вимагають конкретної робочої температури для виявлення максимальної чутливості та вибіркової. Нагрівач забезпечує досягнення і підтримання цієї оптимальної температури.

2. Підвищена Чутливість. Нагрівання газочутливого матеріалу дозволяє сенсору ефективніше реагувати на взаємодію з газами. Це особливо важливо для матеріалів, які виявляють температурно-залежну провідність, таких як напівпровідникові оксиди металів.

3. Мінімізація Дрейфу. Підтримання стабільної та контрольованої температури допомагає зменшити температурно-залежний дрейф, підвищуючи загальну точність та надійність газового сенсора.

4. Запуск. Нагрівач сприяє швидкому нагріванню сенсора під час запуску, дозволяючи йому швидко досягти бажаної робочої температури [14].

Трансдюсер відповідає за перетворення змін властивостей газочутливого матеріалу в електричний сигнал, який можна виміряти та інтерпретувати. Різні типи трансдюсерів використовуються в залежності від характеру газочутливого матеріалу та бажаного вихідного сигналу. Основні методи трансдукції включають:

1. Резистивні Трансдюсери. Вимірюють зміни електричного опору газочутливого матеріалу.

2. Конденсаторні Трансдюсери. Виявляють варіації ємності, викликані змінами діелектричних властивостей матеріалу.

3. Оптичні Трансдюсери. Використовують зміни оптичних властивостей, таких як поглиблення або люмінесценція, для газового виявлення.

4. Масо-Чутливі Трансдюсери. Реагують на зміни маси, часто властиві полімерам, внаслідок взаємодії з газами [15-16].

Якщо говорити про фізику газових сенсорів, то опір газочутливого матеріалу є ключовим аспектом функціонування газових сенсорів, зокрема тих, які використовують газочутливі матеріали, прикладом можуть бути оксиди металів. Розглянемо прості але основні аспекти фізики цього процесу.

Перше про що можна сказати говорячи про опір газочутливого матеріалу це зміна опору при взаємодії з газами.

Газочутливі матеріали, такі як оксиди металів, мають властивість зміни електричного опору при взаємодії з різними газами. Ця зміна опору може бути виміряна і використовується для виявлення та кількісного визначення концентрації газів у навколишньому середовищі.

Математичне представлення залежності опору:

Опір (позначений як R) газочутливого матеріалу залежить від концентрації газу (позначеної як C) та температури. Математично це може бути виражено рівнянням:

$$R = R_0 \cdot e^{(\beta \cdot C)} \quad (1.1)$$

де R_0 – опір відсутності газу, β – коефіцієнт, а C – концентрація газу [17].

Температурна залежність:

Температурна залежність є важливим фактором для багатьох газочутливих матеріалів. Зазвичай сенсори працюють при підвищених температурах для забезпечення необхідної чутливості. Температурна залежність може бути описана рівнянням:

$$R = R_0 \cdot e^{(\alpha \cdot (T - T_0))} \quad (1.2)$$

де α – коефіцієнт температурної залежності, T_0 - початкова температура [18].

Вибірковість та чутливість:

Властивості газового сенсора можуть бути налаштовані для виявлення конкретних газів. Вибірковість та чутливість сенсора визначаються хімічним складом газочутливого матеріалу.

Відновлення та стійкість:

Сенсор повинен бути здатний відновлювати свій вихідний стан після виявлення газу (відновлення). Стійкість вказує на тривалість, протягом якої сенсор може залишатися чутливим при тривалому виявленні газу.

1.4 Сенсори, які реагують на гази за допомогою ZnO.

Сенсори, які реагують на гази за допомогою ZnO (цинкового оксиду) є активно вивченою та широко використовуваною галуззю в газосенсорній технології. ZnO визначається своєю великою поверхнею, високою стабільністю, вигідною електрохімічною активністю та можливістю формування різноманітних морфологій, таких як наноструктури та наночастинки, що робить його ідеальним матеріалом для газових сенсорів. Такі газові сенсори мають кілька ключових аспектів та характеристик.

Чутливість до газів:

Оксид цинку (ZnO) привернув значну увагу в галузі газового зондування завдяки своїй надзвичайній чутливості до різних газів, включаючи чадний газ (CO), діоксид азоту (NO₂), сірководень (H₂S), аміак (NH₃) та деякі інші. Завдяки унікальним властивостям поверхні ZnO, він здатний утворювати міцні зв'язки з молекулами газу. Це підвищує чутливість сенсорів на основі ZnO до різних газів,

що робить їх цінним матеріалом для розробки газових сенсорів для різних застосувань. Численні дослідження висвітлюють газочутливі властивості ZnO, підкреслюючи його ефективність у виявленні низьких концентрацій цільових газів [19].

Селективність:

Однією з ключових переваг газових сенсорів на основі цинкового оксиду є їхня селективність, що може налаштовуватися. Завдяки структурним модифікаціям або включенню специфічних каталізаторів тепер можна успішно адаптувати сенсори ZnO для виявлення цільових газів з високою точністю [20].

Стабільність та відновлення:

Сенсори на основі ZnO зберігають свою чутливість і точність протягом тривалого часу, що робить їх надійними для застосування в системах безперервного моніторингу. Крім того, ці датчики можуть ефективно відновлюватися після виявлення газу [21].

Температурна залежність:

Експлуатаційні характеристики сенсорів, що використовують ZnO тісно пов'язані з підвищеними температурами. Високі робочі температури суттєво впливають на їхню чутливість та селективність [22].

Розглядаючи фізичні основи та механізм роботи ZnO-сенсорів можна виділити пару моментів які висвітлюються в літературі більше всього.

Механізм детектування газу:

Фундаментальний принцип, що лежить в основі роботи газочутливого сенсора на основі ZnO, полягає в адсорбції молекул газу на поверхні ZnO. Ця адсорбція викликає зміни електричних властивостей матеріалу, що призводить до зміни електричного струму, який проходить через датчик. Механізм включає складну взаємодію між поверхнею ZnO і молекулами газу, що призводить до вимірюваних змін, які є основою для виявлення газу [23].

Наноструктури та наночастинки:

Наноструктури на основі ZnO, такі як нанодропи або наночастинки, дозволяють значно підвищити ефективність газового зондування. Ці наноструктури пропонують значно більшу площу поверхні порівняно з об'ємними матеріалами, що сприяє підвищенню чутливості і покращенню поверхні відгуку [24].

Контроль температури:

Температура відіграє вирішальну роль в оптимальному функціонуванні сенсора, виготовленого з ZnO. Деякі сенсори працюють у нагрітому режимі, де підвищені температури покращують кінетику адсорбції та десорбції газу, сприяючи покращенню чутливості. З іншого боку, існують сенсори, призначені для роботи при кімнатній температурі, що забезпечує практичні переваги з точки зору енергоефективності та портативності пристрою [25].

Таким чином, механізм виявлення газу в сенсорах на основі ZnO ґрунтується на взаємодії між молекулами газу і поверхнею ZnO, а наноструктури і контроль температури відіграють ключову роль в оптимізації чутливості і загальної продуктивності сенсора.

1.5 Сенсорні властивості фрактально-перколяційної наносистеми на основі ZnO/NiO.

Фрактально-перколяційні наносистеми – це клас матеріалів, які характеризуються особливими структурними та електрофізичними властивостями, що сприяють їхньому застосуванню в різних областях, зокрема в сенсорних технологіях, ключові аспекти цих систем:

1. Фрактальна структура: Фрактал - це геометрична структура, яка виявляє той самий ступінь нерегулярності на всіх масштабах. У випадку фрактально-перколяційних наносистем це може означати те, що їхні структури мають складні та нерегулярні форми на нанометровому рівні.

2. Перколяційні властивості: Перколяція вказує на те, як провідні області (наприклад, наночастинки) розташовані у матеріалі. Фрактально-перколяційні наносистеми мають таку структуру, що сприяє створенню провідних шляхів через систему навіть при невеликій концентрації провідних елементів.

3. Сенсорні застосування: Ці наносистеми можуть бути використані в сенсорах, оскільки їхні провідні властивості можуть змінюватися при контакті з різними газами чи речовинами. Наприклад, зміна провідності може бути пов'язана з адсорбцією газу на поверхні наночастинок.

4. Велика поверхня: Фрактальна структура може мати велику поверхню у порівнянні з об'ємом, що робить їх особливо чутливими до взаємодії з оточуючим середовищем.

5. Електрофізичні властивості: Електрофізичні характеристики таких систем можуть бути досить унікальними, що використовується для створення чутливих до газів чи інших зовнішніх впливів датчиків.

На основі існуючих досліджень та поглибленого вивчення, можна дізнатися як створювалися унікальні фрактально-перколяційні наносистеми ZnO/NiO. Детально розглянути кожний етап формування цих наноструктур, включаючи методи синтезу та використані матеріали, які визначають їхні унікальні властивості.

Створення фрактально-перколяційної наносистеми ZnO/NiO, це доволі не квапливий процес. Спочатку створюється контактний майданчик, для цього використовувалася техніка відкладання ІТО на лабораторних скляних підложках перед нанесенням ZnO/NiO наносистем. Контакти були стійкими до окиснення та мали омичний характер щодо ZnO. Відстань між площадками становила 1.5 мм.

Контактні майданчики відіграють критичну роль у забезпеченні ефективної взаємодії наносистем з зовнішнім середовищем та датчиковими системами. Вони становлять точку з'єднання між наноструктурами та зовнішнім

електричним чи сенсорним колом, визначаючи сприйнятливість та реакцію системи на зовнішні впливи.

Для створення фрактально-перколяційних наносистем на основі ZnO/NiO використовувалася технологія з трьома етапами. На першому етапі отримували цинкові наноструктури у вигляді тривимірних мереж за умов наближених до рівноважних в чистому аргоні. Діаметр окремих цинкових нанодротів варіювався від 50 нм до 140 нм (рис. 1.2).

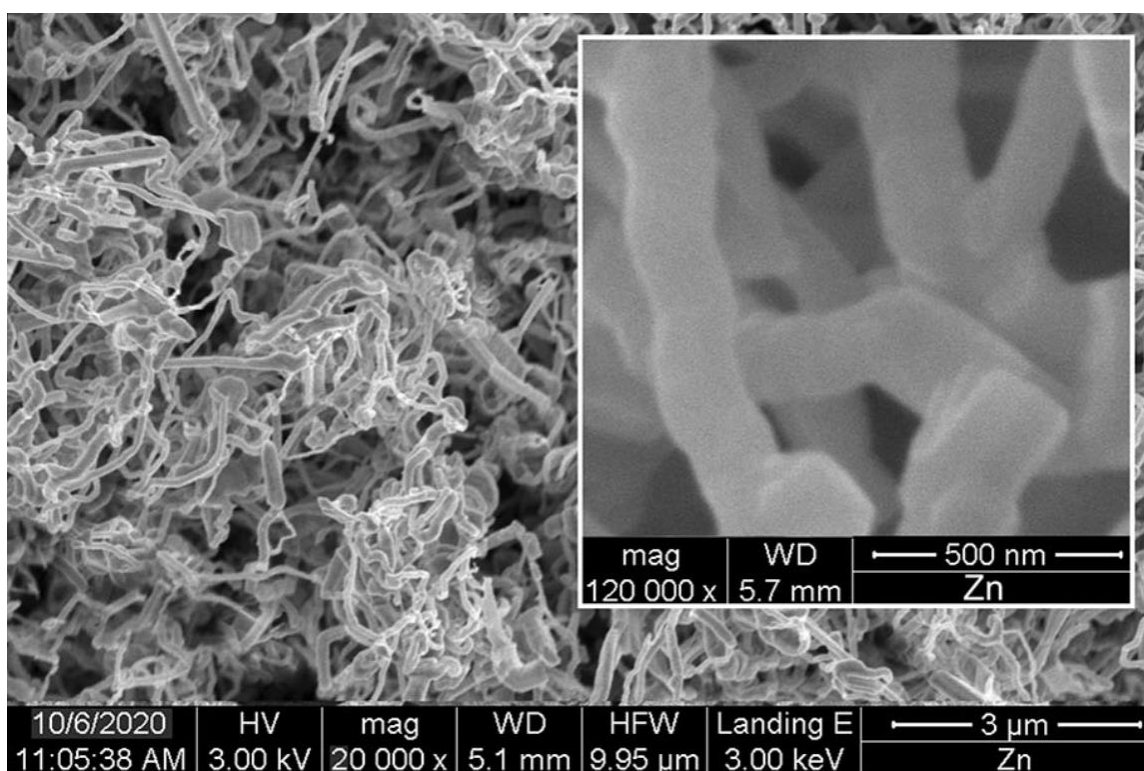


Рисунок 1.2 — Перший етап формування фрактально-перколяційної наносистеми

На другому етапі цинкові наносистеми піддавали окисненню протягом 3 годин в атмосфері повітря при 350 °С. Під час цього процесу формувалися ZnO нанодроти з чіткою полікристалічною структурою (рис. 1.3). Одночасно зникав значний відсоток нанодротів із найменшими діаметрами. Діаметр окремих ZnO нанопроводів варіювався від 80 нм до 100 нм.

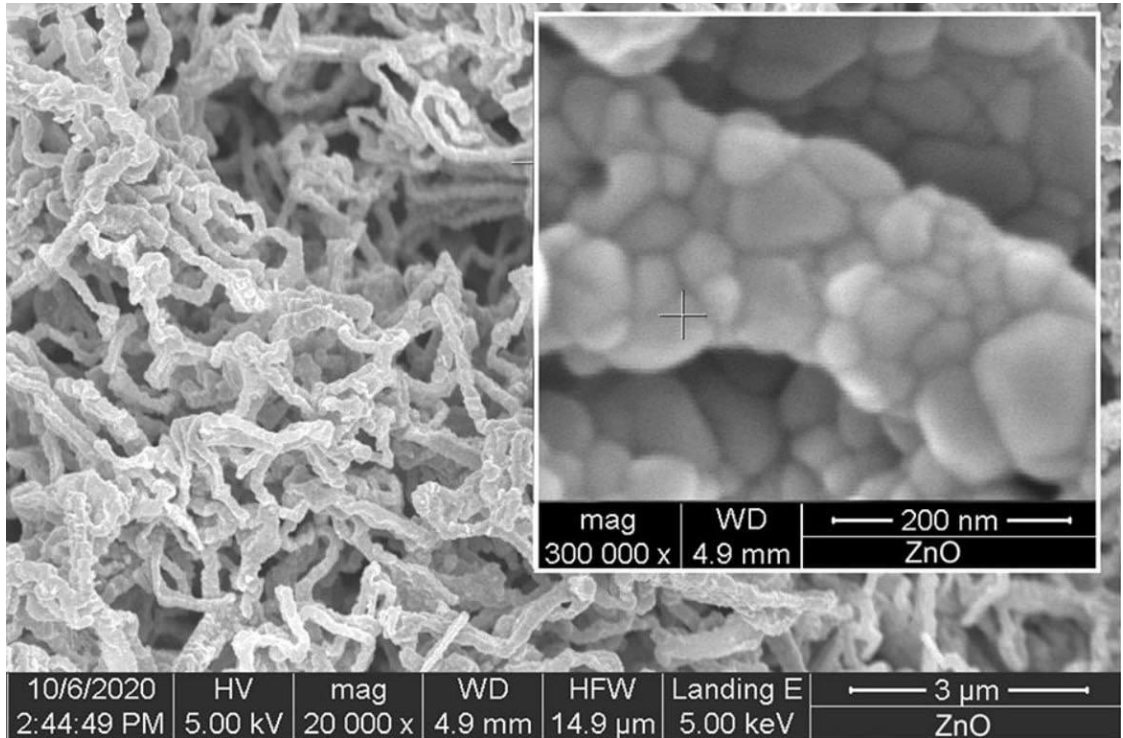


Рисунок 1.3 — Полікристалічна структура після окиснення

На третьому і заключному етапі плівки NiO конденсувалися на поверхні ZnO при температурі 200 °C за допомогою реактивного магнетронного розпилення нікелю в суміші аргону та кисню. Дослідження показали, що структура отриманих конденсатів належить полікристалічному NiO з тригональною симетрією просторової групи (рис. 1.4).

