

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Нейромережа обробки даних хімічних аналізів
(тема)

Виконав:
студент 2 курсу, групи СШМ-22-3
Михно Є. В.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту
(повна назва спеціалізації)

Керівник проф. Сніжко Д. В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

В.О. Філатов
(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
(повна назва)
Кафедра _____ Штучного інтелекту _____
(повна назва)
Рівень вищої освіти _____ другий (магістерський) _____
Спеціальність _____ 122 Комп'ютерні науки _____
(код і повна назва)
Тип програми _____ освітньо-наукова _____
(освітньо-професійна або освітньо-наукова)
Освітня програма _____ Системи штучного інтелекту _____
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
« _____ » _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Михну Євгену Віталійовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Нейромережа обробки даних хімічних аналізів _____

затверджена наказом університету від 1 квітня 20 24 р. № 260Ст

2. Термін подання студентом роботи до екзаменаційної комісії 5 червня 20 24 р.

3. Вихідні дані до роботи _____ Науково-технічні публікації та дані різних печатних та _____
інтернет-джерел щодо тематики кваліфікаційної роботи _____

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Використання систем штучного інтелекту при обробці даних хімічних аналізів _____

2) Опрацювання даних з хімічних сенсорів «електронного язика» засобами штучного інтелекту _____

3) Попередня обробка даних з хімічних сенсорів _____

4) Проектування нейронної мережі для обробки даних хімічних сенсорів _____

5) Практичне порівняння різних способів аналізу даних з хімічних сенсорів _____

РЕФЕРАТ

Пояснювальна записка: 68 с., 24 рис., 5 табл., 1 дод., 39 джерел.

ВІЗУАЛЬНІ ТРАНСФОРМЕРИ, ЕЛЕКТРОННИЙ ЯЗИК, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, МАШИННЕ НАВЧАННЯ, МЕТОД ГОЛОВНИХ КОМПОНЕНТ, НЕЙРОМЕРЕЖІ, ОБРОБКА ХІМІЧНИХ АНАЛІЗІВ, ШТУЧНИЙ ЯЗИК.

Об'єкт дослідження – аналіз речовин на основі даних хімічних сенсорів.

Предмет дослідження – різні методи класифікації хімічних речовин з використанням «штучного язика» – набору хімічних сенсорів.

Мета роботи – дослідити ефективність різних підходів до класифікації хімічних речовин, використовуючи методи класичного машинного навчання, а також різні види нейронних мереж. Акцент робиться на пошуку оптимального шляху для класифікації та підвищення точності результатів.

Методи дослідження охоплюють глибоке дослідження наявних наукових робіт з дотичних тем, комплексний аналіз технологій, що використовуються для розробки сенсорів для аналізу та класифікації речовин, пошук даних для аналізу та експериментальні дослідження.

ABSTRACT

Master's thesis contains: 68 pp., 24 fig., 5 tabl., 1 ann., 39 sources.

ARTIFICIAL TONGUE, CHEMICAL ANALYSIS PROCESSING, CONVOLUTIONAL NEURAL NETWORKS, ELECTRONIC TONGUE, MACHINE LEARNING, NEURAL NETWORKS, PRINCIPAL COMPONENT ANALYSIS, VISUAL TRANSFORMERS.

The object of research is a classification of substances based on chemical sensors.

The subject of research is different approaches to the classification of chemicals using using an «artificial tongue» – a set of chemical sensors.

The study's purpose is to investigate the effectiveness of different approaches to chemical classification using classical machine learning methods and various types of neural networks. The emphasis is on finding the best way to classify and improve the accuracy of the results.

Research methods include an in-depth study of existing scientific papers on related topics, a comprehensive analysis of technologies used for sensor development for chemical analysis and classification, data mining, and experimental studies.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	8
Вступ	9
1 Аналіз предметної галузі	11
1.1 Хімічний аналіз	11
1.2 Електронний язык як інструмент хімічного аналізу	11
1.3 Типи електронних языків та вихідні дані	14
1.4 Метод головних компонент	16
1.5 PCA при обробці даних сенсорів	17
1.6 Штучні нейронні мережі	18
1.7 ШНМ при обробці даних сенсорів	22
2 Аналіз опорних даних	24
2.1 Вибір даних. Опорна стаття	24
2.2 Методологія опорної статті	24
2.3 Дані вин для аналізу	27
3 Методологія та експериментальні дослідження	28
3.1 Робота з часовими рядами	28
3.2 Попередня обробка даних	29
3.2.1 Розгортання матриць	29
3.2.2 Перетворення часових рядів на картинки	30
3.3 Попередня обробка наявних даних	36
3.4 Класифікація зображень	40
3.4.1 Глибоке навчання у класифікації зображень	41
3.5 Класифікація графічного відображення винних напоїв	43
3.6 Результати досліджень	45
3.6.1 Результати для нерозгорнутих матриць	45
3.6.2 Результати для розгорнутих матриць	57
Висновки	61
Перелік джерел посилання	63

Додаток А Відомість кваліфікаційної роботи 68

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

- MH – машинне навчання;
NM – нейронна мережа;
ШІ – штучний інтелект;
ШНМ – штучні нейронні мережі;
AI – Artificial Intelligence – штучний інтелект;
ANN – Artificial Neural Network – штучні нейронні мережі;
CNN – Convolutional Neural Network – згорткова нейронна мережа;
ET – Electronic Tongue – електронний язик;
GAF – Gramian Angular Field – кутове поле Грама;
ISEs – Ion Selective Electrodes – іоноселективні електроди;
KNN – k-Nearest Neighbours – k найближчих сусідів;
ML – Machine Learning – машинне навчання;
MTF – Markov transition field – Марківське перехідне поле;
NB – Naive Bayes – наївний Байєс;
NN – Neural Network – нейронна мережа;
PARC – Pattern Recognition – розпізнавання патернів;
PCA – Principal Component Analysis – метод головних компонент;
RP – Recurrence Plot – рекурентне перетворення;
RQA – Recurrence Quantification Analysis – аналіз кількісної оцінки повторів;
SGD – Stochastic Gradient Descent – стохастичний градієнтний спуск.

ВСТУП

Останніми роками сфера хімічного аналізу стала свідком глибокої трансформації, спричиненої досягненнями в галузі штучного інтелекту та методів машинного навчання. Серед них нейронні мережі стали потужним інструментом для обробки складних наборів даних, пропонуючи безпрецедентні можливості для розпізнавання образів, класифікації та прогнозного моделювання. Поєднання нейронних мереж з хімічним аналізом має величезний потенціал для революційних змін у різних галузях, включаючи пошук ліків, моніторинг навколишнього середовища, матеріалознавства тощо.

Традиційно хімічний аналіз покладається на звичайні статистичні методи та ручну інтерпретацію даних, які часто виявляються трудомісткими, тривалими й обмеженими у своїй здатності виявляти складні взаємозв'язки у величезних масивах даних. Однак поява нейронних мереж відкрила нові шляхи для подолання цих викликів, уможлививши автоматизоване вилучення ознак, нелінійне моделювання та адаптивне навчання на основі даних.

Окрему увагу у хімічному аналізі привертає концепція електронного язика, що імітує людський процес розпізнавання різних речовин на смак.

Електронний язик являє собою аналітичний інструмент – набір сенсорів та обчислювальний модуль, які можуть виявляти та ідентифікувати різні хімічні сполуки на основі їх смакових профілів і не тільки. Спочатку натхненні смаковою системою людини, електронні язики перетворилися на складні пристрої, здатні розпізнавати тонкі відмінності у складі широкому спектру зразків. Ця технологія має величезні перспективи для таких застосувань, як контроль якості у виробництві харчових продуктів та напоїв, моніторинг навколишнього середовища та медична діагностика.

Центральне місце в ефективності електронних язиків займає

розробка надійних алгоритмів обробки даних, особливо тих, що базуються на нейронних мережах. Нейронні мережі, натхненні біологічними нейронними мережами людського мозку, чудово розпізнають закономірності та витягують значущу інформацію зі складних наборів даних. Використовуючи ці можливості, дослідники підвищують точність, швидкість і універсальність штучних мов.

Окремою можливістю використання такого інструмента як електронний мов є різноманітні задачі класифікації продуктів. В роботі розглянуто можливості сучасних методів машинного навчання та побудови нейронних мереж для класифікації винних напоїв на основі вже готового набору даних.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Хімічний аналіз

Хімічний аналіз – визначення фізичних властивостей або хімічного складу зразків речовини. Велика кількість систематичних процедур, призначених для цих цілей, постійно розвивається в тісному зв'язку з розвитком інших галузей інженерних наук з моменту їх зародження.

Хімічний аналіз, який ґрунтується на використанні вимірювань, поділяється на дві категорії залежно від способу виконання аналізів [1]. Класичний аналіз, який також називають мокрим хімічним аналізом, складається з тих аналітичних методів, які не використовують жодних механічних або електронних приладів, окрім терезів. Метод зазвичай ґрунтується на хімічних реакціях між матеріалом, що аналізується (аналітом), і реагентом, який додається до аналіту. Мокрі методи часто залежать від утворення продукту хімічної реакції, який легко виявити та виміряти. Наприклад, продукт може бути забарвленим або являти собою тверду речовину, яка випадає в осад з розчину.

Більшість хімічних аналізів належить до другої категорії – інструментального аналізу. Він передбачає використання інструменту, відмінного від терезів, для виконання аналізу. У розпорядженні аналітика є широкий асортимент приладів. В одних випадках прилад використовується для характеристики хімічної реакції між аналізованою речовиною і доданим реагентом, в інших – для вимірювання властивості аналізованої речовини.

1.2 Електронний язик як інструмент хімічного аналізу

Електронний язик належить саме до другого типу хімічних аналізів – інструментальних.

Електронний язик – це аналітичний інструмент, що складається з набору неспецифічних, низькоселективних хімічних сенсорів з високою стабільністю і перехресною чутливістю до різних видів у речовині, а також відповідного методу PARC і/або багатовимірного калібрування для обробки даних [2].

Принцип роботи такого штучного язика (рисунок 1.1), був натхненний біологічним розпізнаванням, в якому інформація збирається за допомогою масивів неспецифічних датчиків в носі або язичі, а дані згодом обробляються в мозку. Таким чином електронний язик працює наступним чином: фізичні, хімічні та біохімічні властивості зразків вимірюються за допомогою набору датчиків, які перетворюють ці специфічні характеристики в аналітичний сигнал (оптичний, електрофізіологічний, електрохімічний тощо). Отримані таким чином дані потім аналізуються за допомогою методів хімічного аналізу або нейронних мереж, які надають остаточну інформацію про зразок – наприклад, розрізняють зразки кави за їх географічним походженням [3], [4].

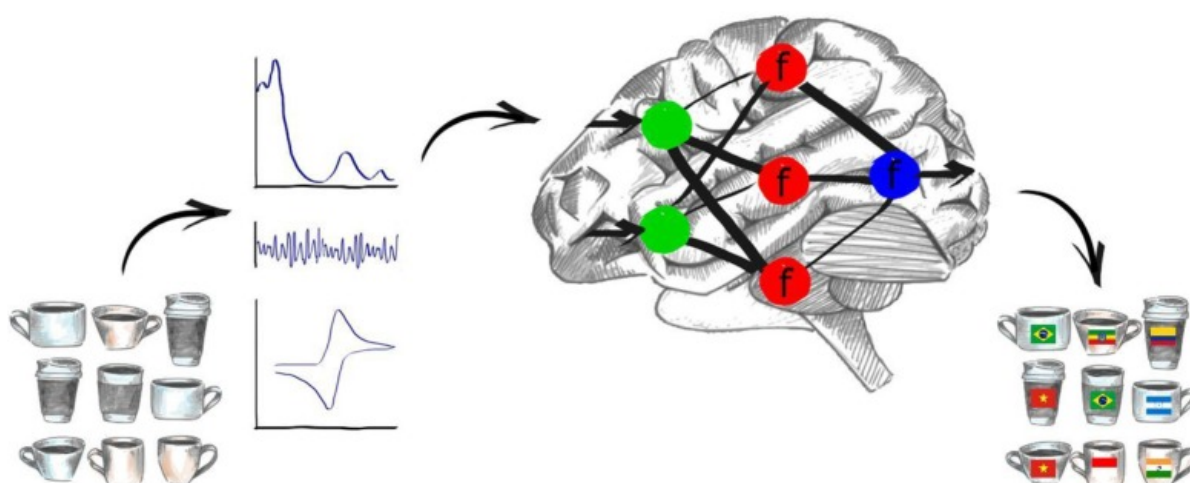


Рисунок 1.1 – Принцип роботи системи електронного язика [3]

В основному, електронний язик складається з блоку збору інформації для використання у водній фазі, підключеного до комп'ютера для обробки даних за допомогою багатовимірної регресії або аналізу штучної нейронної мережі. При обробці зразка, що містить різні сполуки, генерується вихідний патерн, який являє собою синтез всіх компонентів у зразку. Він визначається різною селективністю окремих сенсорних одиниць і асоціюється з певним смаком або аспектом якості. В основі методу лежить те, що хоча специфічність кожного сенсорного елемента є низькою, поєднання декількох класів специфічності надає величезний потенціал для аналітичної роботи [5].

Електронний язик може бути застосований у наступних сферах:

- аналіз харчових продуктів і напоїв: контроль якості продуктів, таких як вино, чай, кава, молочні продукти, виявлення підробок та фальсифікацій, оцінка свіжості продуктів, або зрілості овочів та фруктів тощо [6];

- охорона здоров'я: діагностика захворювань шляхом аналізу біологічних рідин (сеча, слина, кров) [7];

- фармацевтична промисловість: перевірка якості та стабільності фармацевтичних препаратів, оцінка смакових якостей [8];

- екологічний моніторинг: контроль якості води, повітря тощо та виявлення забруднювальних речовин у навколишньому середовищі [9];

- хімічна промисловість: контроль якості хімічних реакцій та продуктів, виявлення домішок у виробничих процесах тощо.