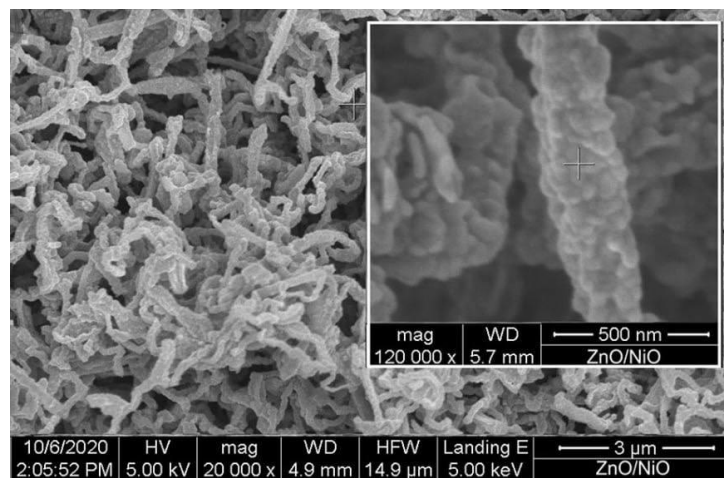


Рисунок 1.4 — Конденсування плівки NiO на поверхню ZnO

Також були проведені рентгенівські та енерго-дисперсійні дослідження результати яких засвідчують присутність лише однієї гексагональної структури (ГЦК), яка відповідає параметрам решітки ZnO. Важливо відзначити, що дифракційні піки, що свідчать про наявність NiO, не були виявлені в результатах. Це підтверджує чистоту та однорідність отриманих наносистем а також елементний аналіз підтверджує високі концентрації цинку, нікелю та кисню у наносистемі.

Сканування електронного мікроскопу (SEM) продемонструвало рівномірні морфологічні характеристики композиту по всій його товщині. А отримані результати підтверджують відсутність значущих дефектів або нерівностей у структурі. Енергодисперсійний аналіз підтверджує однорідний розподіл складових елементів - цинку, кисню та нікелю - по всій площині зразка.

Ця методологія виявляється ключовою для досягнення визначальних структурних та морфологічних характеристик фрактально-перколяційних наносистем ZnO/NiO. Рентгенівські та енергодисперсійні методи є потужними інструментами для аналізу складу і структури матеріалів на атомному рівні. Вони дозволяють отримувати матеріали з високою чистотою та однорідністю, які є необхідними для чутливих сенсорів. [26].

2 МАШИННЕ НАВЧАННЯ У АНАЛІЗІ ДАНИХ ГАЗОВИХ СЕНСОРІВ

2.1 Основні принципи машинного навчання.

Машинне навчання є ключовою галуззю сучасних інформаційних технологій, вона стрімко розвивається та впроваджується в різноманітні сфери нашого життя. Також машинне навчання стрімко розвивається в сенсорних технологіях що відкриває широкі перспективи для розробки інноваційних рішень у вигляді "розумних" систем та пристроїв, здатних навчатися та адаптуватися до змін оточення.

Сама галузь машинного навчання – це по суті спрощена аналогія навчання людини, в ній використовується набір прикладів, спостережень та реакцій на них. Головна мета полягає в створенні моделей, які як найкраще відображають наявні дані і мають змогу надавати достовірні прогнози. Саме ці прогнози можуть стосуватися різних питань, таких як опізнання об'єктів на зображенні, прогнозування тенденцій на ринку або аналіз відгуків на різні товари в мережі Інтернет [27].

Машинне навчання або МН може застосовуватися в різних галузях, що стало можливим завдяки розвитку Інтернету, постійному збільшенню кількості даних та доступності потужних обчислювальних ресурсів. Ця галузь існує вже понад 65 років, вона починала як об'єкт досліджень різних науковців, а тепер вона повністю відкрита для вирішення завдань будь-якою людиною, яка може ознайомитися з логікою роботи алгоритмів та їх реалізаціями.

МН відрізняється від звичайного навчання тим, що ми не просто даємо комп'ютеру інформацію, а й навчаємо його, як її обробляти. Зазвичай навчання ділиться на два типи це індуктивне та дедуктивне.

Індуктивне навчання - це процес, коли комп'ютер робить висновки про загальні закономірності, спостерігаючи за конкретними прикладами. Ці

висновки потім використовуються для побудови моделей, які пояснюють причини явищ. Також моделі періодично перевіряються, деякі з них "виживають" і використовуються, покращуючись з часом, тоді як інші можуть бути відкинуті в процесі навчання.

Дедуктивне навчання - це процес, коли комп'ютер використовує загальні правила, щоб робити висновки про конкретні приклади. Це схоже на звичайне навчання людини, коли її вчать за готовими формулами і потім вона застосовує їх на практичному досвіді.

МН є індуктивним процесом, оскільки ми навчаємо машину, спостерігаючи за реальними прикладами і роблячи на їх основі висновки. Ці висновки потім використовуються для побудови моделей, які можна використовувати для вирішення нових завдань.

Розробка моделі МН включає кілька етапів, і починається вона з підготовки або обробки даних. Цей етап є головним для досягнення високої якості роботи моделі, оскільки ефективність алгоритму напряму залежить від правильного представлення даних. У машинному навчанні для представлення даних часто використовується ознакове описання об'єкту. Це означає, що кожний об'єкт описується набором чисел, які називаються ознаками. Наприклад, якщо взяти соціальне опитування то кожна людина може бути описана рядом ознак, таких як вік, стать, рівень доходу тощо. Вибір ключових ознак є важливим кроком у машинному навчанні, оскільки він визначає, наскільки добре алгоритм зможе виконувати свою задачу.

Іншим методом представлення даних є "сирі" дані, вони можуть бути використані, при обробці зображень. Такі дані відрізняються тим, що кожен об'єкт може мати ознаки в дуже великій кількості, представлених, наприклад, кольором кожного пікселя на зображенні.

Другим етапом є визначення типу задачі, яку ми хочемо вирішити. В машинному навчанні існують різні типи задач, такі як класифікація, кластеризація, регресія та інші. Вибір методу залежить від виду задачі.

Третій та четвертий етапи включають в себе вибір параметрів та підготовку тренувальної вибірки. Дані потрібно розділити на тренувальну частину та тестову, в свою чергу модель навчається на тренувальній частині даних. Далі, модель тестується на тестовій вибірці щоб оцінити її ефективності.

Цей процес дозволяє створити доволі ефективну модель МН, яка може показувати доволі грані результати і використана в інших задачах.

МН поділяється на три загальні типи:

1. Навчання під контролем або з вчителем, де є набір даних і вибірка з вірним розподілом на класи [28].

2. Навчання без вчителя або без контролю, де система знаходить закономірності в великому наборі даних [29].

3. Навчання на основі проб і помилок - це тип машинного навчання, при якому агент навчається шляхом пробування різних дій і спостереження за наслідками цих дій. Агент отримує зворотний зв'язок у вигляді винагороди або покарання, який використовується для коригування його поведінки.

Задачі МН можуть включати класифікацію, регресію, кластеризацію, зниження розмірності та інші, і їх можна застосовувати в різних сферах, для розпізнавання об'єктів, для прогнозування фінансових питань, тощо. МН починає показувати дуже гарні результати і стає більш ефективнішим у багатьох випадках порівняно з традиційними методами.

Після визначення типу задачі наступним етапом є конструювання алгоритму, саме алгоритм буде використовуватися для вирішення обраної задачі МН.

Далі, настає етап тренування алгоритму, коли він навчається на тренувальних даних. Після завершення тренування модель перевіряється на тестових даних для оцінки її загальної ефективності [30].

Якщо модель на тестових даних продемонструвала результати які задовольняють її, то вона вже готова до застосування на практиці. Після тестування вона може застосовуватися до нових даних, для яких правильні

відповіді ще не відомі, і робити передбачення чи класифікації з урахуванням тренувальних даних.

Також важлива частина процесу є постійна валідація та налаштування моделі. Сама валідація дозволяє перевіряти, як добре модель може генерувати свої знання на нових даних, а налаштування це оптимізація параметрів моделі для покращення її продуктивності.

В МН є аспект який потрібно враховувати, а саме перенавчання. Якість відповідей на тестовій вибірці залежить від гарної обробки даних, після перенавчання якість може значно падати.

В той же час тренувальна вибірка це звичайний набір прикладів, які надаються системі для вивчення закономірностей. Якщо модель успішно вивчає цю закономірність, то вона може ефективно прогнозувати відповіді на тестовій вибірці.

Якісна модель може навчитися загальній закономірності даних, а не просто запам'ятати конкретні приклади. Погана модель може запам'ятати конкретні приклади, але вона не зможе навчитися загальній закономірності.

Мета регресійної задачі - це побудувати криву, яка найкраще відображає закономірності в даних. Якісна модель зможе побудувати криву, яка добре прогнозує нові дані. Однак, якщо даних для навчання недостатньо, модель може побудувати криву, яка не відображає реальну закономірність. Це явище називається перенавчанням.

На перенавчання можуть впливати такі фактори, як різноманітність та кількість даних, а також архітектура моделі.

У МН успіх гарного навчання залежить від розміру, різноманітності і якості тренувальних даних та відповідних параметрів для опису кожного об'єкта цих даних.

Мета системи - це навчитися на прикладах, щоб знаходити закономірності в даних і використовувати ці закономірності для прогнозування результатів на нових даних. Якість цих прогнозів залежить від вибору та опису ознак.

Існує кілька способів опису:

1. Бінарні ознаки можуть відповідати на питання "так" або "ні" і представлятися бінарним вектором.
2. Номінальні ознаки, це можуть бути координати точок, вони мають значення від нуля до плюс нескінченності.
3. Порядкові ознаки вказують на певне положення, як у задачах типу ранжування.
4. Кількісні ознаки використовуються для опису конкретних кількісних властивостей, таких як дохід, тощо.

Нейронні мережі або НМ, як один із типів МН, отримують на вході певний вектор ознак та генерують вектор відповідей на виході. НМ може деформувати багатовимірний простір ознак, що у висновку приклади наближаються до очікуваних зон на виході.

Розмірність простору ознак часто відрізняється від розмірності простору відповідей. Функціонал якості грає ключову роль у виборі правильного напрямку навчання системи. Він представляє собою певну функцію, яка виражає рівень помилок в системі, і під час навчання ми можемо мінімізувати цю помилку.

Також важливо відмітити що функціонал якості залежить від конкретної задачі. Наприклад, якщо брати до уваги задачу регресії то використовується функція так званої квадратичної помилки, де різниця між очікуваним і отриманим результатом підноситься до квадрата. Для задач класифікації можна використовувати різні функції помилки, наприклад, бінарна класифікація, де система відносить об'єкт до правильного класу або неправильного.

У МН є процес мінімізації помилки, в ньому ми намагаємося зменшити помилку моделі до низького рівня, який проявляється не лише на тренувальній, але й на тестовій вибірці, щоб після цього цю систему можна було вважати ефективною.

Один з прикладів процесу мінімізації помилки виникає в контексті нейронних мереж. Нейронні мережі мають змогу адаптувати вхідні вектори так, щоб на виході отримувати правильний вектор необхідного розміру.

Лінійна регресія є популярним методом у МН, широко використовуваним у різних галузях, включаючи економіку. Вона дозволяє розв'язувати типові задачі, такі як прогнозування відповідей на основі набору спостережень.

Є загальна формула лінійної регресії на площині, вона представляється як $y = kx + b$, де k визначає кут нахилу, а b - зміщення відносно центру координат. Задача полягає в знаходженні значень k і b , які мінімізують відстань до кожної точки.

Для ефективного знаходження параметрів із лінійної регресії це метод градієнтного спуску. Він дозволяє знаходити глобальний мінімум на поверхні, що представляє собою карту відстаней від кожної можливої прямої.

В свою чергу логістична регресія застосовується у випадках, коли вихідні відповіді є бінарними тобто це може бути 1 або 0. Наприклад, можемо прогнозувати ймовірність певної події, такої як покупка товару що в даному випадку це 1 - так або 0 – ні, на основі доходу певної особи. Логістична регресія знаходить параметри так, щоб відокремити два класи точок у просторі, використовуючи сигмоїдальну функцію. Ця функція надає гладку перехідну область між класами, що дозволяє оцінювати ймовірність належності кожної точки до певного класу.

Таким чином, глибоке розуміння основних принципів машинного навчання є основою для використання цих методів у різних технологіях.

2.2 Машинне навчання для аналізу даних у газових сенсорах.

Машинне навчання (МН) є потужним інструментом для вилучення значущої інформації зі складних даних, що генеруються газовими датчиками. Наприклад, дані можуть бути різного спектру так само як і сам аналіз цих даних.

Доволі цікавим прикладом даних є вольт-амперні характеристики, і ці дані можна обробляти за допомогою МН.

Газові датчики часто створюють складні ВАХ, що відображають реакцію датчика на різні концентрації газів. Методи машинного навчання чудово розшифровують ці складні закономірності, дозволяючи отримувати детальну інформацію про наявність та концентрацію газу. Лінійне розділення на основі статистичної теорії є цікавим підходом до розпізнавання газів за їхніми електричними характеристиками. Два основних методи, які використовуються в цьому підході - це K -найближчих сусідів (KNN) і підтримка векторних машин (SVM).

Алгоритми машинного навчання, які навчаються на прикладах з мітками, такі як метод опорних векторів (SVM) і нейронні мережі, можуть використовуватися для класифікації вольт-амперних характеристик на різні типи газів [31].

SVM є моделлю класифікації, яка шукає гіперплощину в просторі вибірки для розділення представників різних класів. SVM показує високу ефективність при малих вибірках та нелінійних залежностях в даних. Зазвичай гази мають складні характеристики, і SVM може бути використаний для класифікації, враховуючи нелінійність. Деякі дослідження вказують на успішне використання SVM для класифікації газів та управління дрейфом сенсорів [35].

KNN - це метод, що дозволяє класифікувати дані на основі їхньої близькості до інших даних. Для газових сенсорів це означає використання вольт-амперних характеристик для класифікації газів, змішаних газів, оскільки він простий у використанні. Дослідження вказують на високу точність визначення виду газу, також враховуючи деякі особливості, такі як дрейфт [36].

Для оцінки концентрації можна використовувати регресію:

Якщо цікавлять точні рівні концентрації, можна застосувати регресійні алгоритми, такі як лінійна регресія або регресія випадкового лісу (Random Forest). Навчаючись на відомих значеннях концентрації і відповідних вольт-

амперних даних, ці моделі можуть прогнозувати концентрацію газу за новими показаннями датчиків [32].

Ще одним способом вирішення таких задач є застосування Artificial Neural Networks (ANN) або простіше це нейронні мережі. Важливі аспекти включають основні принципи роботи нейронних мереж, їхню структуру, вплив кількості шарів, та порівняння алгоритмів, таких як Backpropagation (BP) і Extreme Learning Machine (ELM).

ANN моделює паралельну систему нейронів, намагаючись імітувати роботу людського мозку. Використовується модель McCulloch–Pitts для пояснення роботи нейронів у ANN, де кожен нейрон отримує сигнали з попереднього шару, ваги зв'язків враховуються, порівнюється з пороговим значенням і генерує вихід за допомогою функції активації, такої як сигмоїдальна функція. В НМ присутня багатошарова структура, Нейронна мережа (ANN) складається з трьох шарів нейронів: вхідного, прихованого та вихідного. Нейрони кожного шару мають свої функції та активаційні функції. Під час навчання нейронна мережа коригує зв'язки між нейронами та пороги кожного функціонального нейрона відповідно до навчальних даних.

Можна застосовувати два алгоритми навчання у ANN:

Backpropagation (BP): Алгоритм, що коригує ваги та пороги через зворотній зв'язок помилки, проте може вимагати тривалого часу навчання.

Extreme Learning Machine (ELM): Швидкий алгоритм навчання нейронної мережі, який пропонується як альтернатива BP.

Порівняння різних алгоритмів, таких як BP, SVM, та ELM, вказує, що кожен з них проявляє хороші результати в своєму власному напрямку та може бути комбінований для вирішення складних проблем (Таблиця 2.1) [37].

Таблиця 2.1 Порівняння алгоритмів

| | Ключовий момент | Переваги | Недоліки | Підсумок |
|-----|---|--|---|--|
| KNN | <ul style="list-style-type: none"> • k-значення • Дані про типи змішаних газів | <ul style="list-style-type: none"> • Зрозумілість • Нечутливість до шуму • Низька вірогідність перенавчання • Гарна комбінація з іншими алгоритмами | <ul style="list-style-type: none"> • Чутливість до розподілу вибірки • Повільна швидкість розпізнавання • Високий рівень просторової складності • Велике обчислювальне навантаження • Погана інтерпретованість | <ul style="list-style-type: none"> • Підвищення селективності до газів • Ідентифікація подібних газів |
| SVM | <ul style="list-style-type: none"> • Функція ядра • Дані про кількість змішаних газів | <ul style="list-style-type: none"> • Велика теоретична база • Обробка малої вибірки • Хороша здатність до узагальнення • Рішення нелінійних питань • Пошук оптимального рішення | <ul style="list-style-type: none"> • Чутливість до шумів • Складний вибір функції ядра • Тривалий час виконання • Погане застосування у великих вибірках | <ul style="list-style-type: none"> • Підвищення точності датчика • Дані зразків малих газів • Калібрування датчиків |
| ANN | <ul style="list-style-type: none"> • Вага • Функція активації • Кількість прихованих шарів | <ul style="list-style-type: none"> • Хороша здібність до навчання • Хороші можливості паралельної обробки • Сильна сумісність на випадок помилок • Вирішує складні нелінійні питання | <ul style="list-style-type: none"> • Вимога до великої кількості параметрів • Погана інтерпретованість результатів • Довгий час навчання • Легко переобладнати | <ul style="list-style-type: none"> • Обробка нелінійних залежностей • Прогнозування взаємодії газів • Калібрування датчиків |

Електронні носи (e-noses) – це сенсорні системи, які можуть виявляти та ідентифікувати різні хімічні речовини в повітрі або інших середовищах. Вони використовуються в різних сферах діяльності, таких як промисловість, медицина та охорона навколишнього середовища.

Електронні носи складаються з наступних основних компонентів:

1. Сенсори, які перетворюють хімічні сигнали в електричні сигнали.
2. Аналітичне обладнання, яке обробляє електричні сигнали та визначає присутність та концентрацію хімічних речовин.
3. Система управління, яка контролює роботу електронного носа.

Електронні носи можуть використовувати різні датчики для виявлення запахів. Наприклад, вони можуть використовувати метал-оксидні резистори, поляризаційно-чутливі транзистори або оптичні сенсори. Аналітичне обладнання може використовувати методи штучного інтелекту, такі як машинне навчання та нейронні мережі, для обробки даних.

МН може бути використано для підвищення ефективності електронних носів. Наприклад, машинне навчання можна використовувати для навчання електронного носа розпізнавати нові хімічні речовини або для покращення точності визначення концентрацій хімічних речовин.

Останнім часом у світі відбувається значний розвиток інтелектуальних газових датчиків та електронних носів (е-носів) з використанням машинного навчання. Машинне навчання в основному впливає на два основні фактори сенсорного пристрою: компенсацію дрейфу та селективність. Вчені та інженери розробили низку методів для боротьби з похибкою дрейфу в машинному навчанні. В даний час машинне навчання використовується як для компенсації дрейфу, так і для селективності. МН включає в себе обробку даних з датчика, зменшення розмірності, а потім навчання системи або мережі для прогнозування. Обробка даних має на меті вилучення надійної інформації про особливості з динамічного відгуку датчика, яка може представляти унікальні шаблони для певного газу, щоб забезпечити ефективність подальшого алгоритму розпізнавання. Існує низка методів обробки сигналів і даних, таких як стаціонарні (difference, relative difference, fractional difference, normalization, logarithmic difference, тощо), перехідні (інтегральні та диференціальні), та частотні (швидке перетворення Фур'є (FFT), неперервне або дискретне вейвлет-

перетворення (C/DWT)) моделі. Серед них системи та моделі, побудовані на перехідній системі та методах обробки даних виділення ознак у частотній області, демонструють максимальну точність.

По суті, поєднання методів машинного навчання з даними газових датчиків, особливо в контексті вольт-амперних характеристик і не тільки, відкриває нові можливості для точної ідентифікації газу, оцінки концентрації та виявлення аномалій. Згадана наукова література забезпечує міцну основу для вивчення різноманітних застосувань машинного навчання в аналізі даних газових датчиків.

3 РОЗРОБКА ПРОГРАМНОГО ІНСТРУМЕНТУ

3.1 Вибір та обґрунтування мови програмування (Python).

Вибір мови програмування для розробки системи машинного навчання для аналізу даних в моєму випадку вольт-амперних характеристик газових сенсорів зазвичай базується на кількох ключових факторах. Нижче я навів певні обґрунтування вибору мови Python для моєї задачі:

Python є популярною мовою програмування для машинного навчання, завдяки своїй потужній екосистемі бібліотек. Серед них такі популярні бібліотеки, як NumPy, Pandas, Scikit-learn, TensorFlow, і PyTorch. Ці бібліотеки дозволяють легко використовувати потужні інструменти для обробки даних, створення моделей машинного навчання та їхньої оптимізації.

NumPy надає масиви та математичні операції для ефективної обробки даних. Pandas забезпечує потужний інструмент для роботи з табличними даними, зокрема зчитування та запис різноманітних форматів. Scikit-learn пропонує широкий вибір алгоритмів машинного навчання та інструменти для підготовки даних та оцінки моделей.

TensorFlow і PyTorch - це два популярні фреймворки для глибокого навчання, які дозволяють розробникам створювати та навчати нейронні мережі для різних завдань. TensorFlow є більш популярним вибором для великих проектів, оскільки має велику спільноту та ресурси. PyTorch є більш гнучким вибором для дрібних проектів, оскільки використовує динамічний граф обчислень та має простий інтерфейс.

Простота читання та розуміння коду: Python відомий своєю простотою та лаконічністю синтаксису. Це робить код зрозумілим та легко читаним, що є важливим для розробки, тестування та удосконалення алгоритмів машинного навчання.

Зручність у використанні для розробників: Python активно використовується у галузі науки про дані та машинного навчання, що означає, що велика кількість розробників вже знайомі з цією мовою. Це спрощує співпрацю, обмін досвідом і кодом у галузі.

Відкритий код та активна спільнота: Python є відкритою мовою програмування з активною спільнотою розробників. Це забезпечує швидку підтримку, вирішення проблем та доступ до новітніх розробок у галузі машинного навчання.

Інтеграція з іншими технологіями: Python легко інтегрується з іншими технологіями, що може бути важливим, наприклад, для обробки даних в реальному часі чи взаємодії з іншими частинами системи.

Таким чином, обрання Python для розробки машинного навчання для аналізу ВАХ газових сенсорів забезпечить зручність у роботі, швидкий розвиток та високий рівень підтримки в галузі машинного навчання.

3.2 Алгоритми обробки та аналіз вольт-амперних характеристик.

Як вже проговорювалося дослідження включає в себе вимірювання вольт-амперних характеристик в атмосфері повітря з різними концентраціями метану та метанолу. Струм вимірювався при зміні напруги від 0 В до 5 В та зменшенні до нуля зі швидкістю сканування 13.5 мВ/с. Вольт-амперні криві мають асиметричну та гістерезисну поведінку з вираженими максимумами та мінімумами. Крім того, спостерігається виражена гістерезисна поведінка та максимумами, що пояснюються наявністю додаткових заряджаючих струмів конденсаторних систем із різними характеристиками RC (рис. 3.1).

Відзначається, що вольт-амперні криві, виміряні в повітрі з метанолом або метаном, мають суттєво різний характер. Тренд до майже лінійної поведінки з менш вираженим гістерезисом, який спостерігається для метану, пояснюється

тим, що адсорбований кисень в основному замінений великою кількістю метану, що призводить до мінімізації мікроконденсаторів у системі.

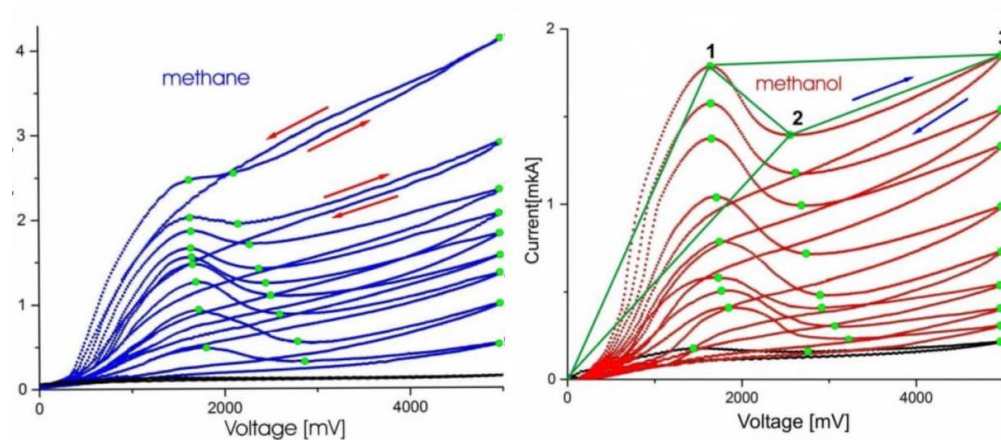


Рисунок 3.1 — ВАХ метану та метанолу.

В експериментальній середовищі було виміряно різні дані для метану та метанолу.

Три різних значення струму (I_1 , I_2 , I_3), виміряного при збільшенні та зменшенні напруги від 0 V до 5 V. Ці дані характеризують електричну відповідь системи при різних режимах напруги.

Три відповідні значення напруги (U_1 , U_2 , U_3), при яких вимірювався струм. Це напруги, при яких відбуваються вимірювання для вольт-амперних характеристик.

Концентрація газів (метану та метанолу) в атмосфері під час проведення вимірювань. Це визначає вплив конкретного газу на характеристики наносистеми.

Також відношення струму до напруги для кожного з значень (I_1/U_1 , I_2/U_2 , I_3/U_3). Ці величини можуть вказувати на кондуктивні властивості системи при різних умовах.

А також показники, які визначають зміну струму відносно зміни напруги між двома послідовними точками $((I_2-I_1)/(U_2-U_1), (I_3-I_2)/(U_3-U_2), (I_3-I_1)/(U_3-U_1))$. Ці величини можуть вказувати на характеристики системи, пов'язані із змінами вольт-амперних характеристик.

Також варто відзначити що при вимірюванні цих значень для розробки моделі машинного навчання потрібний доволі великий датасет для гарного навчання і отримати оригінальні значення у великій кількості доволі проблематично. Тому, було використано метод сплайн інтерполяції. Після вимірювання певної кількості експериментальних даних за допомогою цього методу можна створити синтетичні дані які будуть відповідати експериментальним даним тим самим даючи можливість створювати достатньо великі датасети для навчання моделі МН.

Сплайн інтерполяція - це метод математичної апроксимації, що використовується для створення гладкої кривої (сплайну) на основі набору експериментальних даних, коли точки не знаходяться на регулярній сітці. Цей метод особливо корисний тоді, коли нам потрібно заповнити проміжки між існуючими точками або вирівняти шум.

Основні ідеї сплайн інтерполяції:

Сплайн: Це кусочно-поліноміальна функція, яка розділяє набір точок на сегменти і використовує поліноми низького ступеня для апроксимації даних в кожному сегменті.

Гладкість: Сплайн інтерполяція прагне забезпечити гладкість кривої, уникати різких змін або "ривків", що можуть бути характерними для великодopusтимих поліноміальних апроксимацій.

Граничні умови: Сплайн може мати різні граничні умови, такі як фіксовані значення перших та останніх похідних. Це дозволяє керувати поведінкою сплайну на кінцях.

Процес сплайн інтерполяції включає в себе вибір типу сплайну (зазвичай кубічний), розбиття набору даних на сегменти, апроксимацію кожного сегмента

поліномом, і обчислення коефіцієнтів полінома так, щоб гладкість та неперервність були задовільні.

У моєму випадку, сплайн інтерполяція використовувалася для створення синтетичних додаткових даних для метану та метанолу, коли експериментальні дані були обмеженими.

Після того як датасет для метану та метанолу був заготовлений потрібно розробити алгоритм для обробки даних.

Моя мета полягає в розробці ефективної моделі класифікації, яка здатна автоматично розпізнавати ці два гази на основі їх вольт-амперних характеристик. А також при розпізнаванні газу мати змогу розпізнати концентрацію газу.

Спочатку я завантажую та обробляю дані, які стосуються газів. Важливим кроком є вибір ключових стовпців, які мають найбільший вплив на розпізнавання метану та метанолу. Потім я стандартизую ці дані за допомогою методу масштабування, зокрема `MinMaxScaler`, щоб підготувати їх для подальшого аналізу.

Розділяю дані на навчальну вибірку та тестову для ефективного оцінювання продуктивності моделей. Було прийнято рішення про поділ вибірок у співвідношенні 60% та 40%. Вибравши різні моделі класифікації, такі як логістичну регресію, наївний баєсівський класифікатор та дерево рішень, навчаю їх на навчальних даних.

Далі слідує етап оцінки продуктивності, де я використовую різні метрики для визначення точності та надійності кожної моделі. Крім того, будує криві навчання для вивчення того, як моделі пристосовуються до росту обсягу даних.

Завершую аналіз ROC-AUC для бінарної класифікації. Через цей комплексний підхід я очікую отримати ефективну модель класифікації, яка буде успішно визначати наявність метану та метанолу на основі газовмісних характеристик. Отримані результати можуть стати важливим внеском у розробку систем моніторингу газових речовин у промисловості та забезпечити базу для майбутніх досліджень.

Моя наступна фаза дослідження спрямована на розробку моделі, яка зможе не лише розпізнавати наявність метану та метанолу, але й точно передбачати їхню концентрацію на основі вольт-амперних характеристик. Це важливо для промислових застосувань, де точність вимірювань газових концентрацій має вирішальне значення.

Для реалізації цієї мети я розпочинаю з завантаження та обробки газовмісних даних, вибираючи ключові функції, які найкраще впливають на концентрацію газу. Далі, я використовую метод масштабування, наприклад, `MinMaxScaler`, для стандартизації функцій та підготовки їх до подальшого аналізу.

Після цього дані розділяються на навчальні та тестові набори для ефективного тренування та оцінювання моделі. Вибравши моделі регресії, такі як лінійна регресія, поліноміальна регресія чи регресія на основі методу опорних векторів (SVR), я навчаю їх на навчальних даних.

Далі слідує етап оцінки продуктивності, де я використовую різні метрики, такі як середня квадратична помилка (MSE) та коефіцієнт детермінації (R^2), щоб визначити, наскільки точно модель передбачає концентрацію газу.

Очікується, що розроблена модель буде точно передбачати концентрацію газу на основі вольт-амперних характеристик.

Також для повноти бачення результативності машинного навчання, було прийнято рішення розглянути метод машинного навчання без вчителя тобто коли система повинна сама опізнати різницю в даних і розділити їх на певні кластери, цей метод так і називається кластеризація.

У даній частині дослідження використовується метод k-середніх (KMeans) для кластеризації. Крім того, використовується метод t-SNE (t-distributed Stochastic Neighbor Embedding) для візуалізації результатів кластеризації в двовимірному просторі.

Також, важливо відмітити що експеримент було проведено виключно на експериментальних даних без урахування синтетичних. Що дозволили оцінити як працює модель на конкретних прикладах з невеликою кількістю даних.

Очікується що модель гарно розділить дані на два кластера та покаже гарний графічний результат на тестових даних.