Серед переваг застосування подібного інструменту хімічного аналізу можна виділити наступні:

- висока точність та повторюваність: дозволяє уникнути суб'єктивних помилок людського аналізу;

- швидкість аналізу: забезпечує оперативний аналіз великої кількості зразків;

- багатовимірний аналіз: здатен одночасно аналізувати декілька

параметрів речовини;

- зниження вартості: мінімізує витрати на хімічні реагенти.

1.3 Типи електронних язиків та вихідні дані

Для формування масиву сенсорів можуть бути використані практично всі основні сімейства хімічних сенсорів, а саме: потенціометричні, вольтамперометричні, резистивні, гравіметричні та оптичні. Також використовуються гібридні системи, переважно ті, що поєднують потенціометричні та вольтамперометричні датчики [10]. Вибір датчиків має першочергове значення, оскільки в результаті роботи набори датчиків генерують величезну кількість комплексних вихідних даних, це стається через те, що всі датчики можуть реагувати на всі аналіти, таким чином використання електронного язика вимагає перехресної вибірковості [11].

Хімічні сенсори зазвичай класифікують за способом перетворення хімічного сигналу, поділяючи їх на електрохімічні, оптичні тощо. Електрохімічні сенсори зазвичай поділяються на вольтамперометричні, амперометричні, сенсори імпедансної спектроскопії та потенціометричні сенсори [12].

Принцип дії амперометричних сенсорів полягає у зміні струму на робочих електродах залежно від концентрації аналізованої речовини як функції часу при фіксованому потенціалі [12].

Вольтамперометричні датчики виявляють аналіт за зміною струму залежно від концентрації як функції прикладеного потенціалу. Циклічна вольтамперометрія і диференціальна імпульсна вольтамперометрія є основними електрохімічними методами аналізу забруднювачів навколишнього середовища [12].

Датчики електрохімічної імпедансної спектроскопії (ЕІС) використовували для виявлення аналітів шляхом вимірювання зміни

імпедансу залежно від концентрації зразка. Для отримання кількісних результатів часто використовували еквівалентні схеми для побудови імпедансних графіків. Нещодавно було створено біосенсор на основі імпедансу для виявлення *Listeria monocytogenes* двоетапним методом, тобто спочатку імуномагнітне розділення, а потім ЕІС детектування сили іонізації розчину, спричиненої каталізатором, і визначення іонної сили розчину [12].

Потенціометричні сенсори в основному визначають концентрацію аналізованої речовини, вимірюючи зміну різниці потенціалів між робочим і референтним електродами при різних концентраціях аналізованої речовини. До таких сенсорів належать іоноселективні електроди [12]. На основі таких датчиків часто будуються електронні язики для якісної ідентифікації напоїв або розпізнавання їхнього старіння [13], також з використанням такої системи було успішно продемонстровано одночасне мультивизначення металів у шахтному фільтраті, а також фізіологічних іонів у змодельованій плазмі крові людини [4].

При обробці даних з хемометричних даних сенсорів часто використовується метод головних компонент. Таким чином зменшується розмірність змінних шляхом перетворення їх у новий простір основних компонент, що спрощує подальшу роботу з даними. Але, оскільки зазвичай при використанні РСА, треба залучати складні термодинамічні розрахунки та моделювання, на заміну цьому методу приходять штучні нейронні мережі (ШНМ). ШНМ є потужними при моделюванні нелінійних систем, а суміжні дані від наборів датчиків є дуже нелінійними, різні експерименти дали відмінний результат, дозволяючи проводити як якісний аналіз, такий як ідентифікація видів, класифікація сортів зразків, розпізнавання фальсифікації, так і кількісний аналіз, наприклад, для контролю певного процесу.

1.4 Метод головних компонент

Аналіз головних компонент (метод головних компонент, PCA) – це метод зменшення розмірності великих наборів даних, які важко інтерпретувати з використанням інших алгоритмів. Він підвищує можливості для їхньої інтерпретації та водночас мінімізує втрату інформації. Це досягається шляхом створення нових некорельованих змінних, які послідовно максимізують дисперсію. Знаходження таких нових змінних, головних компонент, зводиться до розв'язання проблеми власних значень/власних векторів, а нові змінні визначаються набором даних, а не апріорно, що робить PCA адаптивною технікою аналізу даних.

Оскільки на меті роботи не стоїть дослідження методу головних компонент, буде надано короткий опис алгоритму.

PCA базується на ідеї, що можливо знайти набір змінних, які представляють більшу частину мінливості в наборі даних. Ці змінні називаються головними компонентами, і їх можна обчислити за допомогою дисперсійного та кореляційного аналізу.

Щоб обчислити головні компоненти, спочатку необхідно нормалізувати дані, тобто зробити так, щоб усі змінні мали нульове середнє значення та одиничну дисперсію. Потім слід обчислити коваріаційну матрицю нормалізованих даних. Коваріаційна матриця – це квадратна матриця, що містить коефіцієнти коваріації між усіма змінними. Коваріаційні коефіцієнти показують, як дві змінні пов'язані між собою.

Наступним кроком слід обчислити власні вектори або характеристичні вектори матриці. Власні вектори – це вектори, які не змінюють напрямку, коли до них застосовується лінійне перетворення, наприклад, поворот або відображення. Кожному власному вектору відповідає власне значення або характеристичне значення, яке є скаляром, що представляє величину перетворення, застосованого до

власного вектора. Власні вектори та власні значення можна обчислити, розв'язавши рівняння $(A - \lambda I) \cdot x = 0$, де A – коваріаційна матриця, а λ – власне значення [14].

Після того, як знайдені власні вектори та власні значення коваріаційної матриці, слід власні вектори з найбільшими власними значеннями як головні компоненти. Власні вектори з найбільшими власними значеннями – це ті, що представляють найбільшу варіабельність у наборі даних.

1.5 PCA при обробці даних сенсорів

Як вже було сказано, PCA – це потужний лінійний неконтрольований метод розпізнавання образів, який зменшує розмірність багатовимірної задачі та допомагає візуалізувати різні категорії багатовимірних профілів, виділяючи схожість і відмінності між кластерами вибірки. По суті, PCA виконує зміну напрямків осей у просторі даних таким чином, щоб отримати як першу вісь ті, що мають максимальну дисперсію варіації. Оригінальна матриця даних X перераховується в наближенні як добуток двох нових матриць зменшеної розмірності $X = TP^T$, де P – матриця навантажень, тобто трансформація в нових напрямках, і T – матриця оцінок, тобто координати в нових напрямках. Зазвичай T і P розраховуються таким чином, щоб більша частина вихідної дисперсії зберігалася в перших 2 – 4 напрямках. Таким чином, низьковимірний гіперплощина аналізується для вивчення угруповань або тенденцій X . Розрахунок власного вектора $(X^T X)$ навантажень дозволяє ранжувати їх у відсотках поясненої дисперсії [10].

Застосуванням PCA для кількісного аналізу чудово демонструє приклад використання масиву іоноселективних електродів для отримання вхідних даних [10]. У досить складній системі, де для певного первинного іона можуть бути присутні певні завади, термодинамічне співвідношення

можна встановити, використовуючи вираз Нікольського-Ейзенмана [10], який визначає реакцію i сенсора на активність іонів, що заважають:

$$E_i = K_i + s_i \log[a_1 + k_{1,j}^{pot} \cdot (a_j)^{z_1/z_j} + k_{1,k}^{pot} \cdot (a_k)^{z_1/z_k} + \dots], \quad (1.1)$$

де a_1 – первинний іон цього датчика;

a_j, a_k, \dots – іони, що заважають, із зарядом z_j, z_k, \dots відповідно;

K_i – константа електрода;

s_i – чутливість електрода.

Перехресна чутливість виражається через потенціометричний коефіцієнт селективності $k_{1,h}^{pot}$, міру того, як іон, що заважає, створює спотворювальний відгук при класифікації. Враховуючи, що кількість ISEs буде більшою, ніж кількість класів, для будь-якої ситуації перехресної реакції, що розглядає більш як два іони, подальша система рівнянь, що утворюється, є складним завданням для будь-якого фахівця з хемометрії [10].

1.6 Штучні нейронні мережі

Штучні нейронні мережі з'явилися під натхненням вивчення принципів роботи людського мозку [15]. ШНМ це математична модель, що здатна змінювати свій стан відносно кінцевої функції. Такі моделі відмінно підходять для розв'язання нелінійних задач, фактично – задач оптимізації, в основі яких лежать нечіткі правила.

Основним елементом ШНМ є вузли (рисунок 1.2) – обчислювальні елементи (ОЕ), що приймають на вхід та подають на вихід числові структури даних, як правило, тензори. Кожен вузол входом та виходом зв'язаний з іншими вузлами або з зовнішнім середовищем, а також містить в собі певну функцію для перетворення даних. Кожен зв'язок

характеризується вагою, з якою пари вузлів збуджуються або гальмуються. Позитивні значення вказують на зв'язки, що збуджують, негативні – на гальмівні [16]. Зв'язки між обчислювальними елементами можуть змінюватися протягом часу, що, власне, і є процесом навчання мережі. Процес навчання є одним з основних механізмів роботи нейронної мережі: під впливом зовнішніх даних, вона змінює свій стан, адаптуючись до середовища. Всі вузли у ШНМ з'єднані з іншими вузлами у своєму оточенні, спосіб встановлення цих зв'язків може відрізнятися між підтипами нейронних мереж. За типами зв'язків можна виокремити наступні ШНМ:

- одношарові перцептрони (Single-layer Perceptrons) – складаються з одного шару вузлів (нейронів), де кожен вхідний сигнал з'єднаний з кожним вихідним через вагові коефіцієнти;

- багатошарові перцептрони (Multi-layer Perceptrons) – мають декілька шарів нейронів (вхідний шар, один або більше прихованих шарів, і вихідний шар);

- рекурентні нейронні мережі (Recurrent Neural Networks) – вид нейронних мереж, де зв'язки між елементами утворюють спрямовану послідовність. Завдяки цьому з'являється можливість обробляти серії подій у часі або послідовні просторові ланцюжки;

- згорткові нейронні мережі (Convolutional Neural Networks) – в таких мережах використовуються згорткові шари, які обробляють локальні області вхідного сигналу;

- радіально-базисні функції нейронних мереж (Radial Basis Function Networks) – використовують радіально-базисні функції як активаційні функції;

- нейронна мережа Хопфілда (Hopfield Networks) – повнозв'язні нейронні мережі, де кожен вузол з'єднаний з кожним іншим вузлом. У процесі роботи динаміка таких мереж сходиться до одного з положень рівноваги. Ці положення рівноваги визначають заздалегідь у процесі

навчання, вони є локальними мінімумами функції, що зветься енергією мережі;

- автокодувальники (Autoencoders) – в таких мережах мережа-кодер виокремлює ключові ознаки вхідних даних, а мережа-декодер приймає ці виокремлені ознаки як вхідні дані;

- генеративно-змагальні мережі (Generative Adversarial Networks) – складаються з двох нейронних мереж: генератора і дискримінатора, які змагаються між собою.

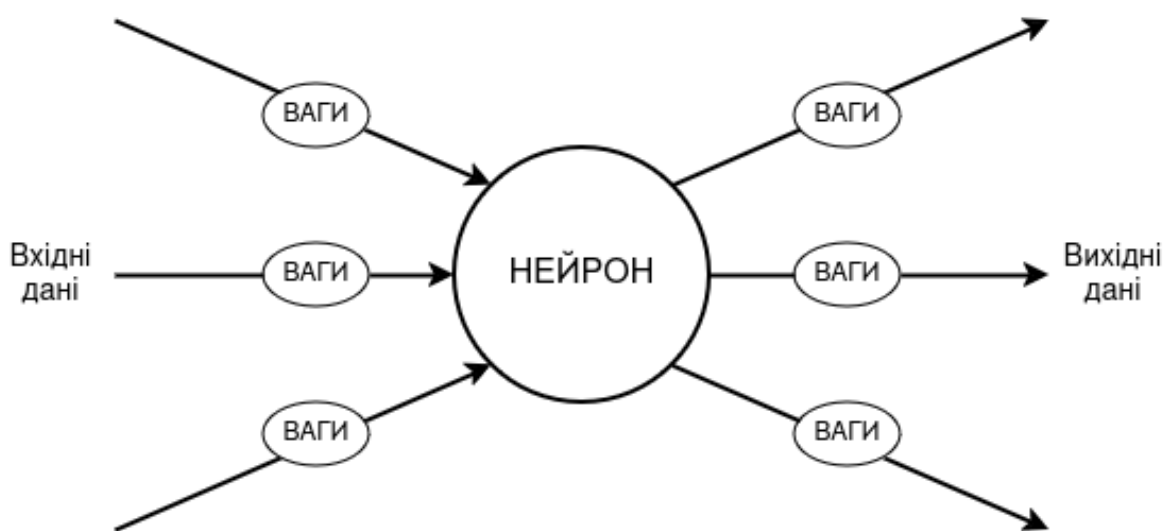


Рисунок 1.2 – Схема вузла нейронної мережі

Рисунок 1.3 демонструє найбільш розповсюджений спосіб розташування вузлів в ШНМ, проте в теорії вони можуть бути організовані будь-яким топологічним способом. Як показано на цьому рисунку, частина вузлів організована у вхідний шар, кількість елементів в ньому залежить від кількості змінних у вхідному векторі даних. Приховані шари ШНМ виконують всі необхідні обчислення, а вихідний шар надає результат; кількість вузлів в цьому шарі також може змінюватися залежно від задачі [17].