4 МЕТОДОЛОГІЯ ТА РЕЗУЛЬТАТИ МН В АНАЛІЗІ ВАХ ВИСОКОПОРИСТИХ СИСТЕМ ZNO

4.1 Використання машинного навчання для розпізнавання газів та їх концентрації.

4.1.1 Набір даних для аналізу.

Для того щоб модель МН працювали повинен бути датасет даних. Датасет, що був обраний для подальшої роботи, містить в сумі 344 записи по 172 записи метану і метанолу і 12 колонок з наступним змістом:

I_1, I_2, I_3 (mkA) - три різних значення струму, що були виміряні при збільшенні та зменшенні напруги.

U_1, U_2, U_3 (mV) - три відповідні значення напруги.

Concentration (ppm) - концентрація газів.

$I_1/U_1, I_2/U_2, I_3/U_3$ - відношення струму до напруги.

$(I_2-I_1)/(U_2-U_1), (I_3-I_2)/(U_3-U_2), (I_3-I_1)/(U_3-U_1)$ - зміна струму відносно зміни напруги між двома послідовними точками.

Під час досліду всі колонки можна контролювати для дослідження правильності навчання моделі.

Також важливо зауважити що із набору даних експериментально виведених записів всього 9 для метану і 8 для метанолу.

4.1.2 Методи класифікації у задачі газового аналізу.

Щоб відрізнити метанол від метану, я використав метод машинного навчання, який називається логістична регресія. Логістична регресія – це метод, який можна використовувати для класифікації даних на два класи. Він працює, використовуючи логістичну функцію, яка перетворює лінійну комбінацію вхідних даних на ймовірність того, що дані належать до одного з двох класів.

Під час навчання модель адаптується до тренувальних даних, визначаючи оптимальні ваги для ознак та поріг рішення.

Після навчання модель може класифікувати нові дані та визначати ймовірність належності до кожного класу.

В моєму випадку для застосування логістичної регресії потрібні два класи даних, тобто метан або метанол. Я розробив в датасеті так званий label що показую відношення газу до одного з двох класів, для більш кращого навчання моделі деякі колонки (фічі) можна прибрати. Також для більшого розуміння я додав стовбець з назвою самого газу (рис. 4.1).

| | Concentration | I1 | U1 | I2 | U2 | I1/U1 | I2/U2 | I3/U3 | (I2-I1)/(U2-U1) | (I3-I2)/(U3-U2) | (I3-I1)/(U3-U1) | Gas | Label |
|----|---------------|--------|--------|--------|--------|---------|----------|----------|-----------------|-----------------|-----------------|---------|-------|
| 0 | 90 | 450.00 | 1800.0 | 250.00 | 2900.0 | 0.25000 | 0.086207 | 0.100000 | 0.181820 | 0.119050 | 0.015625 | Methane | 1 |
| 1 | 100 | 454.67 | 1799.8 | 253.02 | 2899.1 | 0.25262 | 0.087275 | 0.100940 | 0.183430 | 0.119790 | 0.015630 | Methane | 1 |
| 2 | 110 | 468.68 | 1799.4 | 262.10 | 2896.2 | 0.26046 | 0.090498 | 0.103750 | 0.188350 | 0.121990 | 0.015644 | Methane | 1 |
| 3 | 120 | 492.04 | 1798.6 | 277.22 | 2891.5 | 0.27357 | 0.095874 | 0.108440 | 0.196560 | 0.125670 | 0.015665 | Methane | 1 |
| 4 | 130 | 524.74 | 1797.5 | 298.39 | 2884.9 | 0.29193 | 0.103430 | 0.115000 | 0.208160 | 0.130780 | 0.015697 | Methane | 1 |
| 5 | 140 | 566.78 | 1796.1 | 325.61 | 2876.4 | 0.31556 | 0.113200 | 0.123440 | 0.223240 | 0.137310 | 0.015740 | Methane | 1 |
| 6 | 150 | 618.16 | 1794.4 | 358.89 | 2866.1 | 0.34449 | 0.125220 | 0.133760 | 0.241920 | 0.145220 | 0.015791 | Methane | 1 |
| 7 | 160 | 672.70 | 1792.5 | 394.44 | 2854.8 | 0.37529 | 0.138170 | 0.144690 | 0.261940 | 0.153380 | 0.015829 | Methane | 1 |
| 8 | 170 | 724.21 | 1790.3 | 428.51 | 2843.6 | 0.40452 | 0.150690 | 0.155010 | 0.280740 | 0.160700 | 0.015836 | Methane | 1 |
| 9 | 180 | 772.70 | 1788.0 | 461.11 | 2832.6 | 0.43216 | 0.162790 | 0.164690 | 0.298290 | 0.167190 | 0.015806 | Methane | 1 |
| 10 | 190 | 818.16 | 1785.5 | 492.22 | 2821.6 | 0.45822 | 0.174450 | 0.173760 | 0.314580 | 0.172860 | 0.015747 | Methane | 1 |
| 11 | 200 | 860.59 | 1782.8 | 521.85 | 2810.8 | 0.48272 | 0.185660 | 0.182190 | 0.329510 | 0.177740 | 0.015653 | Methane | 1 |
| 12 | 210 | 900.00 | 1780.0 | 550.00 | 2800.0 | 0.50562 | 0.196430 | 0.190000 | 0.343140 | 0.181820 | 0.015528 | Methane | 1 |
| 13 | 220 | 936.38 | 1777.0 | 576.67 | 2789.3 | 0.52694 | 0.206740 | 0.197180 | 0.355340 | 0.185120 | 0.015371 | Methane | 1 |

Рисунок 4.1 — Вивід списку даних датасетів.

Також окрім логістичної регресії були використані ще інші методи машинного навчання для класифікації та оцінки результатів та порівняння ефективності різних моделей МН. Крім логістичної регресії, я використовував:

1. Наївний Баєсівський класифікатор (Naive Bayes): Цей метод ґрунтується на теоремі Баєса та вважає всі ознаки незалежними. Я використовував GaussianNB, що припускає, що ознаки мають гаусовий розподіл.

2. Дерево рішень (Decision Tree): Цей метод розділяє дані на основі ряду умов і рішень. Мій код використовує DecisionTreeClassifier для створення моделі на основі дерева рішень.

Усі ці методи були використані для класифікації та оцінки результатів мого дослідження. Кожен з методів має свої особливості, переваги, застереження та обмеження.

Для того щоб модель добре працювала та видавала гарні результати потрібно щоб алгоритм обробив дані належним чином, для цього я використовую нормалізацію даних.

Нормалізація даних, є важливим етапом перед використанням багатьох алгоритмів машинного навчання. Нормалізація дозволяє стандартизувати значення ознак так, щоб вони мали однаковий масштаб, що може поліпшити збіжність алгоритмів та уникнути проблем, пов'язаних з великими різницями в значеннях ознак.

У моєму випадку я використовую MinMaxScaler з бібліотеки scikit-learn. Цей метод масштабує кожен ознаку до заданого діапазону (зазвичай від 0 до 1) (рис. 4.2), застосовуючи наступну формулу:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (4.1)$$

де X - оригінальне значення ознаки, X_{min} і X_{max} - мінімальне та максимальне значення ознаки в навчальному наборі відповідно.

| | I1 | U1 | ... | (I3-I2)/(U3-U2) | (I3-I1)/(U3-U1) |
|---|----------|----------|-----|-----------------|-----------------|
| 0 | 0.024390 | 0.781250 | ... | 0.172173 | 0.032671 |
| 1 | 0.026668 | 0.780625 | ... | 0.173656 | 0.032681 |
| 2 | 0.033502 | 0.779375 | ... | 0.178064 | 0.032710 |
| 3 | 0.044898 | 0.776875 | ... | 0.185438 | 0.032754 |
| 4 | 0.060849 | 0.773438 | ... | 0.195677 | 0.032821 |

Рисунок 4.2 — Вивід списку даних після нормалізації.

Це особливо корисно для методів, які використовують відстані між точками, таких як логістична регресія, метод опорних векторів або k -найближчих сусідів. Нормалізація дозволяє уникнути того, що одна ознака буде мати великий вплив через великі значення, порівняно з іншими ознаками.

Після нормалізації даних йде не менш важливий пункт в МН, це розподіл даних. Я розділив дані на тренувальний, валідаційний та тестовий набори за допомогою функції `train_test_split`. Це дозволило мені використовувати тренувальний набір для навчання моделі, валідаційний - для налаштування гіперпараметрів, а тестовий - для оцінки загальної продуктивності (рис. 4.3).

```
Розмір навчального набору: 206  
Розмір валідаційного набору: 69  
Розмір тестового набору: 69
```

Рисунок 4.3 — Розподіл даних на навчальний, валідаційний і тестовий набори.

Тільки після цих всіх маніпуляцій по обробці даних і їх нормалізації модель може навчатися та видавати нормальні результати.

В цьому дослідженні зокрема методів які я описав, ще використовував крос-валідацію для кожної з моделей, що дозволяє надійно оцінити їхню продуктивність шляхом розділення навчального набору на кілька піднаборів для тренування та перевірки. Кожен піднабір використовується як тренувальний та перевірочний набір з певною часткою даних, і цей процес повторюється кілька разів.

Це допомагає уникнути перенавчання та недонавчання, оскільки модель оцінюється на різних піднаборах даних. Такий підхід є ефективним способом визначити, наскільки добре модель може узагальнити свої навички на нових даних, а не просто "запам'ятати" тренувальний набір.

Крос-валідація допомагає отримати статистично достовірні оцінки продуктивності моделей та забезпечити більш об'єктивні результати, що може бути важливим у випадку обмежених обсягів даних.

4.1.3 Передбачення концентрації газу.

У другій частині мого дослідження метою було розробити модель для кількісного передбачення концентрації газу. Для цього я використовував той самий набір даних, який включав характеристики метану та метанолу.

В даному етапі роботи для передбачення концентрації газу працює вже зовсім інший алгоритм в якому не потрібно розділяти дані на дві умови або так звані label. В цьому випадку за label тепер виступає сама концентрація газу яка має багато різних умов.

Почавши з підготовки даних, я відокремив ознаки та цільові мітки, а потім провів масштабування функцій за допомогою методу Min-Max Scaling для забезпечення однакового масштабу всіх ознак.

Розділивши дані на тренувальний, валідаційний та тестовий набори, я використовував тренувальний для навчання моделі, валідаційний для налаштування гіперпараметрів та тестовий для оцінки кількісного передбачення концентрації.

Я використовував метод градієнтний бустінг (XGBoost) для передбачення. Оцінював результати за допомогою середньоквадратичної помилки та коефіцієнта детермінації.

XGBoost (eXtreme Gradient Boosting) є потужним алгоритмом градієнтного бустінгу, який часто виявляється ефективнішим у порівнянні з лінійною регресією, особливо в складних завданнях передбачення. Ось кілька причин, чому XGBoost може виявитися кращим:

1. Узагальнення до складних залежностей: XGBoost здатний моделювати складні нелінійні залежності між ознаками та цільовою змінною. Це особливо корисно, коли взаємодія між ознаками складна або коли існують нелінійні патерни в даних.

2. Регуляризація: XGBoost має вбудовану функцію регуляризації, яка допомагає уникнути перенавчання та поліпшує загальну здатність моделі.

3. Обробка втрат та оптимізація: Алгоритм градієнтного бустінгу дозволяє ефективно оптимізувати втрати та адаптувати модель для зменшення помилок на кожному етапі.

4. Використання ансамблю: XGBoost будує ансамбль дерев рішень, що дозволяє враховувати внесок кожного дерева у прогнозування. Це робить модель більш стійкою та точною.

Цей підхід дозволить мені оцінити точність передбачень концентрації газів та провести аналіз розподілу помилок.

4.1.4 Методи кластеризації у задачі газового аналізу.

В третій частині мого досліджу я намагаюся розділити дані на два кластера, цей метод повністю відрізняється від двох попередніх так як працює вже інший алгоритм так зване навчання без вчителя.

В цьому методі я використовував експериментальні дані без урахування штучно розроблених.

Перші етапи розробки подібні до тих методів які я вже використовував, тобто підготовка даних та нормалізація, після того як дані приведені до потрібного виду, починається обробка за допомогою KMeans.

KMeans - це метод, який використовується для поділу набору даних на кілька груп, або кластерів, на основі їх подібності. Алгоритм намагається розділити дані на задану кількість кластерів (K), при цьому мінімізуючи внутрішній розподіл кожного кластера. Кожен кластер представлений своїм центроїдом, а об'єкти в кластері відносяться до нього так, щоб мінімізувати суму квадратів відстаней між об'єктами і центроїдом. Результатом є групи об'єктів, які подібні між собою за деякими ознаками.

Також важливо урахувати що для методу KMeans, який використовується для кластеризації (unsupervised learning), розділення на тренувальний та тестовий

набори не є обов'язковим в тому сенсі, як це потрібно для задач supervised learning (навчання з учителем), де є мітки класів для навчання та валідації моделі.

В задачах кластеризації, як правило, немає заздалегідь визначених класів, і модель сама намагається згрупувати дані. Таким чином, валідація на тестовому наборі не обов'язкова з точки зору кластеризації. Тому для тестування моделі було прийнято рішення взяти певну кількість синтетичних даних із загального датасету для метану та метанолу, і вже з цими даними протестувати модель.

Також важливою складовою досліду показати як на графіку модель ділить дані і як тестові дані діляться на класи після машинного навчання. Для цього було використано метод зменшення розмірності t-SNE так як фіч для навчання певна велика кількість.

t-SNE - це метод зменшення розмірності для візуалізації високорозмірних даних у двовимірному (або тривимірному) просторі. Основна ідея полягає в тому, щоб зберегти відносини схожості між об'єктами під час переходу в менші розмірності. t-SNE приділяє більше уваги збереженню локальних структур у високорозмірному просторі, що робить його ефективним для виявлення груп об'єктів, які мають близькі схожості один з одним.

Всі ці маніпуляції дозволять мені якісно навчити розділяти дані та графічно оцінити точність передбачень газів.

4.2 Результати роботи програмного інструменту на різних газових сумішах.

4.2.1 Результати машинного навчання методами класифікації.

Після проробленої оптимізації даних можна приступати до оцінки і навчання моделі.

Метрики класифікації допомагають вимірювати ефективність моделі в задачах, де об'єкти потрібно призначити до певних класів. Accuracy вказує на загальну точність класифікації, визначаючи відсоток правильно класифікованих

екземплярів. Confusion Matrix надає деталізовану інформацію про правильні та неправильні класифікації для кожного класу.

Precision вимірює точність класифікації позитивних екземплярів, а Recall вказує на здатність моделі ідентифікувати всі можливі позитивні екземпляри. F1 Score є гармонічним середнім між Precision і Recall і корисний, коли ці метрики мають різний порядок величин.

Першою моделлю для оцінки ефективності є логістична регресія.

Результати для логістичної регресії вказують на високу ефективність моделі в класифікації. Точність складає приблизно 90%, що вказує на загально високу точність класифікації. Матриця плутанини вказує на кількість правильних та неправильних класифікацій для кожного класу. Модель правильно класифікувала 32 екземпляри класу 0 та 30 екземплярів класу 1. Є 2 помилкові класифікації класу 0 і 5 помилкових класифікацій класу 1.

Точність для позитивних екземплярів (Precision) становить 93.75%, що свідчить про точність у класифікації позитивних випадків. Повнота (Recall) складає 85.71%, що вказує на здатність моделі ідентифікувати більшість позитивних екземплярів. F1-показник досягає 89.55%, що є гармонійним середнім між точністю і повнотою (рис. 4.4).

Другою моделлю для оцінки ефективності є мережа Байєса або наївний баєсівський класифікатор.