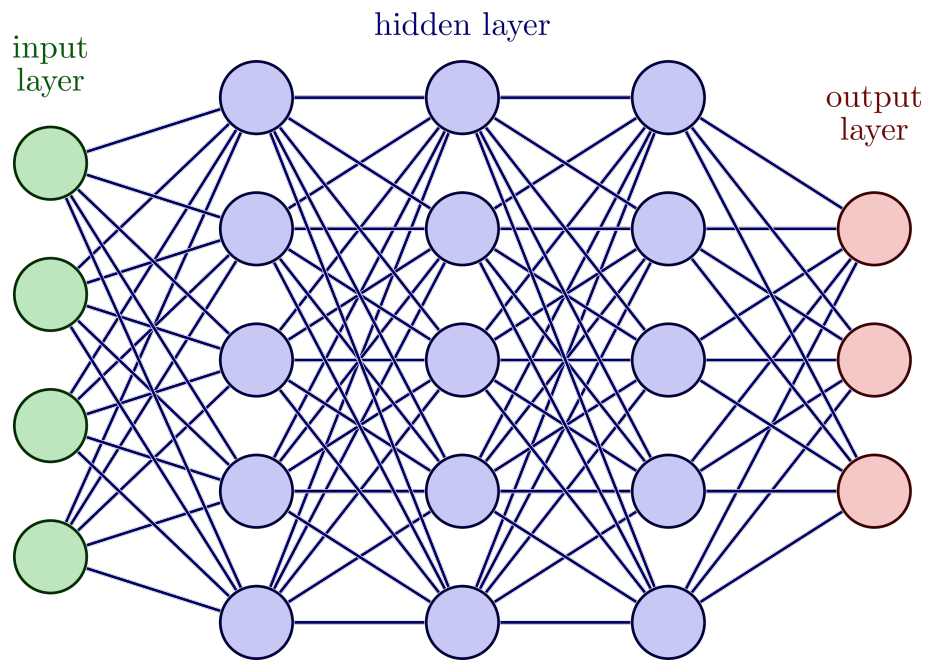


Рисунок 1.3 – Схема нейронної мережі прямого поширення (feed forward topology)

В класичному випадку, коли інформація потрапляє на вхід вузла, вхідний сигнал x_j зі входу j множиться на вагу з'єднання $w_{k,j}$. Таким чином обчислювальний елемент k вираховує зважений сигнал $w_{k,j} \cdot x_j$. Далі він підсумовує вхідні сигнали та додає зміщення θ_k , щоб отримати загальний сигнал y_k :

$$y_k = \sum w_{k,j} \cdot x_j + \theta_k, \quad (1.2)$$

До цього значення, зазвичай, застосовується активаційна функція. Таким чином на виході з вузла буде отримано значення вихідного сигналу a_k . Отже, маємо:

$$a_k = F(y_k), \quad (1.3)$$

де $F()$ – функція активації.

Таким чином повний математичний опис вихідних даних прихованого шару нейронної мережі прямого поширення буде мати вигляд [10]:

$$[X] = F^o\{\theta_j^o + \sum w_{i,j}^o \cdot F^h(\sum w_{i,k}^h \cdot E_k + \theta_k^h)\}, \quad (1.4)$$

де o – означає вихідний шар (output layer);

h – прихований шар (hidden layer);

E_k – вхідна інформація.

1.7 ШНМ при обробці даних сенсорів

Штучні нейронні мережі є досить зручним інструментом для якісного та кількісного аналізу даних, що поступають з сенсорів електронного язика. Якщо опиратися на зазначену раніше інформацію про структуру ШНМ, то простішим варіантом буде подавати на вхід нейронної мережі попередньо опрацьовані дані з язикових хімічних сенсорів, як це схематично зазначено на рисунку 1.4.

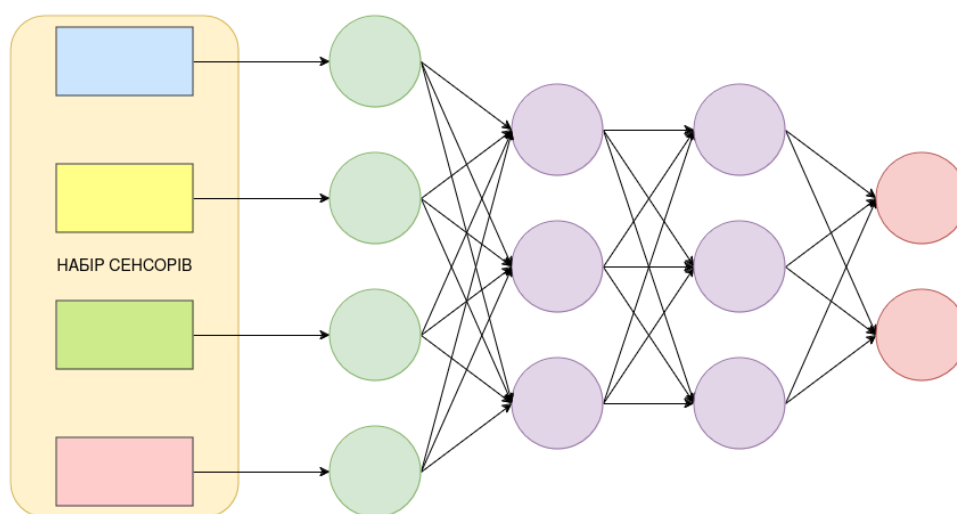


Рисунок 1.4 – Схема роботи електронного язика на основі ШНМ

Оскільки існує нескінченна множина можливих варіантів архітектури ШНМ, не можливо точно відповісти на запитання, як має виглядати готове рішення для штучного язика на основі нейронної мережі. Архітектуру такої нейронної мережі (кількість шарів, активаційні функції, розмірність тензорів, типи шарів тощо) слід підбирати, виходячи з потреб, звертаючи увагу на необхідну точність і швидкодію, а також зважаючи на структуру вхідних даних – даних з сенсорів, оскільки вони можуть являти собою як дискретні значення, так і часові ряди, мати різну розмірність та вигляд.

2 АНАЛІЗ ОПОРНИХ ДАНИХ

Оскільки для відтворення дослідів з використанням електронного язика необхідні лабораторні умови та складне електрохімічне обладнання, було прийняте рішення спиратися на дані вже наявних наукових робіт, що стосуються використання штучних нейронних мереж та машинного навчання для відповідного аналізу. Це надає додаткову можливість для валідації та порівняння результатів практичних досліджень цієї роботи із вже проведеними експериментами.

2.1 Вибір даних. Опорна стаття

За основу в подальшій роботі було обрано статтю «A reconfigurable integrated electronic tongue and its use in accelerated analysis of juices and wines», написану Джанмарком Габріелі, Майклом Мушинські та Патріком В. Рухом.

У цій роботі ручне вилучення ознак з перехідного потенціометричного набору низькоселективних мініатюрних полімерних датчиків поєднано з перетворенням даних для навчання та розгортання моделей машинного навчання на хмарному сервері або периферійному пристрої.

2.2 Методологія опорної статті

Для збору даних дослідники використали набір (рисунок 2.1) потенціометричних датчиків, що складається з 16 провідних полімерних сенсорів, електрополімеризованих на спільній підкладці, що розташовані на власноруч виготовленій поворотній платформі з вісьмома пробірками для зразків і вертикальною віссю для автоматизованого вимірювання рідин (рисунок 2.2). Більш детально установка та датчики описані в статті

«Combining an Integrated Sensor Array with Machine Learning for the Simultaneous Quantification of Multiple Cations in Aqueous Mixtures» [18].

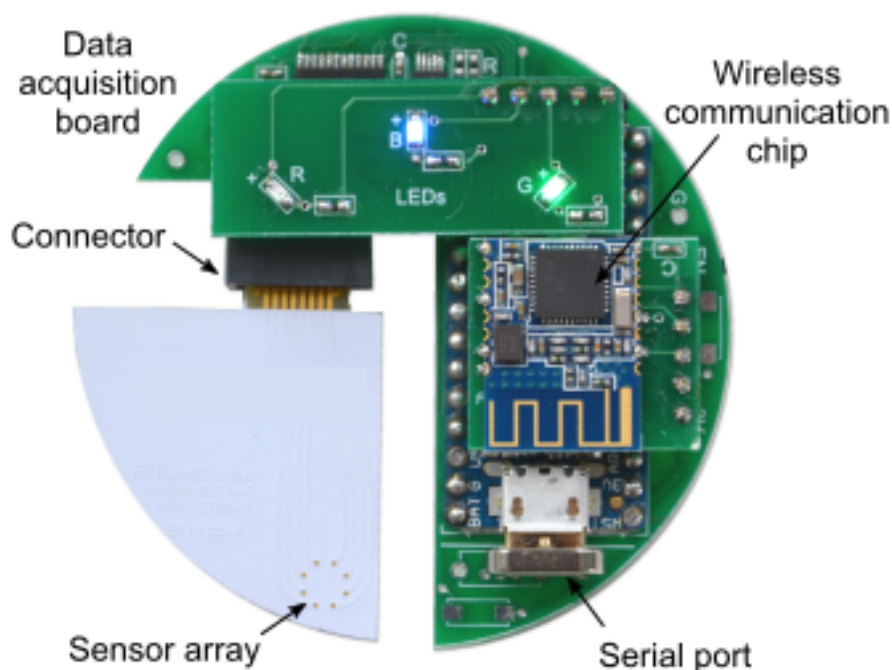


Рисунок 2.1 – Портативний електронний язик з інтегрованим набором датчиків [19]

Дослідники у своїй роботі [19] використовували набір сенсорів для тестування оцтової, лимонної та молочної кислот у п'яти концентраціях між 10^{-5} – 10^{-1} М. Кислотність (рН) розчинів регулювали за допомогою NaOH, щоб підтримувати значення в межах 4,2 – 4,5. Розчин відповідної до 10^{-3} М кислоти використовували як еталонний розчин для порівняння в кожному тесті. Ті ж самі датчики було використано для тестування дев'яти фруктових соків з різними смаками (4 апельсинових, апельсин-маракуйя, грушевий, персиковий, абрикосовий та мультивітамінний), один зі зразків апельсинового соку використовували як еталонну рідину для тестування. Наприкінці масив сенсорів було використано для тестування одинадцяти

пляшок італійського червоного вина, використовуючи одне з вин, Palazzo della Torre, як еталонну рідину.

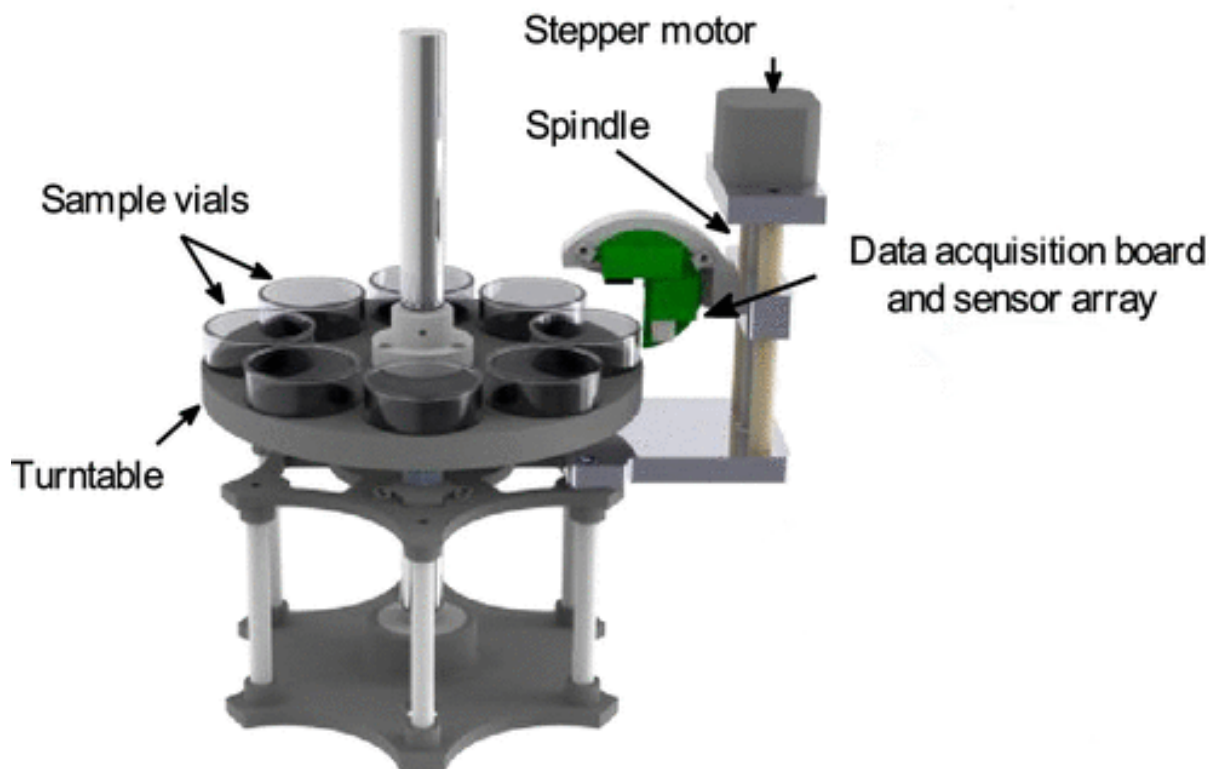


Рисунок 2.2 – Пристрій для дослідження речовин з використанням електронного язика [18]

Загалом 75 дискретних ознак [18], створених вручну, були вилучені з 15 унікальних часових рядів диференціальних напруг під час переходу датчика з відповідної еталонної рідини в тестову. Випробування повторювалися у випадковому порядку з використанням автоматизованої системи відбору зразків [19]. Один апельсиновий сік тестували після зберігання за 40°C протягом 10, 20, 40 або 50 днів, порівнюючи його з тим самим соком, який зберігали за кімнатної температури або в холодильнику, при цьому останній також використовували як еталонну рідину під час тестування [19]. Усі рідини тестували за кімнатної

температури. Усі завдання класифікації розв’язували за допомогою трьох різних алгоритмів машинного навчання: лінійний дискримінантний аналіз (LDA), K-Nearest Neighbors (KNN з $k = 3$) і ансамбль із 50 bagged decision trees [19].

2.3 Дані вин для аналізу

Обрані для пересічної валідації результатів досліджень дані представлені у вигляді часових рядів, які утворюються в результаті вимірювання різних видів пляшкових вин за допомогою 16 електрохімічних датчиків. Детальні відомості про досліджувальні дані зазначені в таблиці 2.1.

Таблиця 2.1 – Особливості даних з сенсорів

Кількість електрохімічних датчиків	16
Кількість зразків винних напоїв	7
Загальна кількість вимірювань	68
Час вимірювання референтного зразка (с)	20
Загальний час одного вимірювання (с)	80

Перші 20 секунд сенсори занурюються в референтний вид вина, після чого переміщується в тестовий. Подібне дослідження проводиться декілька разів для кожного зразка речовини. Дані з сенсорів є сирими вхідними даними системи штучного інтелекту.

3 МЕТОДОЛОГІЯ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

3.1 Робота з часовими рядами

Оскільки надані для аналізу дані являють собою часові ряди, перш ніж використовувати щодо них різні методи класифікації, слід здійснити попередню обробку. При тому, що існує безліч різних підходів для класифікації часових рядів, в певних окремих випадках слід використовувати свої способи обробки даних та свої способи класифікації. Найбільш розповсюдженими способами класифікації часових рядів є [20]:

- дистанційні підходи: подібні алгоритми побудовані на мірі відстані – відстані між вимірюваннями у числовому просторі тощо. Існує багато алгоритмів, що використовують відстань для класифікації, такі як KNN, SVM та інші;

- shapelet (класифікатори за формою даних): дані числових рядів часто мають певну репрезентативну форму, що вказує на клас цих рядів;

- глибоке навчання: моделі глибокого навчання – це тип нейронної мережі, яка має кілька шарів нейронів або перцептронів. Ці моделі, як правило, набагато складніші з набагато більшою кількістю параметрів, ніж інші типи алгоритмічних моделей. Початкові рівні мережі глибокого навчання кодують фундаментальні форми в даних часових рядів, тоді як останні шари кодують уявлення, які потім можна розділити на класи на останньому рівні;

- модельні ансамблі – це набори моделей класифікації, кожна з яких виконує власну класифікацію набору даних. Щоб цей підхід працював, будь-які помилки, створені класифікаторами, не повинні корелюватися;

- інтервальні підходи: методи на основі інтервалів базуються на поділі часових рядів на окремі інтервали, після чого кожен інтервал

використовується для навчання окремої моделі машинного навчання (класифікатора). Цей підхід створює ансамбль класифікаторів, кожен з яких діє на власну підпоследовність або інтервал. Остаточна класифікація визначається на основі найпоширенішого класу, який генерується окремими класифікаторами.

3.2 Попередня обробка даних

3.2.1 Розгортання матриць

Представлені дані не являють собою часові ряди у звичному сенсі, оскільки вони збираються в декілька підходів. Фактично, результатом збору даних є окремі екземпляри, що самі представлені у вигляді специфічних часових рядів. Вигляд цих числових рядів є достатньо уніфікованим, адже є можливість точно закодувати час, за який система збирає заміри у референтному та тестовому зразках. Таким чином дані можна розглядати і як часовий ряд, і як одновимірний вектор ознак, де часова природа даних перестає грати роль.

Виходячи з цього, оптимальним способом попередньої обробки даних було розвернути матриці числових рядів в один вектор, як це схематично зображено на рисунку 3.1. При цьому попередньо для кожного виду вина прораховується значення з кожного сенсора відносно значень референтного виду вина. Для цього спочатку за формулою 3.1 обчислюється середнє значення сенсора у референтній рідині, а після обчислюються відносні значення за формулою 3.2.

$$mean^{ref} = \frac{1}{n^{ref}} \sum_{i=0}^{n^{ref}} x_i, \quad (3.1)$$

$$X^{ref} = x_j - mean^{ref}, \quad j \in (n^{ref}, N), \quad (3.2)$$

де x – значення з сенсора в певний час;

n^{ref} – кількість значень, що стосуються референтного зразка (кількість секунд, що сенсор перебуває в референтній рідині);

X^{ref} – новий утворений вектор відносних значень.

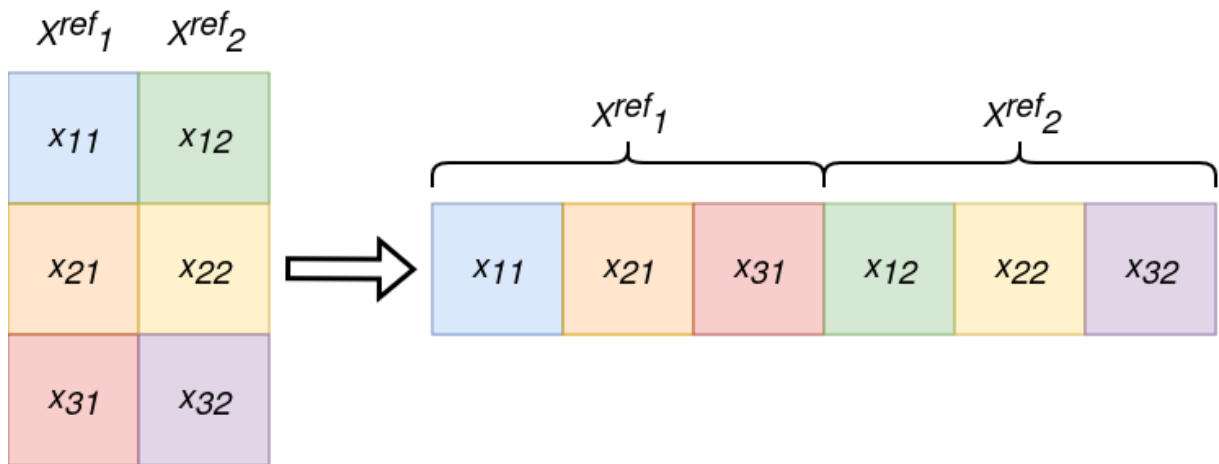


Рисунок 3.1 – Схема попереднього перетворення даних

3.2.2 Перетворення часових рядів на картинки

Одним із найбільш розповсюджених способів конвертації часових рядів та їхньої підготовки для класифікації є перетворення даних на графічні зображення. Існує багато алгоритмів для подібних операцій, основні з них:

- Gramian angular Field (GAF) – метод перетворює часовий ряд у матрицю, використовуючи тригонометричні перетворення. Ідея полягає в кодуванні часової інформації в кутовий простір, де кожна точка часу перетворюється у кут, а потім формується матриця кутів;

- Markov transition field (MTF) – метод створює матрицю

перехідних ймовірностей між різними станами часового ряду. Це дозволяє візуалізувати перехідну динаміку часового ряду;

– recurrence plot (RP) – метод візуалізує точки, коли часовий ряд повертається до попереднього стану. Це робиться шляхом побудови матриці, де кожна точка представляє момент, коли значення часового ряду на двох різних часових кроках дуже близькі;

– spectrogram – спектрограма використовує перетворення Фур'є, щоб показати, як частотний склад сигналу змінюється з часом. Це особливо корисно для аналізу сигналів, які мають явну частотну структуру;

– wavelet transform – дозволяє аналізувати часовий ряд на різних часових масштабах;

– recurrence quantification analysis (RQA) – метод базується на побудові рекурентних діаграм і аналізі різних метричних характеристик, таких як діагональні лінії або вертикальні структури, щоб дати характеристику динаміці часового ряду.

Більш детально слід зупинитися на перших двох методах перетворення.

3.2.2.1 Gramian Angular field

Gramian Angular Field (GAF) є методом, що використовується для перетворення часових рядів у двовимірні зображення. Цей метод застосовується у машинному навчанні та аналізі даних, особливо в задачах класифікації та розпізнавання образів, де обробка двовимірних даних може бути більш ефективною, ніж робота з одномірними часовими рядами. У Gramian Angular field (рисунок 3.2) часові ряди представляються в полярній системі координат замість типових декартових координат. Фактично в подібній матриці кожен елемент є косинусом суми кутів.

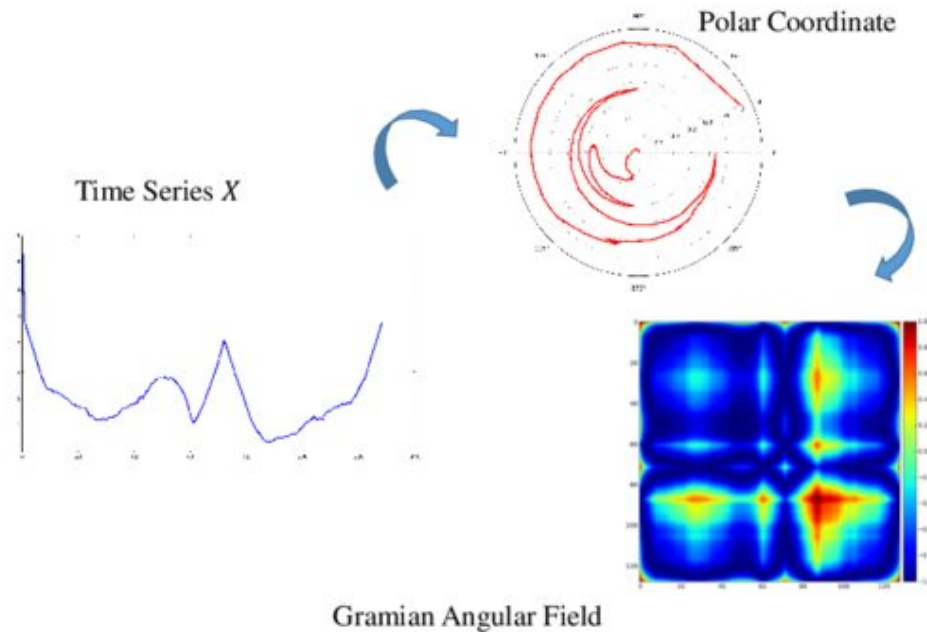


Рисунок 3.2 – Схема роботи GAF [22]

Якщо представляти це математично, отримаємо наступне [21]: для заданого часового ряду $X = \{x_1, x_2, \dots, x_n\}$, що складається зі n спостережень, ми нормалізуємо X так, щоб всі значення вектора потрапляли в інтервал $[-1, 1]$; таким чином ми можемо представити отриманий під час нормалізації вектор \tilde{X} у полярних координатах, закодувавши значення як кутовий косинус і мітку часу як радіус за допомогою рівняння 3.3 [21].

$$\begin{cases} \phi = \arccos(\tilde{x}_i), & -1 \leq \tilde{x}_i \leq 1, \tilde{x}_i \in \tilde{X} \\ r = \frac{t_i}{N}, & t_i \in \mathbb{N} \end{cases}, \quad (3.3)$$

де t_i – часова мітка;

N – постійний коефіцієнт для впорядкування розмаху полярної системи координат.

Після перетворення нормованого часового ряду в полярну систему координат, використовується кутова перспектива: розглядається

тригонометрична сума між кожною точкою, щоб визначити часову кореляцію в різних часових інтервалах. таким чином GAF визначається як [21]:

$$G = \begin{bmatrix} \cos(\phi_1 + \phi_2) & \cdots & \cos(\phi_1 + \phi_n) \\ \cos(\phi_2 + \phi_1) & \cdots & \cos(\phi_2 + \phi_n) \\ \vdots & \ddots & \vdots \\ \cos(\phi_n + \phi_1) & \cdots & \cos(\phi_n + \phi_n) \end{bmatrix} = \quad (3.4)$$

$$= \tilde{X}' \cdot \tilde{X} - \sqrt{I - \tilde{X}^2}' \cdot \sqrt{I - \tilde{X}^2},$$

де I – вектор-рядок одиниць $[1, 1, \dots, 1]$.

Після переходу до полярної системи координат ми розглядаємо часові ряди на кожному часовому кроці як одновимірний простір. Визначаючи внутрішній добуток $\langle x, y \rangle = x \cdot y - \sqrt{1 - x^2} \cdot \sqrt{1 - y^2}$, отримуємо G , що є Gramian матрицею [21]:

$$\begin{bmatrix} \langle \tilde{x}_1, \tilde{x}_1 \rangle & \cdots & \langle \tilde{x}_1, \tilde{x}_n \rangle \\ \langle \tilde{x}_2, \tilde{x}_1 \rangle & \cdots & \langle \tilde{x}_2, \tilde{x}_n \rangle \\ \vdots & \ddots & \vdots \\ \langle \tilde{x}_n, \tilde{x}_1 \rangle & \cdots & \langle \tilde{x}_n, \tilde{x}_n \rangle \end{bmatrix}. \quad (3.5)$$

Існують два основних типи GAF [23]:

- Gramian angular summation field (GASF): використовує суму кутів $\phi_i + \phi_j$;
- Gramian angular difference field (GADF): використовує різницю кутів $\phi_i - \phi_j$.

GAF має беззаперечну перевагу: він дає змогу зберегти часову

залежність, оскільки час збільшується, коли позиція переміщується з лівого верхнього кута в правий нижній. Також подібне перетворення дає змогу використати механізми комп'ютерного зору для аналізу часових рядів.

З мінусів можна визначити те, що вихідна матриця має досить велику розмірність, і для її зменшення доводиться використовувати певні апроксимації.

GAF є потужним інструментом для аналізу часових рядів і може значно покращити результати у завданнях, пов'язаних з класифікацією, розпізнаванням образів та прогнозуванням.

3.2.2.2 Markov Transition Field

Марковське поле переходів (Markov Transition Field, MTF) – це метод перетворення часових рядів у матриці (рисунок 3.3), які можна використовувати для подальшої обробки за допомогою алгоритмів машинного навчання та аналізу зображень.