Результати для баєсівського класифікатора свідчать про його помірну ефективність у порівнянні з логістичною регресією. Точність становить близько 79.71%, що вказує на загально задовільну точність моделі. Матриця плутанини показує, що є помилки як у визначенні позитивних, так і негативних екземплярів. Модель правильно класифікувала 30 екземпляри класу 0 та 25 екземплярів класу 1. Є 4 помилкові класифікації класу 0 і 10 помилкових класифікацій класу 1.

```

Logistic Regression Results:
Accuracy: 0.8985507246376812
Confusion Matrix:
[[32  2]
 [ 5 30]]
Precision: 0.9375
Recall: 0.8571428571428571
F1 Score: 0.8955223880597014

```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.86 | 0.94 | 0.90 | 34 |
| 1 | 0.94 | 0.86 | 0.90 | 35 |
| accuracy | | | 0.90 | 69 |
| macro avg | 0.90 | 0.90 | 0.90 | 69 |
| weighted avg | 0.90 | 0.90 | 0.90 | 69 |

Рисунок 4.4 — Звіт про класифікацію логістичної регресії.

Точність для позитивних екземплярів (Precision) дорівнює 86.21%, що свідчить про точність у класифікації позитивних випадків. Повнота (Recall) становить 71.43%, що означає, що модель не завжди ефективно ідентифікує всі позитивні екземпляри. F1-показник досягає 78.13%, що є середнім гармонічним між точністю і повнотою. В цілому, баєсівський класифікатор показує прийнятні результати з урахуванням того що даних для навчання доволі обмежені в кількості (рис. 4.5).

```

Naive Bayes Classifier Results:
Accuracy: 0.7971014492753623
Confusion Matrix: [[30  4]
 [10 25]]
Precision: 0.8620689655172413
Recall: 0.7142857142857143
F1 Score: 0.7812500000000001

```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.75 | 0.88 | 0.81 | 34 |
| 1 | 0.86 | 0.71 | 0.78 | 35 |
| accuracy | | | 0.80 | 69 |
| macro avg | 0.81 | 0.80 | 0.80 | 69 |
| weighted avg | 0.81 | 0.80 | 0.80 | 69 |

Рисунок 4.5 — Звіт про класифікацію мережі Байєса.

Третя модель для оцінки ефективності є Дерево рішень (Decision Tree).

Результати для Дерева рішень (Decision Tree) свідчать про його високу ефективність у класифікації. Точність моделі становить приблизно 98.55%, що вказує на високу точність класифікації загалом. Матриця плутанини показує лише декілька помилок у визначенні позитивних та негативних екземплярів. F1-показник досягає 98.55%, що є високим значенням і вказує на гармонічний баланс між точністю і повнотою. Узагальнюючи, Дерево рішень виявляється дуже ефективним у розв'язанні даної задачі класифікації (рис. 4.6).

```
Decision Tree Classifier Results:
Accuracy: 0.9855072463768116
Precision: 1.0
Confusion Matrix: [[30  4]
 [10 25]]
Recall: 0.9714285714285714
F1 Score: 0.9855072463768115
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.97 | 1.00 | 0.99 | 34 |
| 1 | 1.00 | 0.97 | 0.99 | 35 |
| accuracy | | | 0.99 | 69 |
| macro avg | 0.99 | 0.99 | 0.99 | 69 |
| weighted avg | 0.99 | 0.99 | 0.99 | 69 |

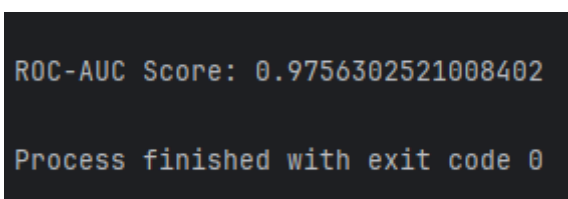
Рисунок 4.6 — Звіт про класифікацію дерева рішень (Decision Tree).

Також для фінального порівняння я ще зробив розрахунок ROC-AUC Score.

ROC-AUC Score (площа під кривою ROC) - це метрика, яка використовується для визначення якості бінарного класифікатора, оцінюючи його здатність розділяти класи. ROC (Receiver Operating Characteristic) - це графік, який показує залежність між чутливістю (True Positive Rate) та специфічністю (1 - False Positive Rate) для різних значень порогу класифікації.

ROC-AUC Score представляє собою площу під кривою ROC. Чим вище це значення, тим краще виконується класифікатор. Максимально можлива ROC-AUC Score дорівнює 1.0, що вказує на ідеальний класифікатор, а значення 0.5 вказує на випадковий вибір.

В контексті результатів, ROC-AUC Score дорівнює 0.9756, що показує дуже гарну ефективності моделі в розділенні класів. Високий ROC-AUC Score вказує на те, що класифікатор має високу чутливість при низькому рівні False Positive Rate, що є бажаним в класифікаційних задачах (рис. 4.7).



```
ROC-AUC Score: 0.9756302521008402
Process finished with exit code 0
```

Рисунок 4.7 — Звіт про результат ROC-AUC Score.

Також було розглянуто результати методу Cross-Validation. Загалом, результати крос валідації виглядають добре. Логістична регресія та дерево рішень мають високу точність, що вказує на те, що ці моделі можуть добре прогнозувати вихідні дані. Наївний Байєс має нижчу точність, але все одно є прийнятним (рис. 4.8).

Логістична регресія є відносно простим у використанні та інтерпретації. Наївний Байєс в свою чергу є швидким методом, що робить його хорошим вибором для задач з великими наборами даних. Дерево рішень може бути складним у використанні та інтерпретації, але цей метод може бути ефективним для задач з нелінійною залежністю між вхідними та вихідними даними.

Крос валідація є ефективним методом оцінки точності моделі. Вона розділяє набір даних на кілька частин і використовує кожну частину для тестування моделі, яка була навчена на інших частинах набору даних. Це

дозволяє оцінити точність моделі без необхідності використовувати тестовий набір даних, який може бути не репрезентативним для загального набору даних.

Крос валідація може бути трудомісткою, оскільки вимагає навчання моделі кілька разів, по одному разу для кожної частини набору даних. Також вона може бути неточною, оскільки залежить від того, як набір даних розділений на частини.

```
Logistic Regression Cross-Validation Accuracy: 0.7325581395348837
Naive Bayes Cross-Validation Accuracy: 0.7267441860465116
Decision Tree Cross-Validation Accuracy: 0.9127906976744187
```

Рисунок 4.8 — Звіт про результат методу Cross-Validation.

При аналізі моделей також було прийнято рішення скористатися learning curves для аналізу та візуалізації продуктивності моделей.

Графіки навчання (learning curves) - це візуальне відображення того, як змінюється продуктивність обраної моделі під час машинного навчання. Це корисний інструмент для визначення того, наскільки добре модель вчиться на тренувальних даних, та чи спостерігається перенавчання або недонавчання.

Якщо подивитися на результати логістичної регресії (рис. 4.9), можна сказати що модель може бути не сильно чутлива до перенавчання і тому може продовжувати покращувати свою продуктивність при додаванні нових даних. Тренувальний бал не дуже високий з невеликою кількістю зразків, але корегується і в подальшому зростає при збільшенні їхньої кількості. Валідаційний бал спочатку низький, але зростає з додатковими зразками. Те що тренувальний бал може збільшуватися також може і означати що модель має потенціал до перенавчання, і застосування цієї моделі для подальшого використання може бути проблематичним.

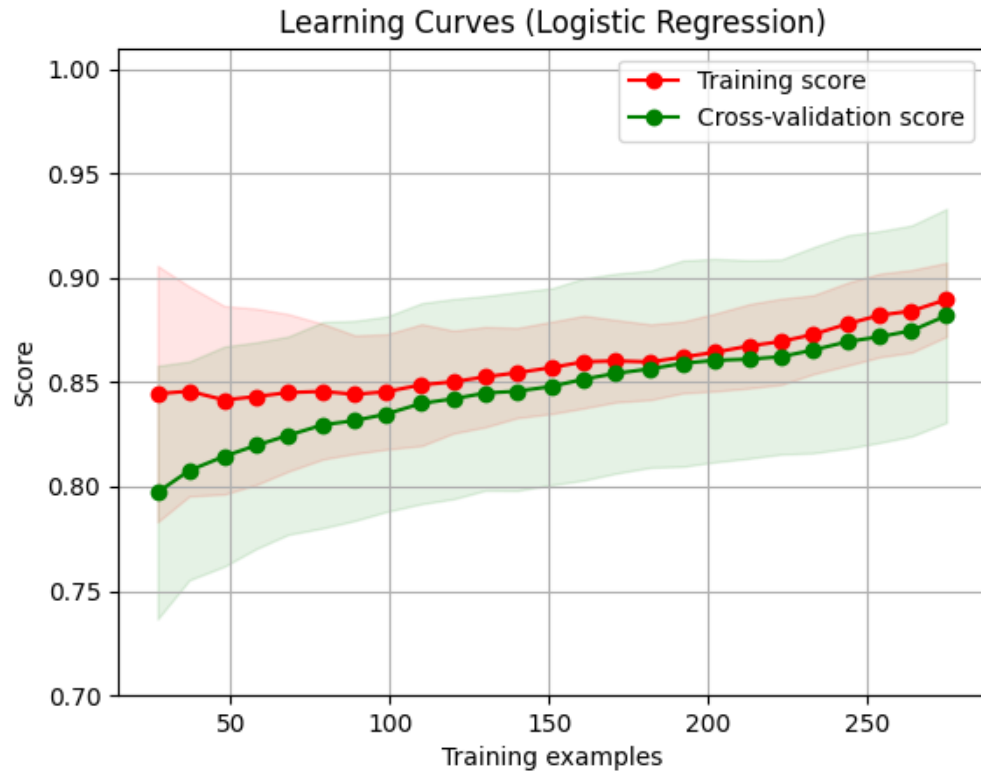


Рисунок 4.9 — Learning curves графік логістичної регресії.

Оцінюючи графік методу наївного Байеса можна сказати що, тренувальний бал високий з невеликою кількістю зразків, але зменшується при збільшенні їхньої кількості. Валідаційний бал спочатку дуже низький, але зростає з додатковими зразками. Модель починає з високого рівня варіативності (перенавчання) та високого рівня систематичної помилки (недонавчання), і обидва стабілізуються при використанні більшої кількості даних для тренування (рис. 4.10).

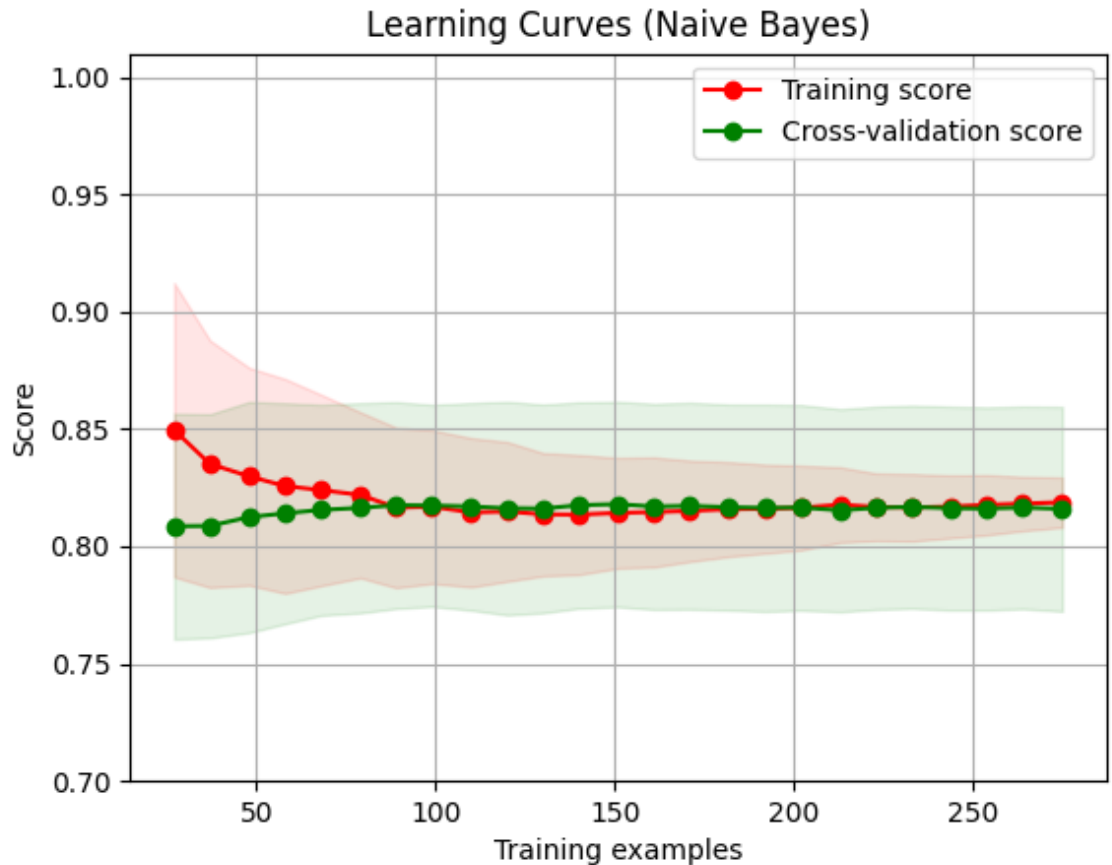


Рисунок 4.10 — Learning curves графік наївного Байєса.

Оцінюючи графік методу дерева рішень (Decision Tree) можна сказати що, тренувальний бал залишається високим незалежно від розміру тренувального набору. Валідаційний бал зростає з розміром тренувального набору, досягаючи плато. Модель має низький рівень варіативності (стійкий тренувальний бал), але високий рівень систематичної помилки спочатку. Зростання тестового балу свідчить про покращену генералізацію, але в кінці досягає плато, що вказує на зменшення вигоди від додаткових даних (рис. 4.11).

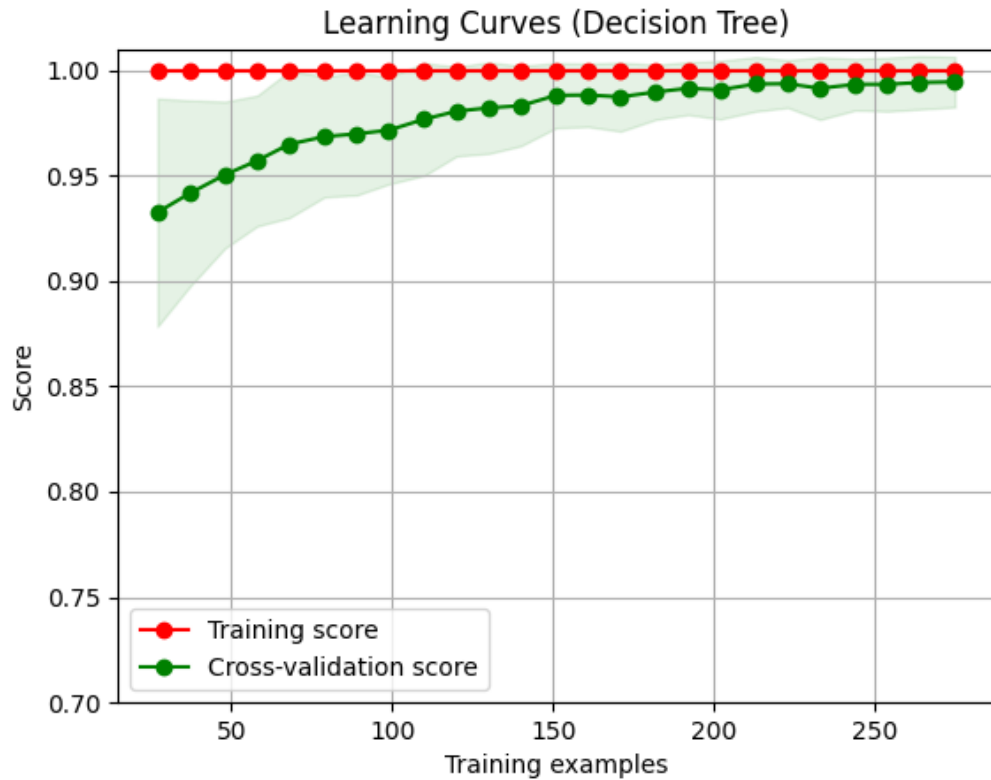


Рисунок 4.11 — Learning curves графік дерева рішень (Decision Tree).

Також після навчання моделі було побудовано графік по двом фічам із тестового набору які можуть показати на скільки гарно працює машинне навчання і як графічно система розпізнає дані метану і метанолу (рис. 4.12). На графіку зображено два класи даних метан який має позначку class 1 а також метанол який має позначку class 2. Як можна побачити модель гарно розрізняє дві фічі різних газів і будує це графічно.

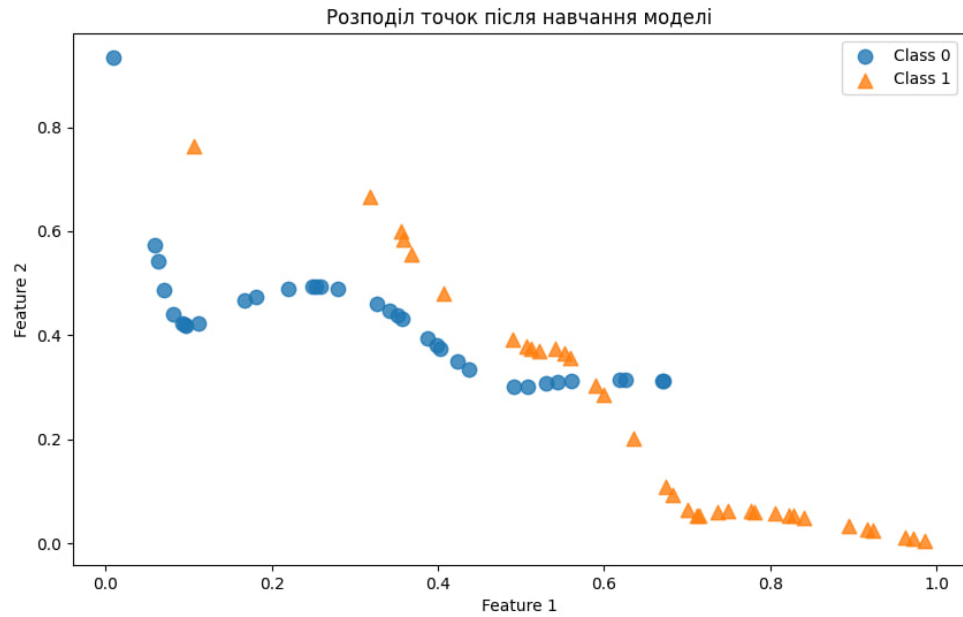


Рисунок 4.12 — Належність даних до двох класів (двох типів газу) у поданні в координатах двох ознак (I_1 , U_1) після навчання моделі.

У результаті проведеного аналізу класифікаційних моделей, використовуючи логістичну регресію, баєсівський класифікатор та дерево рішень, були отримані наступні висновки:

1. Логістична регресія:

- Досягнута точність (Accuracy) становить 89.86%, що свідчить про високий рівень правильних класифікацій.
- Модель виявилася добре себе показуючи як у питанні точності, так і у питанні поліпшення Precision та Recall, що свідчить про здатність визначати істинні позитивні та істинні негативні випадки.

2. Баєсівський класифікатор:

- Модель демонструє точність на рівні 79.71%, що вказує на менший рівень загальної ефективності порівняно з логістичною регресією.
- При цьому, баєсівський класифікатор має деяку непогану здатність визначати позитивні випадки (Precision) та виявляти істинно позитивні випадки (Recall), хоча менш ефективно, ніж логістична регресія.

3. Дерево рішень:

- Дерево рішень виявилось дуже ефективним класифікатором з точністю на рівні 98.55%.
- Висока точність, а також Precision, Recall і F1 Score свідчать про високу здатність моделі визначати обидва класи, зокрема істинно позитивні та істинно негативні випадки.

4. ROC-AUC Score:

- ROC-AUC Score для розглянутих моделей свідчить про високий рівень їх здатності розділяти класи. Особливо, для дерева рішень, де значення досягає 0.9756, що підтверджує високу чутливість і специфічність.

На підставі проведеного аналізу можна зробити висновок, що дерево рішень є найбільш ефективним методом класифікації для цього конкретного завдання. Воно показало високу точність і здатність правильно визначати обидва класи. Логістична регресія також виявилася доброю альтернативою, забезпечуючи високу точність та ефективність у розпізнаванні класів. Баєсівський класифікатор, хоча менш ефективний порівняно з іншими методами, все ж є варіантом, особливо у випадках, де важлива висока Recall.

4.2.2 Результати машинного навчання по передбаченню концентрації газу.

Передбачення концентрації газів засновано на методі класифікації газу. Тобто у нас вже є розподілення і потрібно передбачити концентрацію. Ця задача виступає в ролі багатокласової класифікації. В цьому методі я використовую модель XGBRegressor з бібліотеки XGBoost для передбачення концентрації газів на основі фіч (ознак).

Існує кілька ключових метрик для оцінки якості та точності моделі передбачення:

1. Середньоквадратична помилка (Mean Squared Error - MSE) - вимірює середнє значення квадратів відхилень між фактичними та передбаченими значеннями.

Велике MSE вказує на велику середньоквадратичну відхиленість моделі від реальних даних.

2. Коефіцієнт детермінації (R-squared) - вимірює відсоток варіації в залежній змінній, яку пояснює модель. Він знаходиться в діапазоні від 0 до 1, де 1 вказує на ідеальне узгодження.
3. Пояснена дисперсія (Explained Variance Score) - це метрика, яка вимірює відсоток варіації відгуку моделі, яку вона може пояснити. Іншими словами, ця метрика вказує на те, наскільки добре модель пояснює розкид даних навколо їхнього середнього значення. Значення в діапазоні від 0 до 1.

Модель XGBoost виявилася дуже точною, оскільки має низьке значення середньоквадратичної помилки (MSE) - 17.38, що вказує на малу відхиленість прогнозів від фактичних значень. Крім того, високий коефіцієнт детермінації (R-squared) - 0.998, свідчить про відмінну здатність моделі пояснювати варіацію в даних. Explained Variance Score також дуже близький до 1 (0.998), що вказує на те, що модель ефективно пояснює варіацію в цільовому показнику (рис. 4.13). Це підтверджує високу якість передбачень моделі та вказує на те, що вона добре узгоджується з реальними даними.