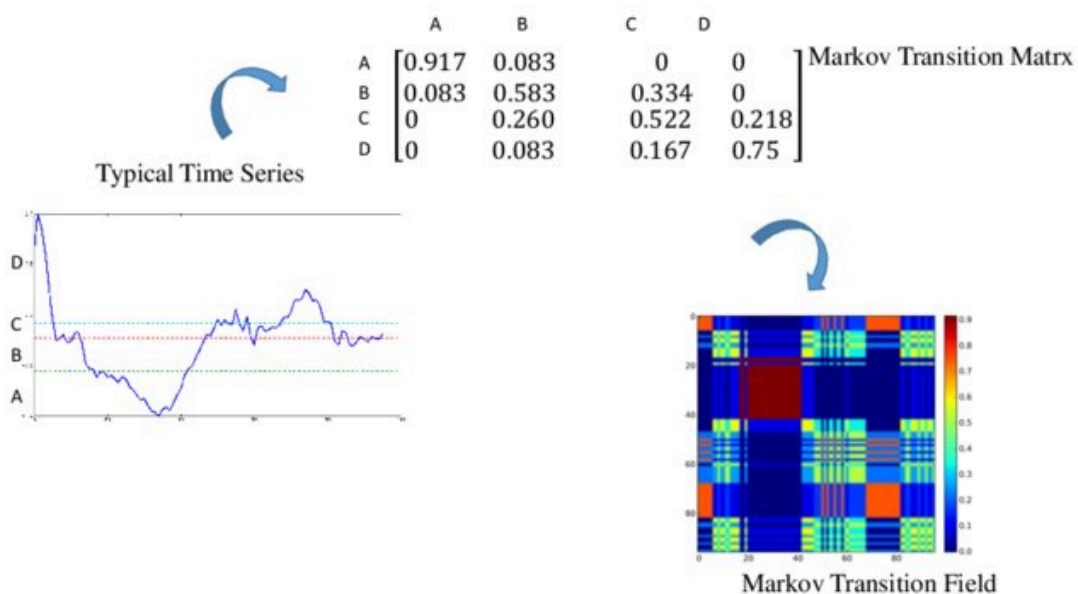


Рисунок 3.3 – Схема роботи алгоритму MTF [22]

Ідея полягає в тому, щоб закодувати часовий ряд у вигляді двовимірної матриці, що зберігає інформацію про перехідні ймовірності між різними станами ряду. Markov Transition Field зберігає інформацію про часову область, представляючи ймовірності марковських переходів.

Для імплементації методу MTF, як і для алгоритму Gramian Angular field, спочатку проводиться нормалізація даних:

$$x' = \frac{x - x_{mean}}{x_{max} - x_{min}}, \quad x \in \{x_1, x_2, \dots, x_n\}. \quad (3.6)$$

Далі нормалізований часовий ряд розбивається на Q бінів, де кожен x_i відповідає біну q_i ($j \in [1, Q]$). Використовуючи ланцюг Маркова першого порядку для підрахунку переходів між бінами квантилей, ми отримуємо $Q \times Q$ зважену матрицю суміжності w . $w_{i,j}$ являє собою частоту, з якою за точкою у квантилі q_j слідує точка у квантилі q_i . Формула розрахунку для $w_{i,j}$ має наступний вигляд [24]:

$$w_{i,j} = \sum_{\forall x \in q_i, y \in q_j, x+1=y} 1 / \sum_{j=1}^Q w_{i,j}. \quad (3.7)$$

Останнім кроком для отримання MTF матриці M конструюється поле переходів маркова [24]:

$$M = \begin{bmatrix} w_{ij|x_1 \in q_i, x_1 \in q_j} & \cdots & w_{ij|x_1 \in q_i, x_n \in q_j} \\ \vdots & \ddots & \vdots \\ w_{ij|x_n \in q_i, x_1 \in q_j} & \cdots & w_{ij|x_n \in q_i, x_n \in q_j} \end{bmatrix}, \quad (3.8)$$

де $M_{i,j}$ позначає ймовірність переходу q_i в q_j .

Можна виокремити наступні переваги MTF:

– збереження часової інформації: на відміну від інших методів, таких як перетворення Фур’є, МТФ зберігає інформацію про часову природу послідовності елементів ряду;

– універсальність: МТФ може використовуватися для різних типів часових рядів і легко інтегрувати з алгоритмами глибокого навчання для аналізу зображень.

МТФ чудово підходить для застосування при аналізі часових рядів, обробці комплексних сигналів тощо.

3.3 Попередня обробка наявних даних

Як вже було зазначено вище, перш за все, для попередньої обробки даних слід сформувати матрицю відносних значень. Далі було прийнято рішення провести декілька досліджень даних. Для цього було зроблено копію матриці та розгорнуто її у векторні значення. Подібним перетворенням відповідає псевдокод, що наведено у лістингу 3.1.

Далі знову робиться дві копії даних, після чого до обох застосовуються алгоритми перетворення часових рядів на зображення, що розглядалися раніше. Таким чином ми отримуємо графічну репрезентацію часових рядів з використанням перетворення МТФ, що відображає рисунок 3.4.

Лістинг 3.1 – Псевдокод обробки даних з розгортанням матриці

```
for each frame in experiment_results
  for each column_name, column in frame
    relative_signal[column_name] = column[20:80]
  - mean(column[0:20])
  return unfold(relative_signal[column_name])
```

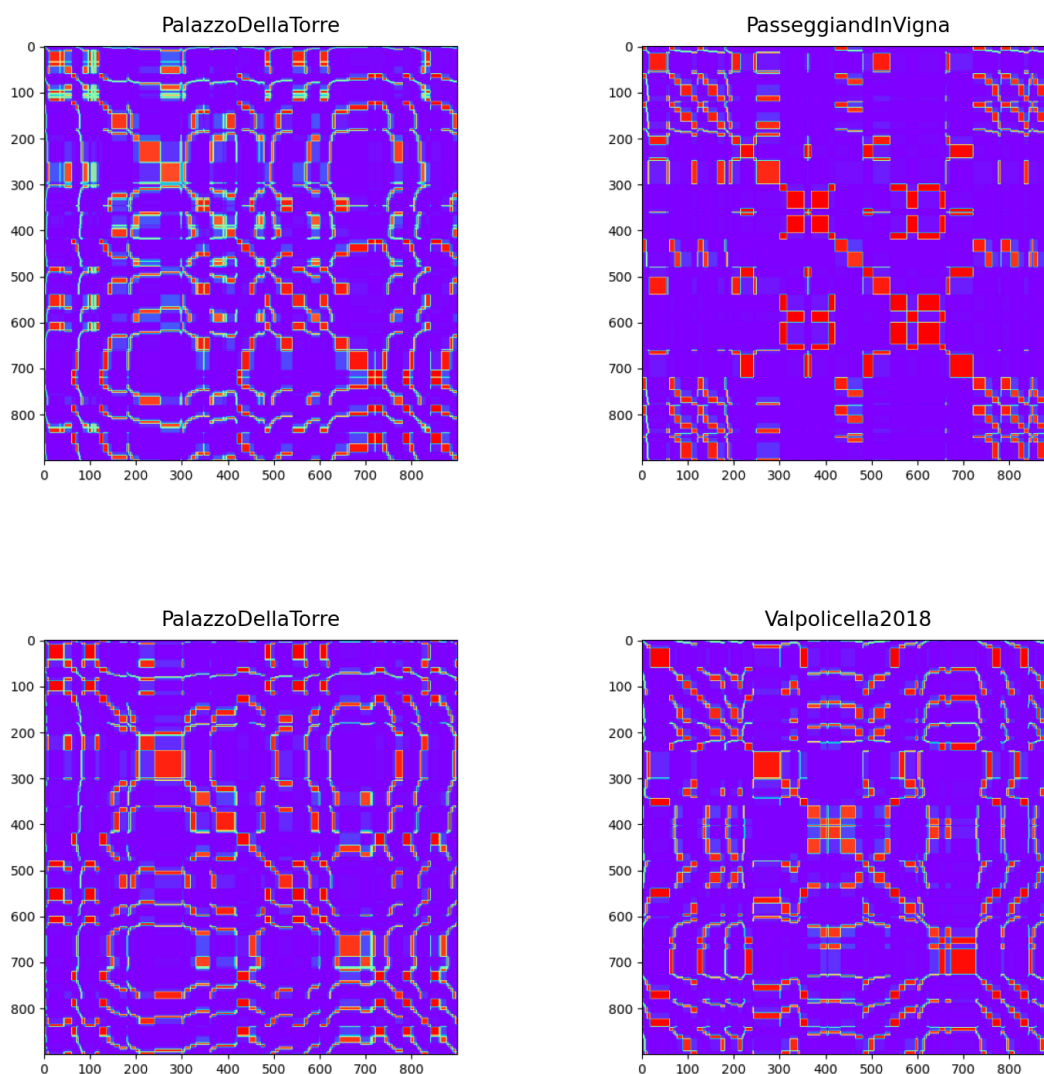


Рисунок 3.4 – Відображення зразків вин, що представлені розгорнутою матрицею, до якої було застосовано перетворення MTF

Відповідне перетворення відбувається з використанням GAF, що зображено на рисунку 3.5.

Проте, щоб дослідити реальний стан даних як часових рядів, слід уникнути розгортання матриці відносних значень, тоді попередня обробка зводиться до побудови матриці відносних значень, що можна описати псевдокодом, наведеним у лістингу 3.2.

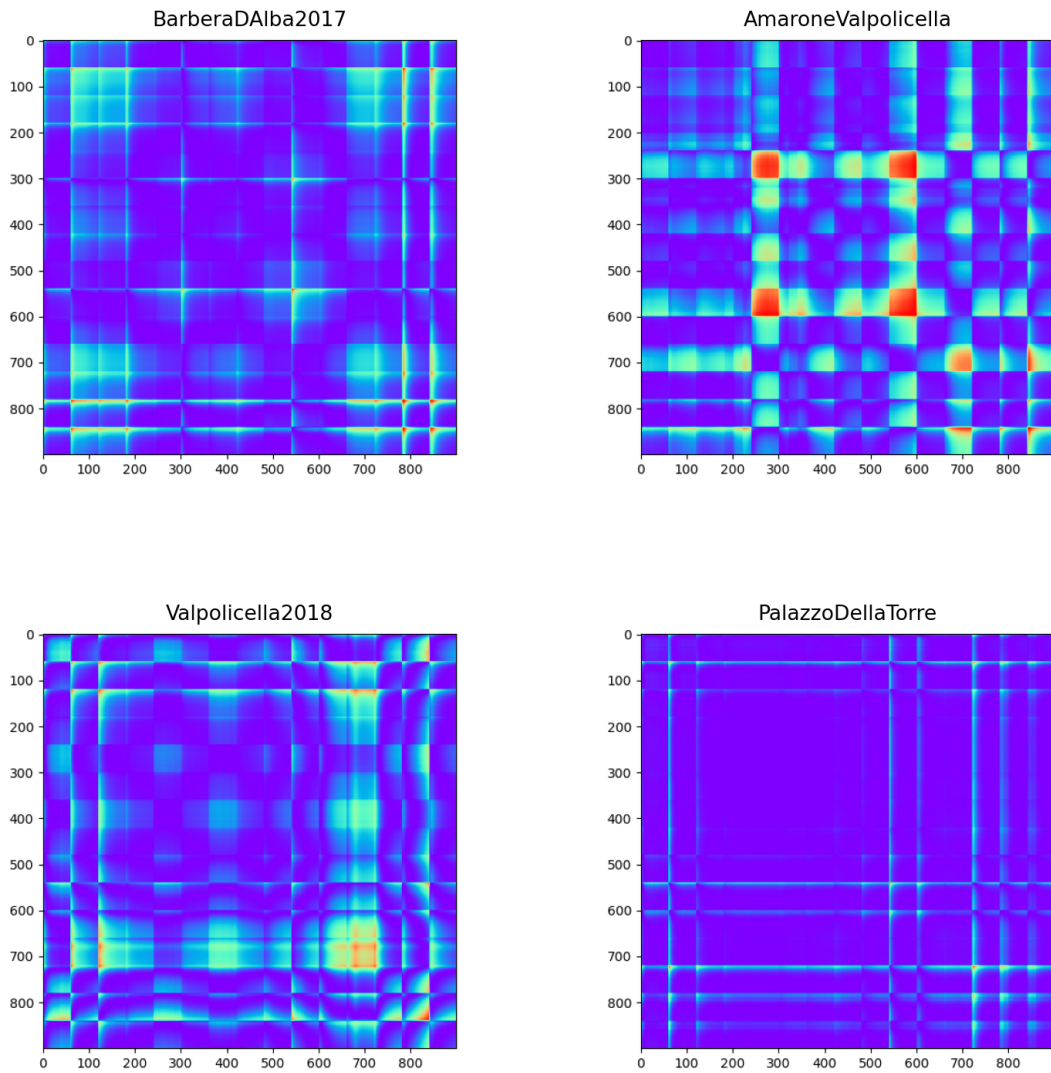


Рисунок 3.5 – Відображення зразків вин, що представлені розгорнутою матрицею, до якої було застосовано перетворення GAF

Застосувавши до цієї матриці алгоритми перетворення MTF та GAF, отримаємо результати, що зображені на рисунках 3.6–3.7.

Лістинг 3.2 – Псевдокод обробки даних без розгортання матриці

```

for each frame in experiment_results
  for each column_name, column in frame
    relative_signal[column_name] = column[20:80]
    - mean(column[0:20])
  return relative_signal

```

Як можна побачити на рисунках 3.6–3.7, вихідні зображення при різних методах перетворення даних значно відрізняються між собою. Це означає, що застосовані до цих даних класифікатори, можуть дати досить відмінні результати.

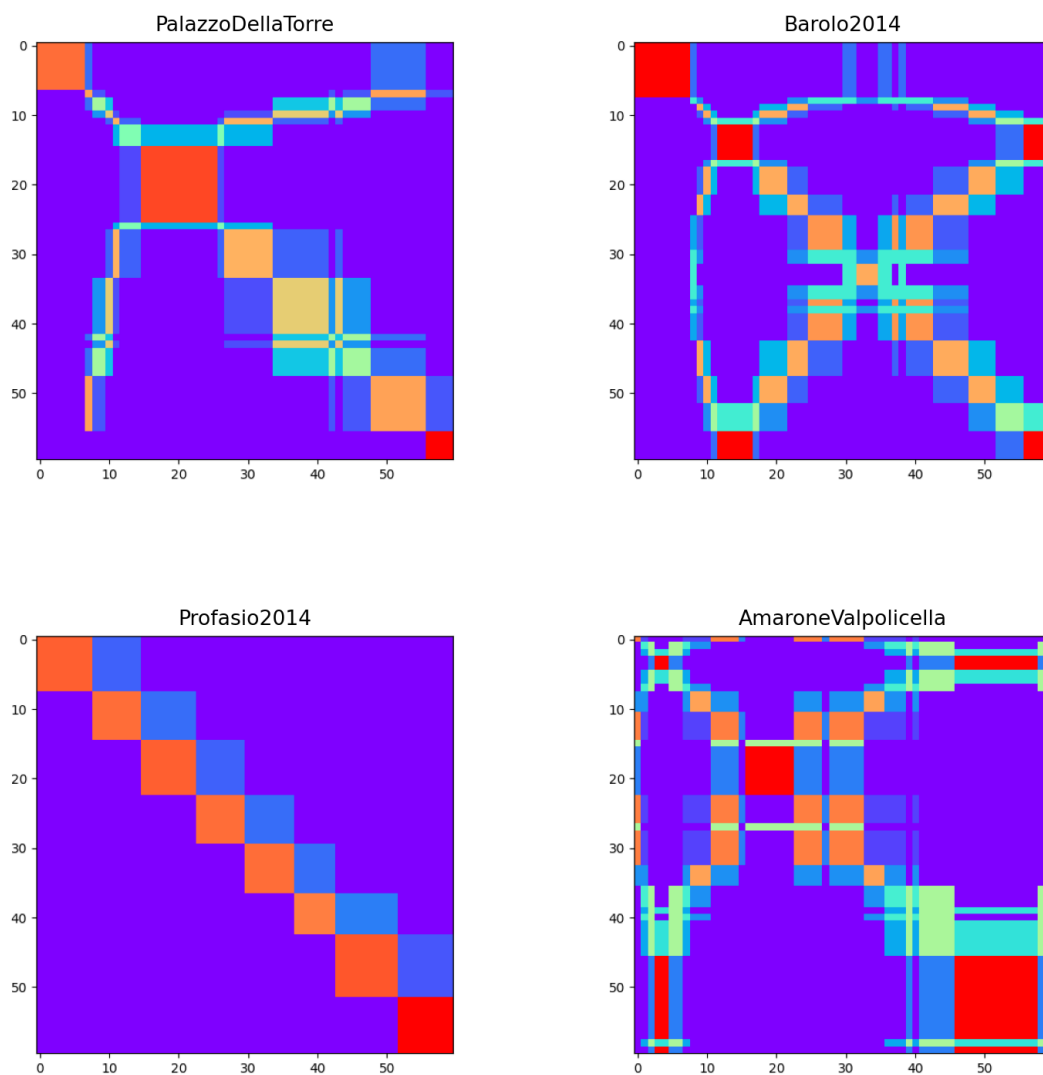


Рисунок 3.6 – Відображення зразків вин, що представлені нерозгорнутою матрицею, до якої було застосовано перетворення MTF

Отримані зображення зразків винних напоїв можна використовувати для подальшого аналізу, вилучення ознак та для роботи з даними, застосовуючи методи машинного зору.

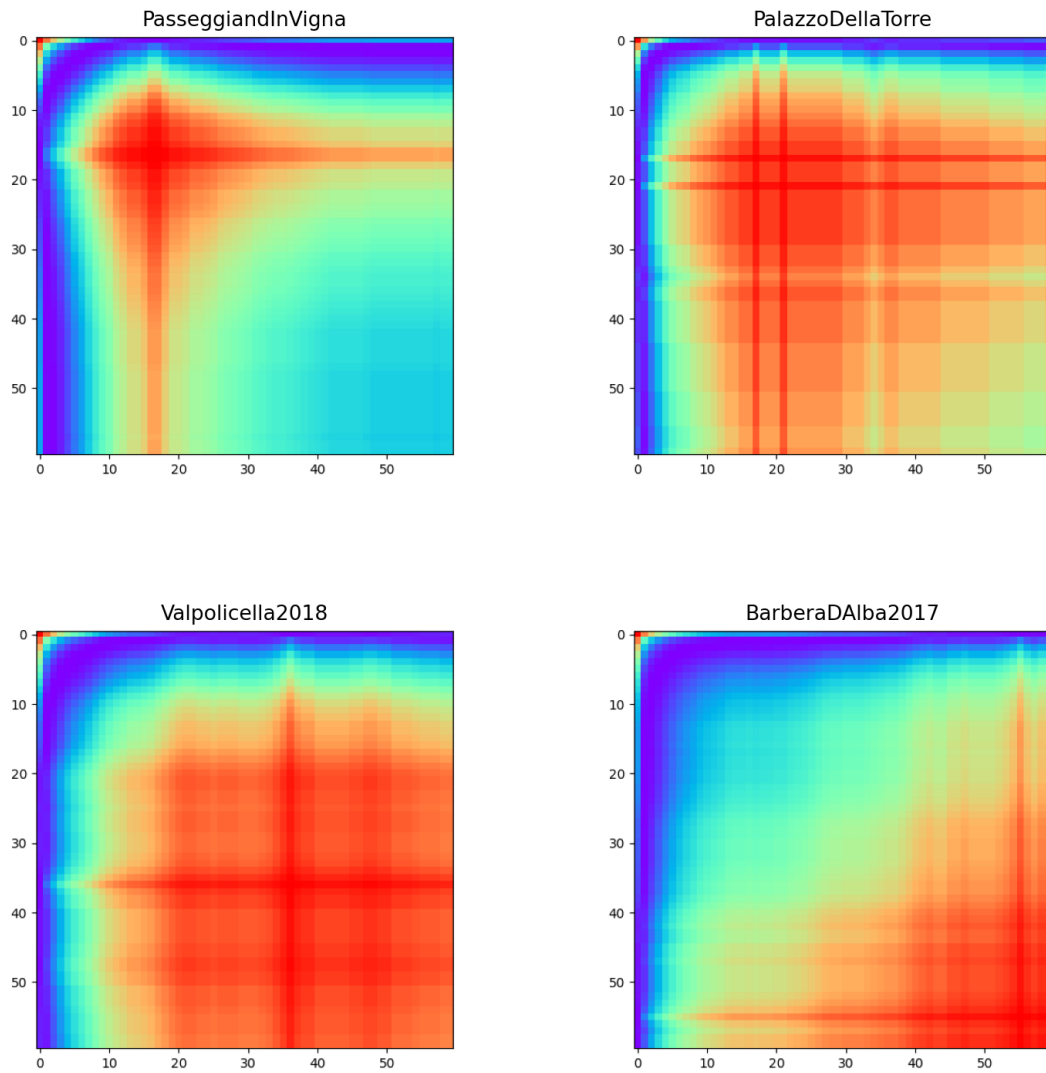


Рисунок 3.7 – Відображення зразків вин, що представлені нерозгорнутою матрицею, до якої було застосовано перетворення GAF

3.4 Класифікація зображень

Класифікація зображень – це фундаментальний аспект комп’ютерного зору, що передбачає автоматичне розподілення зображень на заздалегідь визначені класи на основі їх візуального змісту. Це завдання має широке застосування, включаючи розпізнавання об’єктів, облич, сцен, медичних зображень та інших візуальних даних.

До появи глибоких нейронних мереж, на початкових етапах розвитку класифікації зображень, дослідники та інженери намагалися

створити системи, засновані на правилах, де чіткі інструкції керували категоризацією зображень [25]. Проте такий підхід зазнав значних невдач при застосуванні до реальних даних – складність і мінливість, притаманні зображенням, часто перевищують можливості попередньо визначених правил, що призводить до обмеженої адаптивності та можливості узагальнення. Зазвичай класифікація зображень відбувалася шляхом «ручного» вилучення ознак із зображення. Для цього використовувалися такі алгоритми Scale Invariant Feature Transform (SIFT) [26], histograms of oriented gradients (HOG) [27] тощо. Після чого для класифікації застосовувалися звичайні методи машинного навчання.

Проте, поява алгоритмів глибокого навчання [28], зокрема CNN, змінила підходи до класифікації зображень [29], дозволивши автоматично виділяти релевантні ознаки.

3.4.1 Глибоке навчання у класифікації зображень

На передовій глибокого навчання для класифікації зображень знаходиться широке використання згорткових нейронних мереж [28]. Ці мережі показали свою ефективність у подоланні обмежень традиційних методів, завдяки впровадженню ієрархічного та автоматизованого підходу до виділення ознак. CNN складаються з шарів, що систематично вивчають патерни та ознаки на різних рівнях абстракції, імітуючи роботу зорової системи людини. Згорткові шари дозволяють мережам захоплювати локальні патерни на зображеннях, що дозволяє їм розпізнавати складні візуальні ієрархії, які важко було б виділити вручну традиційними методами. Ключова особливість глибокого навчання, особливо для класифікації зображень, – це його здатність автоматично вивчати та представляти ознаки з необроблених даних.

Також підходи глибокого навчання до класифікації зображень значно просунулися вперед при впровадженні передавального

навчання (transfer learning) [30]. такий підхід передбачає попереднє навчання моделі на великому обсязі даних для спорідненої задачі, перед її точним налаштуванням для конкретної цільової. Передавальне навчання виявилось особливо корисним у ситуаціях, де отримання великих наборів розмічених даних є складним завданням.

3.4.1.1 Convolutional neural networks

CNN – це архітектура глибокої нейронної мережі, що належить до методології керованого навчання. Першочергово вона була розроблена для автоматичного вивчення особливостей зображень за допомогою декількох шарів згорток і масштабування. Хоча зазвичай згорткові мережі використовуються для зображень, вони можуть бути застосовані до будь-якого типу даних, де всі ознаки мають однорідну семантику [31].

Основний принцип згорткової нейронної мережі це саме згортка [32]. Зазвичай середньозважене значення сусідства певної ознаки часто містить більше інформації, ніж одна ознака сама по собі, то ж згортка полягає в застосуванні фільтрів (або ядер) до вхідних даних, вони ковзають по вхідним даним зображення та обчислюють скалярний добуток між вагою ядра і відповідною ділянкою вхідного зображення.

Стандартна архітектура CNN дуже схожа на звичайну нейронну мережу, але при цьому вона обов'язково має в собі набір згорткових, пулінгових та повнозв'язних шарів:

- згорткові шари застосовують кілька фільтрів для виявлення локальних ознак;
- пулінгові шари зменшують розмір просторового представлення даних;
- повнозв'язні шари обчислюють фінальне значення класифікації або регресії.

3.4.1.2 Vision transformers

Поява Vision Transformer (ViT) стала поворотним моментом у галузі комп'ютерного зору. Згорткові нейронні мережі (CNN), що домінували в області класифікації зображень, почали змінюватись з появою ViT – моделі, яка використовує архітектуру трансформера, спочатку призначену для обробки природної мови. Цей інноваційний підхід до класифікації зображень демонструє, як чистий трансформер, застосований безпосередньо до послідовностей фрагментів зображень, може перевершити традиційні ШНМ на основних тестах, таких як ImageNet, навіть зі значно меншими обчислювальними ресурсами [35].

Vision Transformer працює, розбиваючи вхідне зображення на сітку фрагментів, розглядаючи кожен фрагмент як маркер, подібний до слів у реченні. Потім ці токени вбудовуються разом з позиційними вбудовуваннями, щоб зберегти просторову інформацію, яка є вирішальним компонентом для розуміння зображення в цілому. Ядро трансформаційної моделі, стандартний трансформаційний кодер, використовує механізм уваги для зв'язку різних частин зображення, що дозволяє моделі робити прогнози на основі глобальної інформації, а не зосереджуватися на локальних особливостях, як це робить CNN.

3.5 Класифікація графічного відображення винних напоїв

Для подальшого дослідження отриманих раніше перетворених відображень винних напоїв було прийнято рішення використати попередньо натреновані моделі штучних нейронних мереж різних типів, а саме зоровий трансформер та згорткову нейронну мережу. Це дозволить оцінити можливості цих моделей, а також порівняти ефективність моделей на реальних даних.

Для тестування було обрано наступні попередньо натреновані

моделі: згорткові залишкові нейронні мережі (residual neural network, ResNet) [34], а саме ResNet50 та ResNet101, що відрізняються кількістю шарів, що складає 50 та 100 шарів відповідно, та зорові трансформери [35], а саме ViT-Base-Patch16-224 та ViT-Large-Patch16-224, що відрізняються кількістю параметрів, шарів, головок та загальним розміром моделі.

За допомогою обраних моделей було витягнуто ознаки з підготовлених та розподілених по класах (видам вин) картинок, дані були збережені в форматі CSV для подальшого навчання класифікаторів.

Як класифікатори було прийнято рішення використовувати декілька класичних алгоритмів машинного навчання, а саме: Gaussian Naive Bayes classifier [36], k-Nearest Neighbour classifier [37], Stochastic Gradient Descent classifier [38], XGBoost classifier [39].

Окремо було створено глибоку нейронну мережу, що також навчалася на витягнутих з картинок даних. Дана нейронна мережа має архітектуру, що зображена на рисунку 3.8.

Після навчання всіх класифікаторів було отримано фінальні дані, що відповідають точності класифікаторів для різних вхідних даних та їхніми confusion матрицями. Загалом було проведено різні експерименти, що відрізняються за:

- чи представлена початкова відносна матриця у розгорнутому вигляді, чи ні;
- який перетворювач часових рядів у картинку було використано: MTF або GAF;
- яка модель вилучення ознак була залучена: ResNet-50, ResNet-101, ViT-Base-Patch16-224 чи ViT-Large-Patch16-224;
- який класифікатор було використано: NB, KNN, SGD, XGBoost чи нейронна мережа.