```
Mean Squared Error: 17.38097478919428  
R-squared: 0.9988807848699169  
Explained Variance: 0.9988853203994306
```

Рисунок 4.13 — Звіт роботи моделі XGBoost.

При аналізі було розроблено графік прогнозів проти спостережень (scatter plot) в даному контексті є інструментом візуалізації, на якому осі x відображаються фактичні значення (спостереження), а осі y відповідають відповідним прогнозам, зробленим моделлю (рис. 4.14).

У мене точки графіка, йдуть правильно по діагоналі з невеликими відхиленнями, свідчать про те, що прогнози моделі майже ідеально відповідають

фактичним значенням, також на графіку я виділив червоним кольором точки які входять до експериментально визначених даних а синім само собою дані які були розроблені штучно. Тому це вказує на високу точність та добру узгодженість моделі XGBoost із реальними даними.

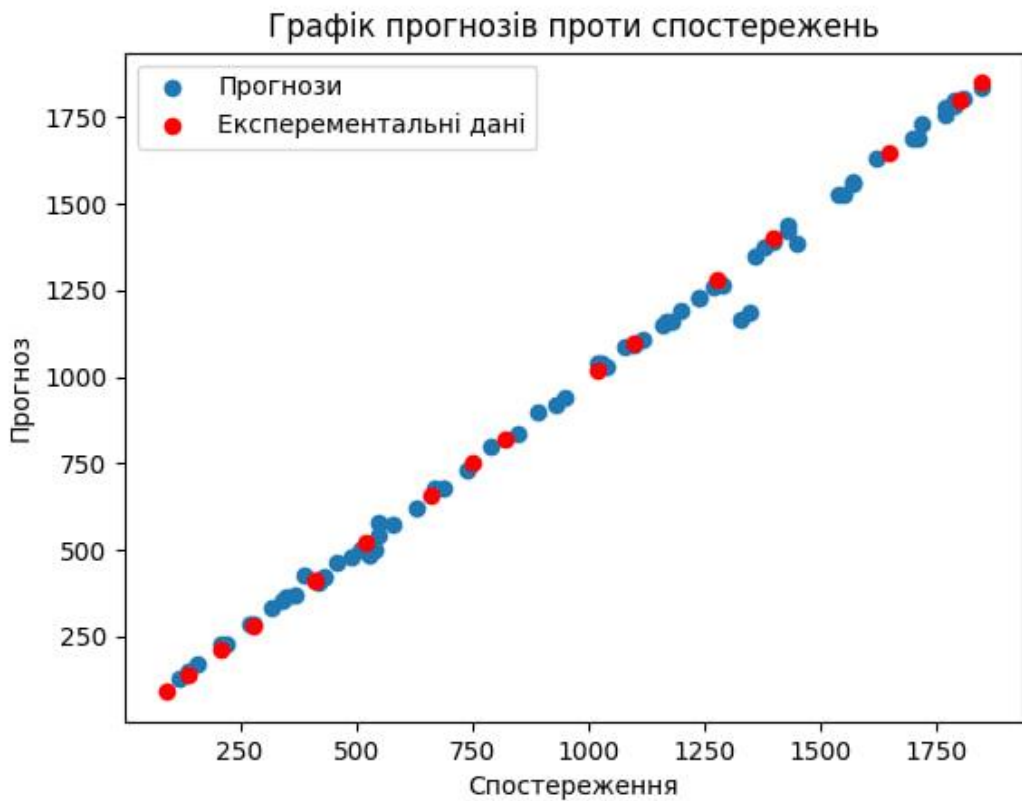


Рисунок 4.14 — Графік прогнозів проти спостережень (scatter plot).

На основі наданих результатів аналізу моделі прогнозування концентрації можна зробити наступні висновки. Експеримент показав, що модель XGBoost виявилася дуже ефективною у визначенні концентрації газу. З низьким значенням середньоквадратичної помилки (MSE) і високими коефіцієнтами детермінації (R-squared та Explained Variance), модель продемонструвала високу точність та здатність пояснювати варіацію вихідних даних. В свою чергу графік

прогнозів підтверджує високу узгодженість між фактичними і передбаченими значеннями.

4.2.3 Результати машинного навчання методом кластеризації.

Мій дослід включає в себе відтворення експериментальних даних для метану та метанолу, їх подальше об'єднання, аналіз та візуалізацію з використанням алгоритму кластеризації KMeans і методу зменшеної розмірності t-SNE. Кластеризація дозволила виділити два чітко визначені групи в наборі даних, що може вказувати на наявність двох різних типів спостережень. Після цього я використав цю модель для прогнозування кластерів для тестових даних, що підтвердило ефективність алгоритму на нових даних. Такий підхід дозволяє розуміти структуру даних, їх взаємозв'язки та потенційні закономірності, що є важливими аспектами для подальшого аналізу та використання в області класифікації та прогнозування.

На графіку (рис. 4.15) можна побачити роботу машинного навчання, результати я отримав за допомогою методів KMeans і t-SNE і вони дуже гарно відображають два кластери в нашому випадку це метан фіолетового кольору а також метанол жовтого кольору. Після отримання цих результатів потрібно протестувати модель на тестових даних та побачити як вона вже після навчання розрізняє кластери в тестовій вибірці. Було прийнято взяти по три точки кожного газу з суттєво різною концентрацією та оцінити результат. Як можна побачити на графіку та на звіті результатів (рис. 4.16) із шести точок система точно розділила їх на два кластери відповідного типу, та показала їх різним кольором на графіку. Сірі точки це тестові точки метану а червоні це тестові точки метанолу.

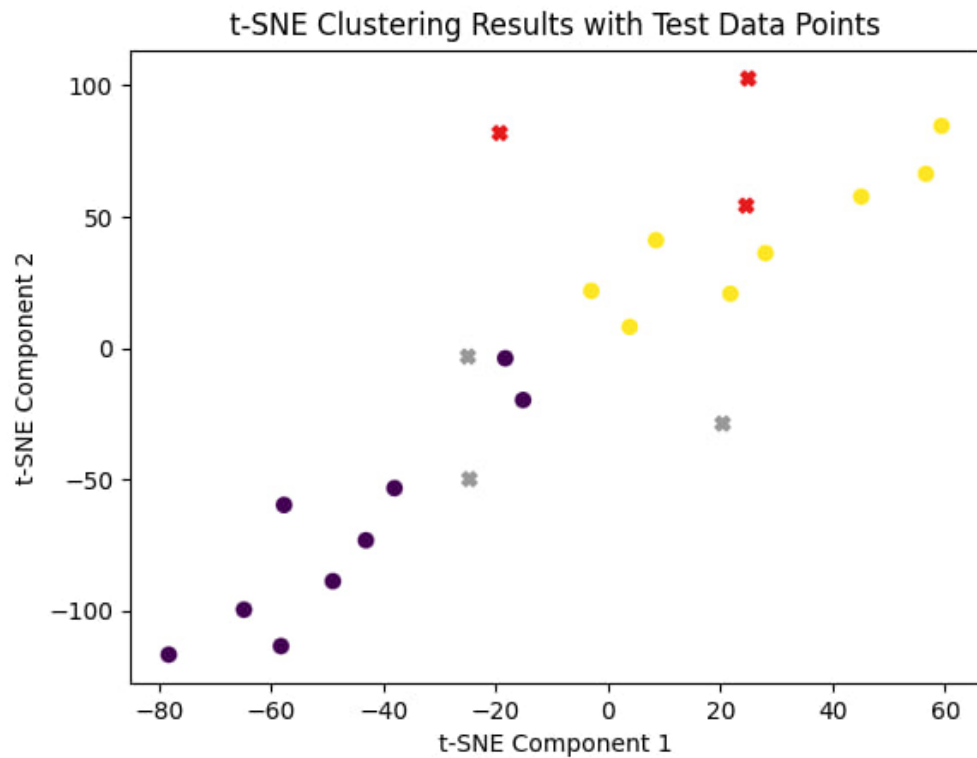


Рисунок 4.15 — Результати кластеризації даних газів за допомогою МН. Графік розділення на кластери для Метану (фіолетовий) та Метанолу (жовтий). Точкове представлення тестових даних: Сірі точки – тестові точки Метану, Червоні точки – тестові точки Метанолу.

```

Test Data:
Cluster
  0
  0
  0
  1
  1
  1
Cluster Summary:
Cluster
0    3
1    3
Name: count, dtype: int64

```

Рисунок 4.16 — Звіт тестових прогнозів розбиття метану та метанолу.

Можна зробити висновок, що кластеризація даних виявилася ефективною, відображаючи чітко виражені групи в експериментальних даних метану та метанолу. Також, модель успішно передбачила кластери для нових тестових даних.

5 ПРАВИЛА ТЕХНІКИ БЕЗПЕКИ ПРИ РОБОТІ ЗА КОМП'ЮТЕРОМ

Робота за комп'ютером є невід'ємною частиною сучасного життя. Комп'ютери використовуються вдома, на роботі, у школі, а також у багатьох інших сферах. Однак, незважаючи на всі переваги комп'ютерів, вони можуть бути небезпечними, якщо не дотримуватися правил техніки безпеки.

Правила техніки безпеки при роботі за комп'ютером можна розділити на кілька груп:

1. Правила безпеки для здоров'я
2. Правила безпеки для обладнання
3. Правила безпеки для інформації

Робота за комп'ютером може негативно впливати на здоров'я людини, зокрема на зір, поставу, нервову систему та психічне здоров'я. Тому важливо дотримуватися правил безпеки, щоб захистити себе від можливих проблем.

Робоче місце за комп'ютером має бути добре освітленим, але без прямих сонячних променів. Монітор повинен знаходитися на рівні очей або трохи нижче, а відстань між очима та екраном повинна становити не менше 50-70 см. Клавіатура та мишка повинні знаходитися в зручному положенні, щоб не потрібно було напружувати руки та спину.

Не варто працювати за комп'ютером безперервно протягом тривалого часу. Робіть перерви кожні 20-30 хвилин, щоб розім'ятися та відпочити очі. Рекомендується працювати за комп'ютером не більше 5-6 годин на день.

Здорове харчування та достатнє споживання рідини допоможуть підтримувати здоров'я та запобігти появі проблем зі здоров'ям, пов'язаних із роботою за комп'ютером.

Захисні окуляри допоможуть захистити очі від шкідливого випромінювання монітора.

Комп'ютерне обладнання є дорогим і чутливим до пошкоджень. Щоб захистити комп'ютер від поломок, важливо дотримуватися правил безпеки:

1. Не допускайте попадання вологи на комп'ютер.
2. Не кладіть на комп'ютер важкі предмети.
3. Не вимикайте комп'ютер, витягуючи шнур живлення з розетки.
4. Правильно експлуатуйте комп'ютерне обладнання.

Комп'ютерна інформація може бути цінною та конфіденційною. Щоб захистити інформацію від несанкціонованого доступу, важливо дотримуватися правил безпеки:

Використовуйте надійні паролі для захисту комп'ютера.

Не зберігайте важливу інформацію на відкритих дисках або флеш-накопичувачах.

Регулярно робіть резервні копії важливої інформації.

Додаткові правила безпеки:

1. Не залишайте комп'ютер без нагляду, якщо він ще працює.
2. Не їжте та не пийте за комп'ютером.
3. Не куріть за комп'ютером.
4. Не користуйтеся комп'ютером у темряві.
5. Не працюйте за комп'ютером, якщо ви втомлені або не почували себе добре.

Важливо пам'ятати, що правила техніки безпеки при роботі за комп'ютером розроблені для того, щоб захистити вас від можливих проблем зі здоров'ям, обладнанням та інформацією. Дотримуючись цих правил, ви зможете працювати за комп'ютером безпечно та комфортно.

Якщо ви дотримуватиметеся цих порад, ви зможете зменшити ризик виникнення проблем зі здоров'ям, пов'язаних із роботою за комп'ютером.

ВИСНОВКИ

У цій магістерській роботі проведено дослідження можливості застосування методів машинного навчання до аналізу ВАХ газочутливих шарів для покращення селективності газових сенсорів на основі фрактально-перколяційних наносистем ZnO/NiO за рахунок комп'ютерної обробки даних вимірювань, а також розроблено прототип програмного продукту. Використання комбінації унікальних фізичних властивостей наносистем ZnO/NiO із методами обробки даних - класифікації та кластеризації, таких як логістична регресія, XGBoost, та інші, дозволило створити модель системи, здатної ефективно ідентифікувати газу.

Для визначення типу газу було випробувано такі алгоритми як логістична регресія, наївний баєсівський класифікатор і дерево рішень (Decision Tree). Найбільш ефективним виявився метод класифікації Decision Tree або дерево рішень, який дав змогу виконати передбачення із точністю 98%, також не менш ефективно себе показала логістична регресія із точністю 89%. Альтернативно було випробувано метод кластеризації як приклад навчання без учителя для того, щоб показати, що, дійсно, дані ВАХ містять у собі приховану інформацію щодо типу газу, яка може бути розкрита через комп'ютерну обробку.

Для передбачення концентрації газу було випробувано такий алгоритм як XGBoost (eXtreme Gradient Boosting). Він виявився доволі ефективним, і при застосуванні метрик оцінювання таких як Mean Squared Error – MSE, коефіцієнт детермінації (R-squared) і Explained Variance Score, вони показали доволі гарні результати. Модель демонструє невелику середньоквадратичну відхиленість в розмірі 17%, що свідчить про її добру точність в прогнозуванні реальних даних. Коефіцієнт детермінації (R-squared) на рівні 0.998 підкреслює високий рівень здатності моделі пояснювати варіацію у вихідних даних. Також, близьке до 1 значення Explained Variance Score (0.998) свідчить про ефективність моделі в поясненні варіації цільового показника. Тому можна зробити висновок що цей

метод дав змогу виконати передбачення концентрації із високою точністю, що також було продемонстровано графічно.

Отримані результати свідчать про ефективність системи у визначенні типу та концентрації газів у реальних умовах. Модель демонструє низку цікавих результатів та аналізів, що робить її потенційно надійним інструментом для моніторингу газових речовин. Таким чином, отримані результати вказують на великий потенціал та перспективи використання методів машинного навчання для розпізнавання газів. Розроблений підхід може бути розвинений в подальшому для розпізнавання більшої кількості газів та їх сумішей, а також може бути застосований в різних галузях, включаючи промисловість, екологію та медицину.

СПИСОК ЛІТЕРАТУРИ

1. Yamazoe N Toward innovations of gas sensor technology. *Sens Actuators B* (2005).
2. Taylor RF, Schultz JS (eds) *Handbook of chemical sensors and biosensors*. IOP Publishing, Bristol (1996).
3. Ghenadii Korotcenkov *Handbook of Gas Sensor Materials. Properties, Advantages and Shortcomings for Applications. Volume 1: Conventional Approaches*. Springer Science+Business Media, LLC 2013.
4. Korotcenkov G, Stetter JR *Chemical gas mixture analysis and the electronic nose: current status, future trends*. In: Korotcenkov G (ed) *Chemical sensors: comprehensive sensor technologies, vol 6, Chemical sensors applications*. Momentum, New York, (2011).
5. Giovanni Neri *First Fifty Years of Chemoresistive Gas Sensors Chemosensors 2015*.
6. Ho CK, Itamura MT, Kelley M, Hughes RC *Review of chemical sensors for in-situ monitoring of volatile contaminants*. Sandia Report SAND2001-0643, Unlimited release, Sandia National Laboratories, Albuquerque, NM (2001).
7. C.I.L. Justino, T.A.P. Rocha-Santos, A.C. Duarte *Advances in point-of-care technologies with biosensors based on carbon nanotubes Trends Anal. Chem.*, 45 (2013).
8. Jacob Fraden. *Handbook of Modern Sensors: Physics, Designs, and Applications*. Third Edition (2004).
9. Jacob Fraden. *Handbook of Modern Sensors: Physics, Designs, and Applications*. Fourth Edition (2010).
10. Colomban, P., Berbyuk, B. (2007). "Metal Oxides in Gas Sensors for Gas Detection." *Sensors and Actuators B: Chemical*, 121(1), 18–35.
11. Khan, S., Cho, B., Tak, C. (2019). "Semiconductor Gas Sensors: Principles, Challenges, and Perspectives." *Sensors*, 19(11), 2749.

12. Corrigan, N., Su, W. (2007). "Polymeric Gas Sensors." *Journal of Macromolecular Science, Part C: Polymer Reviews*, 47(1), 21–49.
13. Li, C., Lu, G., Zhang, D., Tang, S., Li, Z. (2013). "Nanomaterials for Gas Sensors." *Nano-Micro Letters*, 5(4), 247–266.
14. Ju, H., Barsan, N., Weimar, U. (2011). "Micromachined Metal Oxide Gas Sensors: Opportunities to Improve Sensor Performance." *Sensors and Actuators B: Chemical*, 156(2), 907–937.
15. Sberveglieri, G., Nelli, P., Zampolli, S. (2004). "Comini E. Metal oxide nanowires: preparation and application in gas sensing." *Journal of Materials Chemistry*, 14(4), 514–519.
16. Korotcenkov, G., Cho, B. K. (2008). "Metal Oxides for Solid-State Gas Sensors: What Determines Our Choice?" *Materials Science and Engineering: B*, 139(1), 1–23.
17. Lee, J. H. (2005). Gas Sensors Using Hierarchical and Hollow Oxide Nanostructures: Overview. *Sensors and Actuators B: Chemical*, 110(1), 29–45.
18. Barsan, N., & Weimar, U. (2003). Understanding the Fundamental Principles of Metal Oxide Based Gas Sensors; the Example of CO Sensing with SnO₂ Sensors in the Presence of Humidity. *Journal of Physics: Condensed Matter*, 15(20), R813–R839.
19. Wang, J., Zhou, X., Cheng, Y., & Qian, D. (2019). "ZnO-based gas sensors: Synthesis and performance improvement." *Journal of Alloys and Compounds*, 797, 797-811.
20. Liu, H., Wang, Y., Yang, X., & Zhang, J. (2020). "Engineering the selectivity of metal oxide gas sensors." *ACS Sensors*, 5(5), 1257-1269.
21. Li, M., Wang, C., Zhang, Y., & Yin, L. (2018). "Stability and recovery of metal oxide gas sensors." *Sensors and Actuators B: Chemical*, 254, 1191-1205.
22. Jones, A. C., & Smith, R. L. (2017). "Temperature dependence of metal oxide gas sensors." *Journal of Applied Physics*, 121(16), 164502.

23. Zhang, L., Chen, Y., Liu, J., & Shi, Y. (2017). "Gas adsorption on metal organic framework and zeolite-based materials." *Chemical Reviews*, 117(16), 9979-10031.
24. Liang, J., Zhang, L., Xie, T., Cao, R., & Zhang, Q. (2021). "ZnO nanostructures for gas sensing applications: A review." *Sensors and Actuators B: Chemical*, 329, 129209.
25. Wang, Y., Wang, C., Zou, X., & Xiang, D. (2018). "Temperature-dependent gas sensing properties of ZnO nanorods." *Sensors and Actuators B: Chemical*, 266, 124-130.
26. Korniyushchenko, A., Kosminska, Y., Stas, S., Wilde, G., & Perekrestov, V. (2021). Structural, Morphological and Sensor Properties of the Fractal-Percolation Nanosystem ZnO/NiO. *Journal of Electronic Materials*, 50(1).
27. Mitchell, T. M. (1997). "Machine Learning." McGraw-Hill.
28. Bishop, C. M. (2006). "Pattern Recognition and Machine Learning." Springer.
29. Hastie, T., Tibshirani, R., & Friedman, J. (2009). "The Elements of Statistical Learning." Springer.
30. Zheng, B., & Casari, A. (2018). "Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists." O'Reilly Media.
31. Lee, J., Kim, H., & Lee, H. (2019). "Gas identification using support vector machines with a tin oxide gas sensor array." *Sensors and Actuators B: Chemical*, 297, 126731.
32. Wang, Y., Sun, Y., & Cai, G. (2020). "Gas concentration prediction using machine learning algorithms with feature selection from different gas sensors." *Sensors and Actuators B: Chemical*, 312, 127965.
33. Kim, H., Lee, J., & Lee, H. (2018). "Anomaly detection for gas sensors using autoencoders." *Sensors and Actuators B: Chemical*, 256, 365-371.
34. Zheng, B., & Casari, A. (2018). "Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists." O'Reilly Media.

35. Vergara, A.; Vembu, S.; Ayhan, T.; Ryan, M.A.; Homer, M.L.; Huerta, R. Chemical gas sensor drift compensation using classifier ensembles. *Sens. Actuators Chem.* 2012, 166, 320–329.
36. Deng, C.; Lv, K.; Shi, D.; Yang, B.; Yu, S.; He, Z.; Yan, J. Enhancing the discrimination ability of a gas sensor array based on a novel feature selection and fusion framework. *Sensors* 2018, 18, 1909.
37. Feng, S., Farha, F., Li, Q., Wan, Y., Xu, Y., Zhang, T., & Ning, H. (2019). Smart Gas Sensing Technology. *Sensors*, 19(17), 3754.

ДОДАТОК А

Машинне навчання. Код по класифікації двох газів.

```

# Імпорт потрібних бібліотек
import os
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, precision_score,
recall_score, f1_score, roc_auc_score
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import learning_curve, ShuffleSplit
from sklearn.metrics import classification_report
from sklearn.model_selection import cross_val_score
from sklearn.preprocessing import MinMaxScaler

# Проектування шляхів до потрібних файли
base_path = r'd:\onedrive - сумський державний університет\робочий стіл\диплом
магістра\Data\Dataset'
file_path_methane = os.path.join(base_path, 'methane_features.csv')

# Обробка даних про метан
methane_data = pd.read_csv(file_path_methane, header=None,
                           names=["Concentration", "I1", "U1", "I2", "U2", "I3",
                                   "I1/U1", "I2/U2", "I3/U3",
                                   "(I2-I1)/(U2-U1)", "(I3-I2)/(U3-U2)", "(I3-
                                   I1)/(U3-U1)"], delimiter=';', decimal=',')

# Загрузка даних та розподілення стовпців даних
columns_of_interest = ["Concentration", "I1", "U1", "I2", "U2", "I1/U1", "I2/U2",
                        "I3/U3", "(I2-I1)/(U2-U1)",
                        "(I3-I2)/(U3-U2)", "(I3-I1)/(U3-U1)"]
methane_features = methane_data[columns_of_interest]

# Створення датафрейму для розрахунку метану
methane_labels = pd.DataFrame({"Gas": "Methane", "Label": 1},
                              index=methane_features.index)

# Обробка даних про метанол
file_path_methanol = os.path.join(base_path, 'methanol_features.csv')

```

```

methanol_data = pd.read_csv(file_path_methanol, header=None,
                             names=["Concentration", "I1", "U1", "I2", "U2", "I3",
                                     "I1/U1", "I2/U2", "I3/U3",
                                     "(I2-I1)/(U2-U1)", "(I3-I2)/(U3-U2)", "(I3-
I1)/(U3-U1)"], delimiter=';', decimal=',')
methanol_features = methanol_data[columns_of_interest]
# Створення датафрейму для розрахунку метанолу
methanol_labels = pd.DataFrame({"Gas": "Methanol", "Label": 0},
                                index=methanol_features.index)
# Створення набору даних що об'єднує характеристики по метану і метанолу
methane_data_labeled = pd.concat([methane_features, methane_labels], axis=1)
methanol_data_labeled = pd.concat([methanol_features, methanol_labels], axis=1)
combined_data = pd.concat([methane_data_labeled, methanol_data_labeled],
                            ignore_index=True)
# Вилучення фіч та міток, ці лінії відокремлюють ознаки (X) та мітки (y) від
об'єданого набору даних
X = combined_data.drop(["Concentration", "Gas", "Label"], axis=1)
y = combined_data["Label"]
X = X.fillna(0)
def plot_learning_curve(estimator, title, X, y, ylim=None, cv=None,
                        n_jobs=1, train_sizes=np.linspace(.1, 1.0, 25)):
    plt.figure()
    plt.title(title)
    if ylim is not None:
        plt.ylim(*ylim)
    plt.xlabel("Training examples")
    plt.ylabel("Score")
    train_sizes, train_scores, test_scores = learning_curve(
        estimator, X, y, cv=cv, n_jobs=n_jobs, train_sizes=train_sizes)
    train_scores_mean = np.mean(train_scores, axis=1)
    train_scores_std = np.std(train_scores, axis=1)
    test_scores_mean = np.mean(test_scores, axis=1)
    test_scores_std = np.std(test_scores, axis=1)
    plt.grid()
    plt.fill_between(train_sizes, train_scores_mean - train_scores_std,
                    train_scores_mean + train_scores_std, alpha=0.1,
                    color="r")
    plt.fill_between(train_sizes, test_scores_mean - test_scores_std,
                    test_scores_mean + test_scores_std, alpha=0.1, color="g")
    plt.plot(train_sizes, train_scores_mean, 'o-', color="r",
            label="Training score")
    plt.plot(train_sizes, test_scores_mean, 'o-', color="g",

```