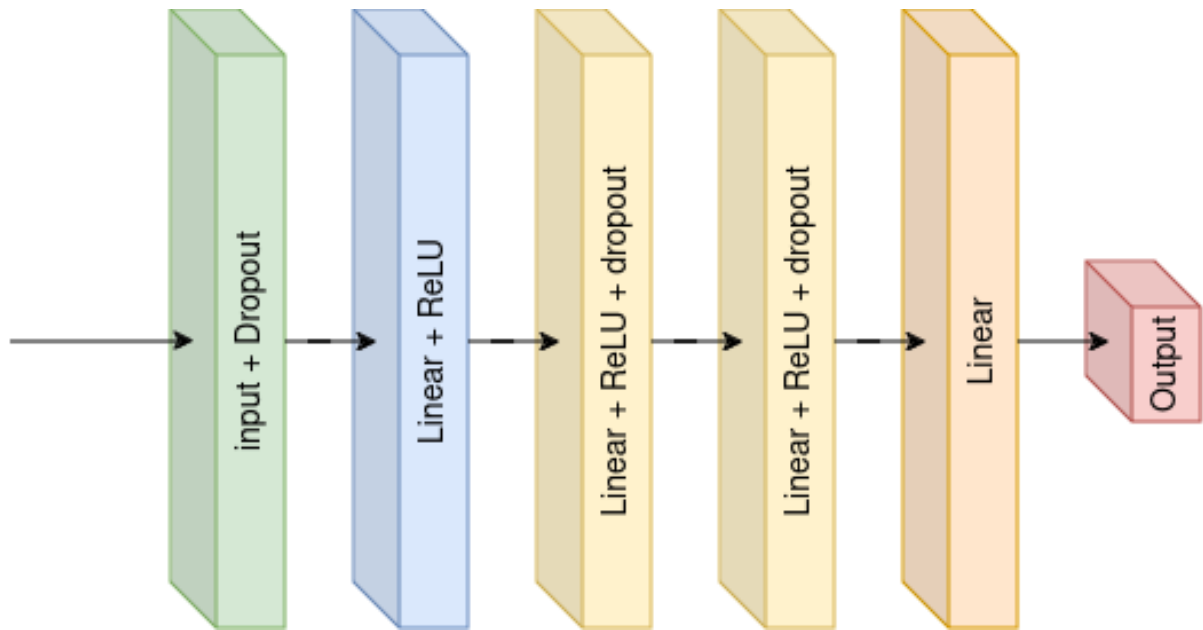


Рисунок 3.8 – Архітектура нейронної мережі для класифікації

3.6 Результати досліджень

Нижче будуть наведені результати досліджень в таблицях та рисунки матриць плутанини (confusion matrix).

3.6.1 Результати для нерозгорнутих матриць

Як можна побачити в таблиці 3.1, класифікатори, що навчені на нерозгорнутих перетворених за допомогою MTF даних демонструють досить пересічні результати. Всі моделі вилучення ознак показують приблизно однакові результати, однак ResNet-50 виділяється в кращу сторону. Набагато ліпші результати демонструє штучна нейронна мережа, там точність класифікації в більшості випадків досягає 100 відсотків. Помилки, що виникли при поточній класифікації, можна побачити на рисунках 3.9–3.13.

Таблиця 3.1 – Результати для нерозгорнутих матриць при використанні MTF

Model	Classifier	Accuracy	F1 Score
ResNet-50	Gaussian Naive Bayes	0.643	0.681
	KNN	0.929	0.933
	Stochastic Gradient Descent	0.286	0.443
	XGBoost	0.929	0.933
	Neural Network	1	1
ResNet-101	Gaussian Naive Bayes	0.500	0.571
	KNN	0.786	0.795
	Stochastic Gradient Descent	0.143	0.238
	XGBoost	0.500	0.543
	Neural Network	0.86	0.85
ViT-Base-224	Gaussian Naive Bayes	0.500	0.471
	KNN	0.857	0.867
	Stochastic Gradient Descent	0.357	0.474
	XGBoost	0.714	0.733
	Neural Network	1	1
ViT-Large-224	Gaussian Naive Bayes	0.357	0.348
	KNN	0.857	0.867
	Stochastic Gradient Descent	0.143	0.250
	XGBoost	0.786	0.781
	Neural Network	1	1