```

        label="Cross-validation score")
    plt.legend(loc="best")
    plt.show()
# Масштабування функцій, цей блок стандартизує функції за допомогою StandardScaler
від scikit-learn
scaler = MinMaxScaler()
X_scaled = scaler.fit_transform(X)
print(pd.DataFrame(X_scaled, columns=X.columns).head())
# Розподіл набору даних на навчальний, валідаційний та тестовий набори
# Розробка навчального набору (60%)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.4,
random_state=36, stratify=y)
# Розробка тестового та валідаційного набору з решти даних (40%)
X_val, X_test, y_val, y_test = train_test_split(X_test, y_test, test_size=0.5,
random_state=36, stratify=y_test)
# Виклик cross_val_score
kf = 4
# Результати
print("Розмір навчального набору:", len(X_train))
print("Розмір валідаційного набору:", len(X_val))
print("Розмір тестового набору:", len(X_test))
logistic_classifier = LogisticRegression()
logistic_cross_val_results = cross_val_score(logistic_classifier, X_scaled, y,
cv=kf, scoring='accuracy')
print(f'\nLogistic          Regression          Cross-Validation          Accuracy:
{logistic_cross_val_results.mean()}')
nb_classifier = GaussianNB()
nb_cross_val_results = cross_val_score(nb_classifier, X_scaled, y, cv=kf,
scoring='accuracy')
print(f'Naive Bayes Cross-Validation Accuracy: {nb_cross_val_results.mean()}')
dt_classifier = DecisionTreeClassifier()
dt_cross_val_results = cross_val_score(dt_classifier, X_scaled, y, cv=kf,
scoring='accuracy')
print(f'Decision Tree Cross-Validation Accuracy: {dt_cross_val_results.mean()}')
# Використання логістичної регресії як класифікатор
classifier = LogisticRegression(random_state = 0).fit(X_train, y_train)
# Прогнозування типу газу для нових точок даних
predictions = classifier.predict(X_test)
# Оцінка продуктивності моделі, вивід Confusion Matrix, вивід точності, відгуку і
результат F1.
accuracy = accuracy_score(y_test, predictions)
conf_matrix = confusion_matrix(y_test, predictions)

```

```

precision = precision_score(y_test, predictions)
recall = recall_score(y_test, predictions)
f1 = f1_score(y_test, predictions)
print('\nLogistic Regression Results:')
print(f'Accuracy: {accuracy}')
print(f'Confusion Matrix:\n{conf_matrix}')
print(f'Precision: {precision}')
print(f'Recall: {recall}')
print(f'F1 Score: {f1}')
# Звіт про класифікацію
print(classification_report(y_test, predictions))
# Learning Curve (Крива навчання)
title = "Learning Curves (Logistic Regression)"
cv = ShuffleSplit(n_splits=100, test_size=0.2, random_state=12)
plot_learning_curve(classifier, title, X_scaled, y, ylim=(0.7, 1.01), cv=cv,
n_jobs=4)
# Навчання та оцінювання Naive Bayes Classifier
naive_bayes_classifier = GaussianNB()
naive_bayes_classifier.fit(X_train, y_train)
nb_predictions = naive_bayes_classifier.predict(X_test)
nb_accuracy = accuracy_score(y_test, nb_predictions)
nb_precision = precision_score(y_test, nb_predictions)
nb_recall = recall_score(y_test, nb_predictions)
nb_f1 = f1_score(y_test, nb_predictions)
nb_conf_matrix = confusion_matrix(y_test, nb_predictions)
print('Naive Bayes Classifier Results:')
print(f'Accuracy: {nb_accuracy}')
print(f'Confusion Matrix: {nb_conf_matrix}')
print(f'Precision: {nb_precision}')
print(f'Recall: {nb_recall}')
print(f'F1 Score: {nb_f1}')
# Звіт про класифікацію
print(classification_report(y_test, nb_predictions))
# Learning Curve (Naive Bayes)
title = "Learning Curves (Naive Bayes)"
cv = ShuffleSplit(n_splits=100, test_size=0.2, random_state=0)
plot_learning_curve(naive_bayes_classifier, title, X_scaled, y, ylim=(0.7, 1.01),
cv=cv, n_jobs=4)
# Навчання та оцінювання Decision Tree Classifier
decision_tree_classifier = DecisionTreeClassifier()
decision_tree_classifier.fit(X_train, y_train)
dt_predictions = decision_tree_classifier.predict(X_test)

```

```

dt_accuracy = accuracy_score(y_test, dt_predictions)
dt_precision = precision_score(y_test, dt_predictions)
dt_recall = recall_score(y_test, dt_predictions)
dt_f1 = f1_score(y_test, dt_predictions)
print('Decision Tree Classifier Results:')
print(f'Accuracy: {dt_accuracy}')
print(f'Precision: {dt_precision}')
print(f'Recall: {dt_recall}')
print(f'F1 Score: {dt_f1}')
# Звіт про класифікацію
print(classification_report(y_test, dt_predictions))
# Learning Curve (Decision Tree)
title = "Learning Curves (Decision Tree)"
cv = ShuffleSplit(n_splits=100, test_size=0.2, random_state=0)
plot_learning_curve(decision_tree_classifier, title, X_scaled, y, ylim=(0.7,
1.01), cv=cv, n_jobs=4)
# Обчислення і вивід оцінки ROC-AUC, якщо проблема полягає у бінарній класифікації
if len(set(y_test)) == 2:
    roc_auc = roc_auc_score(y_test, classifier.predict_proba(X_test)[:, 1])
    print(f'\nROC-AUC Score: {roc_auc}')
else:
    print('\nROC-AUC Score does not apply to multi-class classification.')
# Візуалізація рішучих границь для нового класифікатора
plt.figure(figsize=(10, 6))
plt.scatter(X_test[y_test == 0][:, 0], X_test[y_test == 0][:, 1], marker='o',
label='Class 0', s=80, alpha=0.8)
plt.scatter(X_test[y_test == 1][:, 0], X_test[y_test == 1][:, 1], marker='^',
label='Class 1', s=80, alpha=0.8)
plt.title('Розподіл точок після навчання моделі')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.legend()
plt.show()

```

Машинне навчання. Код по розпізнаванню концентрації газів.

```

import os
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

```

```

from sklearn.metrics import explained_variance_score
import matplotlib.pyplot as plt
import statsmodels.api as sm
from xgboost import XGBRegressor
from sklearn.metrics import mean_squared_error, r2_score
# Читання даних
base_path = r'd:\onedrive - сумський державний університет\робочий стіл\диплом
магістра\Data\Dataset'
file_path_methane = os.path.join(base_path, 'methane_features.csv')
file_path_methanol = os.path.join(base_path, 'methanol_features.csv')
methane_data = pd.read_csv(file_path_methane, header=None,
                           names=["Concentration", "I1", "U1", "I2", "U2", "I3",
                                  "I1/U1", "I2/U2", "I3/U3",
                                  "(I2-I1)/(U2-U1)", "(I3-I2)/(U3-U2)", "(I3-
I1)/(U3-U1)"], delimiter=';', decimal=',')
methane_data["Gas"] = "Methane"
methanol_data = pd.read_csv(file_path_methanol, header=None,
                             names=["Concentration", "I1", "U1", "I2", "U2", "I3",
                                    "I1/U1", "I2/U2", "I3/U3",
                                    "(I2-I1)/(U2-U1)", "(I3-I2)/(U3-U2)", "(I3-
I1)/(U3-U1)"], delimiter=';', decimal=',')
methanol_data["Gas"] = "Methanol"
combined_data = pd.concat([methane_data, methanol_data], ignore_index=True)
combined_data['Gas'] = combined_data['Gas'].astype('category')
combined_data['Gas'] = combined_data['Gas'].cat.codes
print(combined_data)
# Експериментальні дані для метанолу і метану
methanol_data_lab = {
    "I1": [1780, 1560, 1350, 1050, 750, 550, 500, 400],
    "I2": [1400, 1200, 950, 700, 580, 400, 300, 230],
    "I3": [1800, 1500, 1280, 950, 650, 550, 400, 300],
    "U1": [1650, 1650, 1650, 1700, 1700, 1700, 1760, 1870],
    "U2": [2500, 2550, 2700, 2800, 2900, 2950, 3100, 3200],
    "Concentration": [1850, 1650, 1400, 1100, 750, 410, 280, 140]
}
methane_data_lab = {
    "I1": [2450, 1950, 1800, 1600, 1500, 1450, 1250, 900, 450],
    "I2": [2530, 1900, 1650, 1400, 1250, 1100, 900, 550, 250],
    "I3": [4100, 2800, 2350, 1900, 1750, 1500, 1280, 950, 500],
    "U1": [1550, 1570, 1580, 1650, 1670, 1670, 1700, 1780, 1800],
    "U2": [2050, 2150, 2300, 2400, 2450, 2500, 2600, 2800, 2900],
    "Concentration": [1800, 1280, 1020, 820, 660, 520, 410, 210, 90]
}

```



```

}
# Вилучення фіч та міток
X = combined_data.drop(["Concentration", "I3"], axis=1)
y = combined_data["Concentration"]
print(y)
print(y.describe())
# Масштабування функцій
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
# Розбиття на навчальний та тестовий набори
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2,
random_state=32)
# XGBoost для передбачення концентрації
xgb_model = XGBRegressor(objective="reg:squarederror", n_estimators=140,
learning_rate=0.1)
xgb_model.fit(X_train, y_train)
# Прогнози та оцінка результатів
predictions = xgb_model.predict(X_test)
mse = mean_squared_error(y_test, predictions, squared=False)
r2 = r2_score(y_test, predictions)
print(f'Mean Squared Error: {mse}')
print(f'R-squared: {r2}')
# Експлікативна дисперсія (Explained Variance): Вимірює частку дисперсії, яку
пояснює модель.
explained_variance = explained_variance_score(y_test, predictions)
print(f'Explained Variance: {explained_variance}')
#Графік прогнозів проти спостережень: Виведення графіку, на якому осі x - це
спостереження, а осі y - відповідний прогноз.
# Графік прогнозів проти спостережень
plt.scatter(y_test, predictions, label='Predictions')
plt.scatter(methanol_data_lab["Concentration"],
methanol_data_lab["Concentration"], color='red', label='Experimental point')
plt.scatter(methane_data_lab["Concentration"],
methane_data_lab["Concentration"], color='red')
plt.xlabel('Спостереження')
plt.ylabel('Прогноз')
plt.title('Графік прогнозів проти спостережень')
plt.legend()
plt.show()

```

Машинне навчання. Код по кластеризації двох газів.

```

import os
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.manifold import TSNE
from sklearn.preprocessing import MinMaxScaler
# Дані для метану
methane_data_lab = {
    "I1": [2450, 1950, 1800, 1600, 1500, 1450, 1250, 900, 450],
    "I2": [2530, 1900, 1650, 1400, 1250, 1100, 900, 550, 250],
    "I3": [4100, 2800, 2350, 1900, 1750, 1500, 1280, 950, 500],
    "U1": [1550, 1570, 1580, 1650, 1670, 1670, 1700, 1780, 1800],
    "U2": [2050, 2150, 2300, 2400, 2450, 2500, 2600, 2800, 2900],
    "Concentration": [1800, 1280, 1020, 820, 660, 520, 410, 210, 90]
}
# Створення DataFrame для метану
methane_df = pd.DataFrame(methane_data_lab)
# Додавання нових стовпців
methane_df['I1/U1'] = methane_df['I1'] / methane_df['U1']
methane_df['I2/U2'] = methane_df['I2'] / methane_df['U2']
methane_df['I3/U3'] = methane_df['I3'] / 5000 # U3 = 5000
methane_df['(I3-I1)/(U3-U1)'] = (methane_df['I3'] - methane_df['I1']) / (5000 -
methane_df['U1'])
# Дані для метанолу
methanol_data_lab = {
    "I1": [1780, 1560, 1350, 1050, 750, 550, 500, 400],
    "I2": [1400, 1200, 950, 700, 580, 400, 300, 230],
    "I3": [1800, 1500, 1280, 950, 650, 550, 400, 300],
    "U1": [1650, 1650, 1650, 1700, 1700, 1700, 1760, 1870],
    "U2": [2500, 2550, 2700, 2800, 2900, 2950, 3100, 3200],
    "Concentration": [1850, 1650, 1400, 1100, 750, 410, 280, 140]
}
# Створення DataFrame для метанолу
methanol_df = pd.DataFrame(methanol_data_lab)
# Додавання нових стовпців
methanol_df['I1/U1'] = methanol_df['I1'] / methanol_df['U1']
methanol_df['I2/U2'] = methanol_df['I2'] / methanol_df['U2']
methanol_df['I3/U3'] = methanol_df['I3'] / 5000 # U3 = 5000
methanol_df['(I3-I1)/(U3-U1)'] = (methanol_df['I3'] - methanol_df['I1']) / (5000
- methanol_df['U1'])
# Виведення даних
print("Methane Data:")
print(methane_df)
print("\nMethanol Data:")
print(methanol_df)
# Об'єднання даних
combined_data_lab = pd.concat([methane_df, methanol_df], ignore_index=True)
# Шкалювання даних
X_combine_lab = MinMaxScaler().fit_transform(combined_data_lab.drop(["Concentration", "I3"],
axis=1))
# Вибір кількості кластерів
clusterNum = 2
# Метод KMeans
kmeans = KMeans(init="k-means++", n_clusters=clusterNum, n_init=10)
labels = kmeans.fit_predict(X_combine_lab)
# Виведення деяких точок даних для кожного кластера
print("\nMethane and Methanol Data:")
print(pd.DataFrame({'Concentration': combined_data_lab['Concentration'],
'Cluster': labels}).to_string(index=False))

```

```

# t-SNE
tsne = TSNE(n_components=2, perplexity=4, random_state=42)
reduced_data_tsne = tsne.fit_transform(X_combine_lab)
# Вигрузка даних для тестування
base_path = r'd:\onedrive - сумський державний університет\робочий стіл\диплом
марістра\Data\Dataset'
file_path_methane = os.path.join(base_path, 'methane_features.csv')
file_path_methanol = os.path.join(base_path, 'methanol_features.csv')

methane_data = pd.read_csv(file_path_methane, header=None,
                           names=["Concentration", "I1", "U1", "I2", "U2", "I3",
                                   "I1/U1", "I2/U2", "I3/U3",
                                   "(I2-I1)/(U2-U1)", "(I3-I2)/(U3-U2)", "(I3-
                                   I1)/(U3-U1)"], delimiter=';', decimal=',')

methanol_data = pd.read_csv(file_path_methanol, header=None,
                             names=["Concentration", "I1", "U1", "I2", "U2", "I3",
                                     "I1/U1", "I2/U2", "I3/U3",
                                     "(I2-I1)/(U2-U1)", "(I3-I2)/(U3-U2)", "(I3-
                                     I1)/(U3-U1)"], delimiter=';', decimal=',')
# Фічі для кластеризації
columns_of_interest = ["I1", "I2", "I3", "U1", "U2", "Concentration", "I1/U1",
                       "I2/U2", "I3/U3", "(I3-I1)/(U3-U1)"]
# Різні дані для тестингу
test_methane_data = methane_data[columns_of_interest].sample(n=3)
test_methanol_data = methanol_data[columns_of_interest].sample(n=3)
combined_data_test = pd.concat([test_methane_data, test_methanol_data],
                                ignore_index=True)
# Створення DataFrame для тестових даних метану
test_methane_df = pd.DataFrame(combined_data_test)
print(test_methane_df)
# Вилучення зайвого стовпця перед шкалюванням
X_test = MinMaxScaler().fit_transform(test_methane_df)
print(X_test)
print("Shape of X_test_methane before prediction:", X_test.shape)
# Передбачення кластерів для тестових даних
test_labels_methane = kmeans.fit_predict(X_test)
print(test_labels_methane)
# Виведення результатів для тестових даних метану
print("\nTest Data:")
print(pd.DataFrame({'Cluster': test_labels_methane}).to_string(index=False))
# Підсумування кластерів (1s and 0s)
cluster_summary = pd.DataFrame({'Cluster': test_labels_methane})
cluster_counts = cluster_summary['Cluster'].value_counts()
# Вивід суми кластерів
print("Cluster Summary:")
print(cluster_counts)
# Графік t-SNE кластеризації
plt.scatter(reduced_data_tsne[:, 0], reduced_data_tsne[:, 1], c=labels,
            cmap='viridis')
# Графік t-SNE кластеризації тестових даних
test_data_tsne = tsne.fit_transform(X_test)
plt.scatter(test_data_tsne[:, 0], test_data_tsne[:, 1], c=test_labels_methane,
            cmap='Set1', marker='X')
plt.title('t-SNE Clustering Results with Test Data Points')
plt.xlabel('t-SNE Component 1')
plt.ylabel('t-SNE Component 2')
plt.show()

```