resnet_50

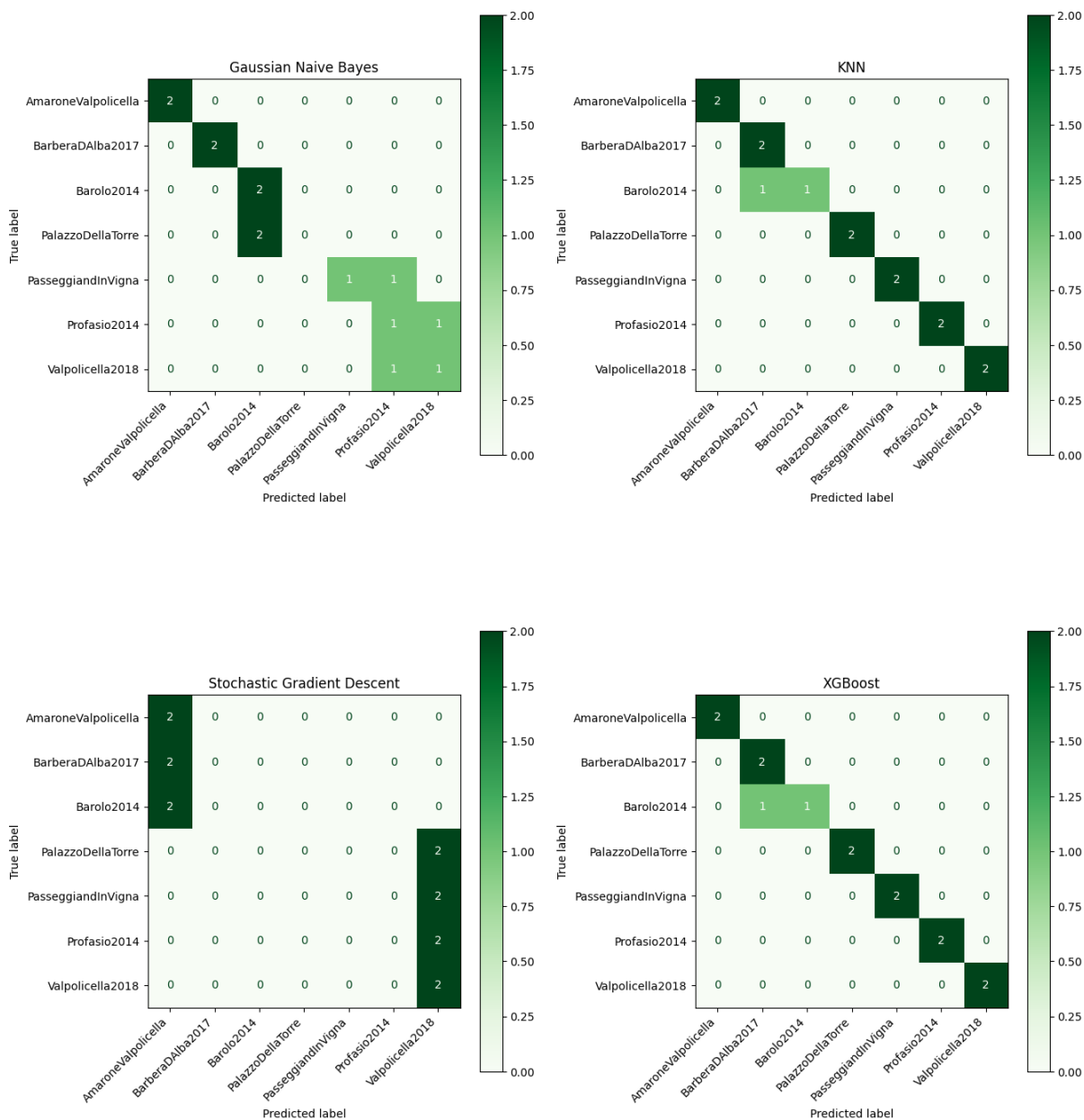


Рисунок 3.9 – Матриця невдач – нерозгорнуті дані, MTF, ResNet-50

resnet_101

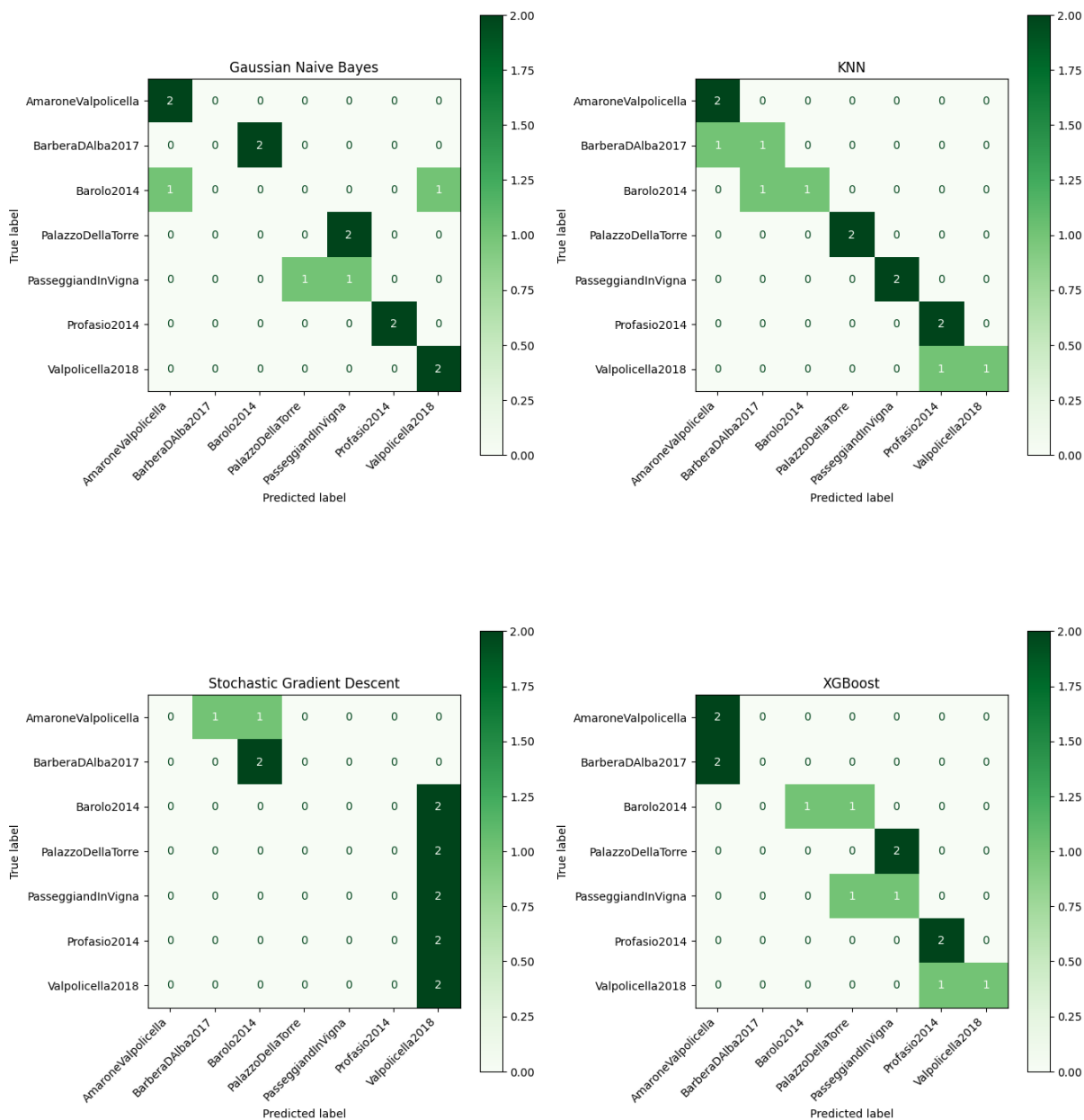


Рисунок 3.10 – Матриця невдач – нерозгорнуті дані, MTF, ResNet-101

vit_base_224

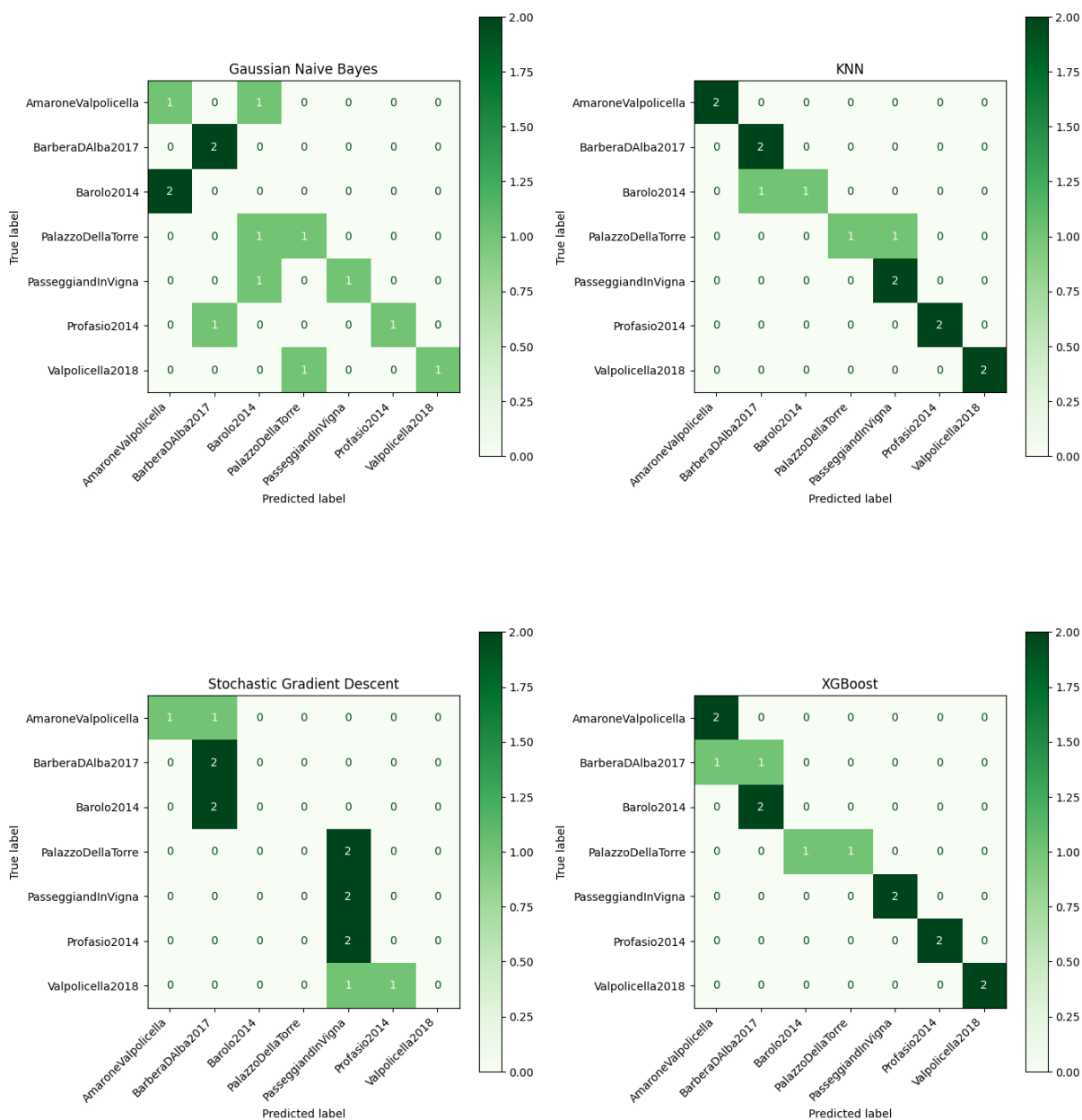


Рисунок 3.11 – Матриця невдач – нерозгорнуті дані, MTF, ViT-Base-224

vit_large_224

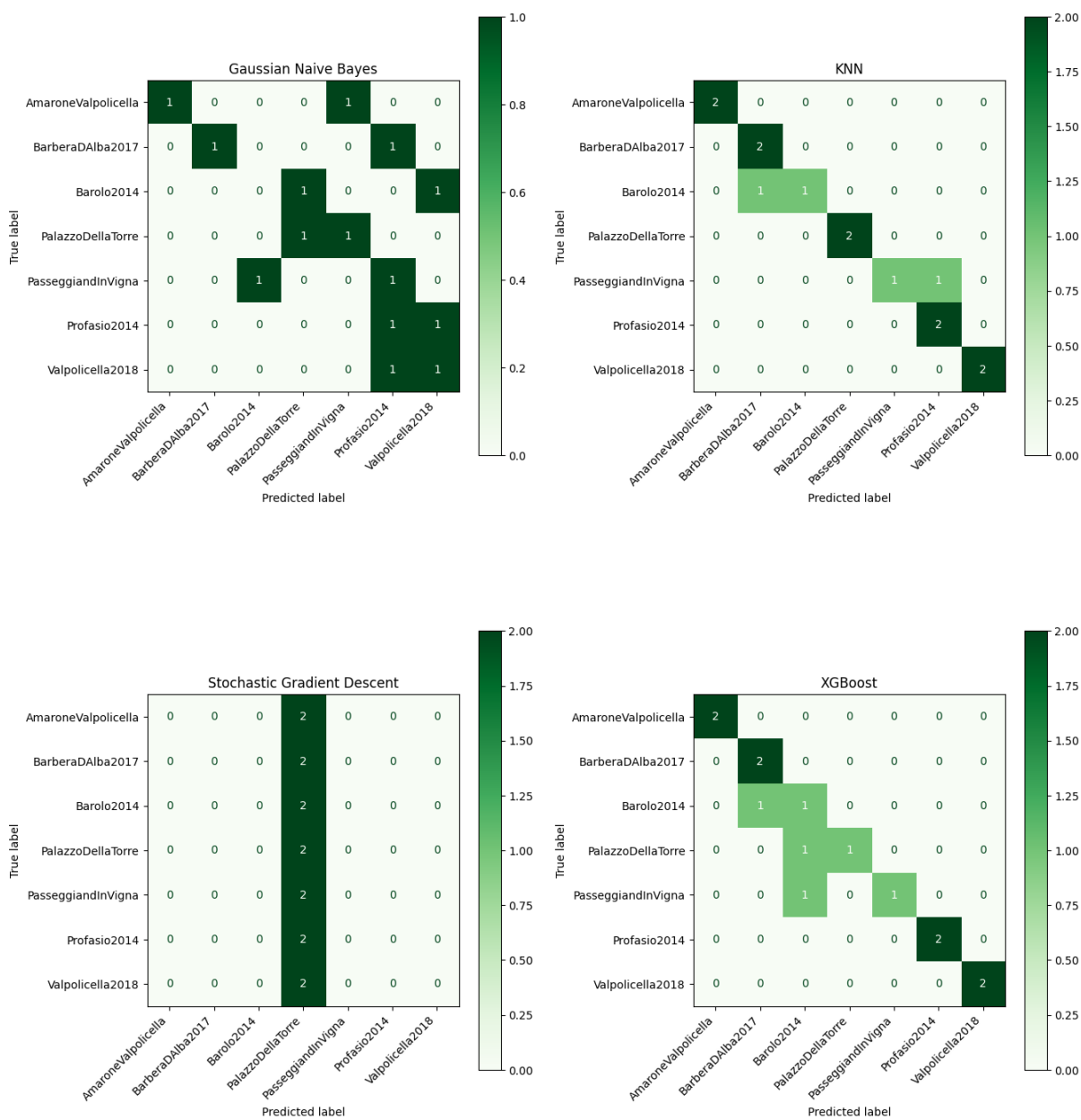


Рисунок 3.12 – Матриця невдач – нерозгорнуті дані, MTF, ViT-Large-224

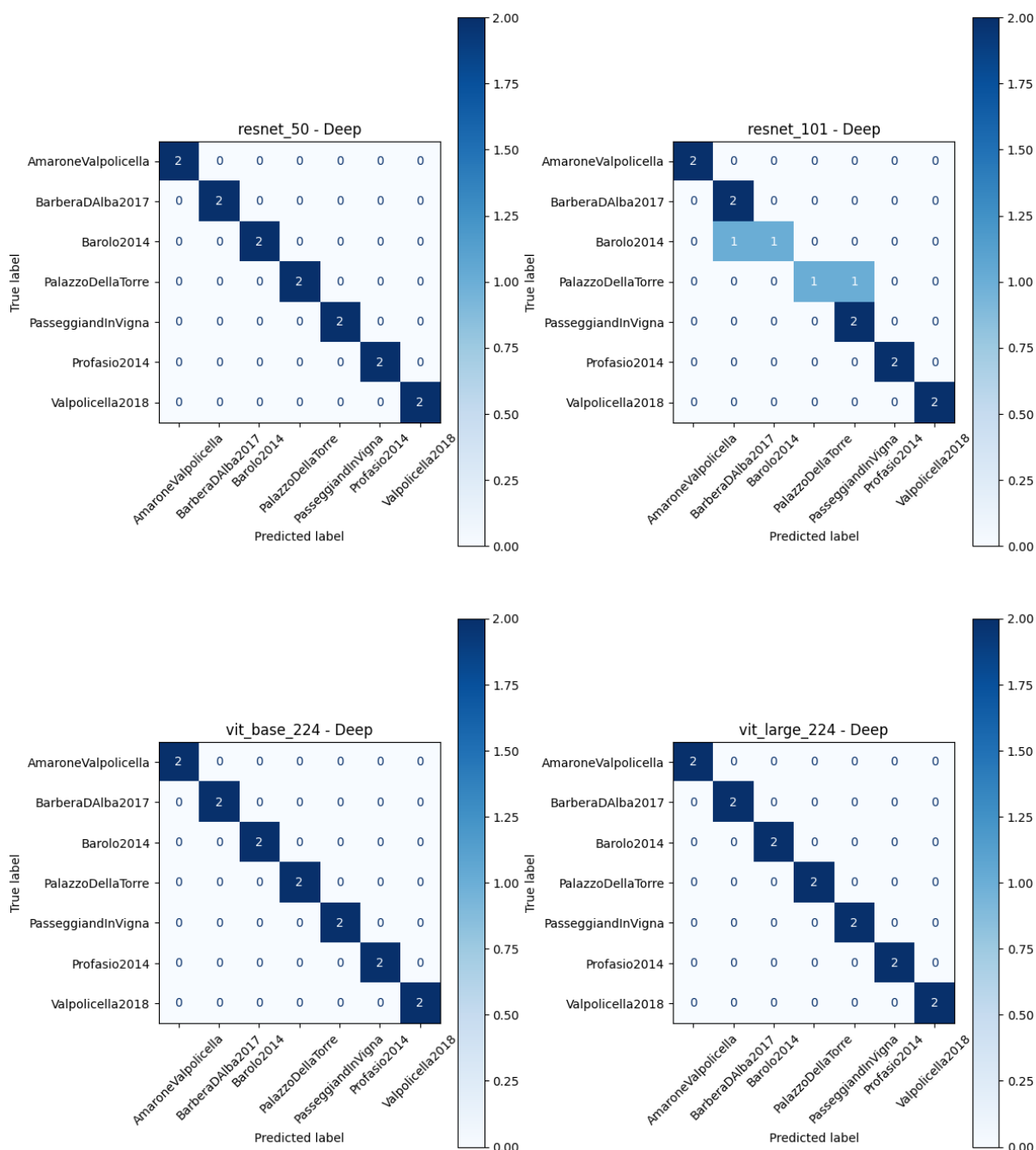


Рисунок 3.13 – Матриця невдач – нерозгорнуті дані, MTF, NN

При класифікації нерозгорнутих даних, що були перетворені за допомогою GAF (таблиця 3.2, класифікатори видають значно ліпші результати – навіть деякі з методів машинного навчання демонструють стовідсоткову точність, а нейронна мережа стабільно демонструє повний успіх. Помилки класифікаторів можна побачити на рисунках 3.14–3.18.

Таблиця 3.2 – Результати для нерозгорнутих матриць при використанні GAF

Model	Classifier	Accuracy	F1 Score
ResNet-50	Gaussian Naive Bayes	0.643	0.648
	KNN	0.929	0.933
	Stochastic Gradient Descent	0.143	0.245
	XGBoost	0.786	0.790
	Neural Network	1	1
ResNet-101	Gaussian Naive Bayes	0.714	0.767
	KNN	0.929	0.933
	Stochastic Gradient Descent	0.143	0.250
	XGBoost	0.857	0.905
	Neural Network	1	1
ViT-Base-224	Gaussian Naive Bayes	0.500	0.543
	KNN	1.000	1.000
	Stochastic Gradient Descent	0.214	0.324
	XGBoost	0.643	0.671
	Neural Network	1	1
ViT-Large-224	Gaussian Naive Bayes	0.500	0.500
	KNN	0.929	0.933
	Stochastic Gradient Descent	0.429	0.562
	XGBoost	0.857	0.862
	Neural Network	1	1

resnet_50

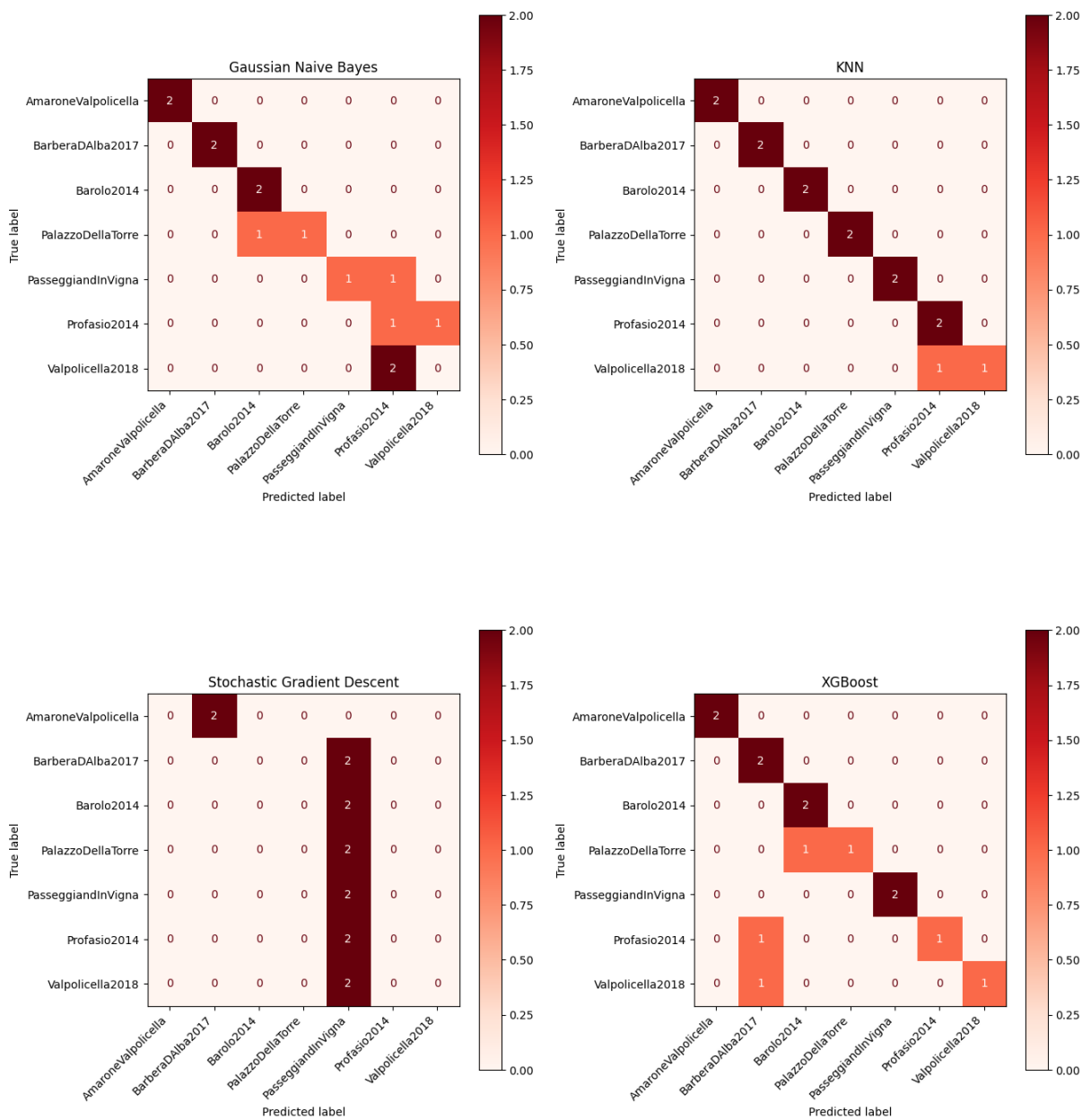


Рисунок 3.14 – Матриця невдач – нерозгорнуті дані, GAF, ResNet-50

resnet_101

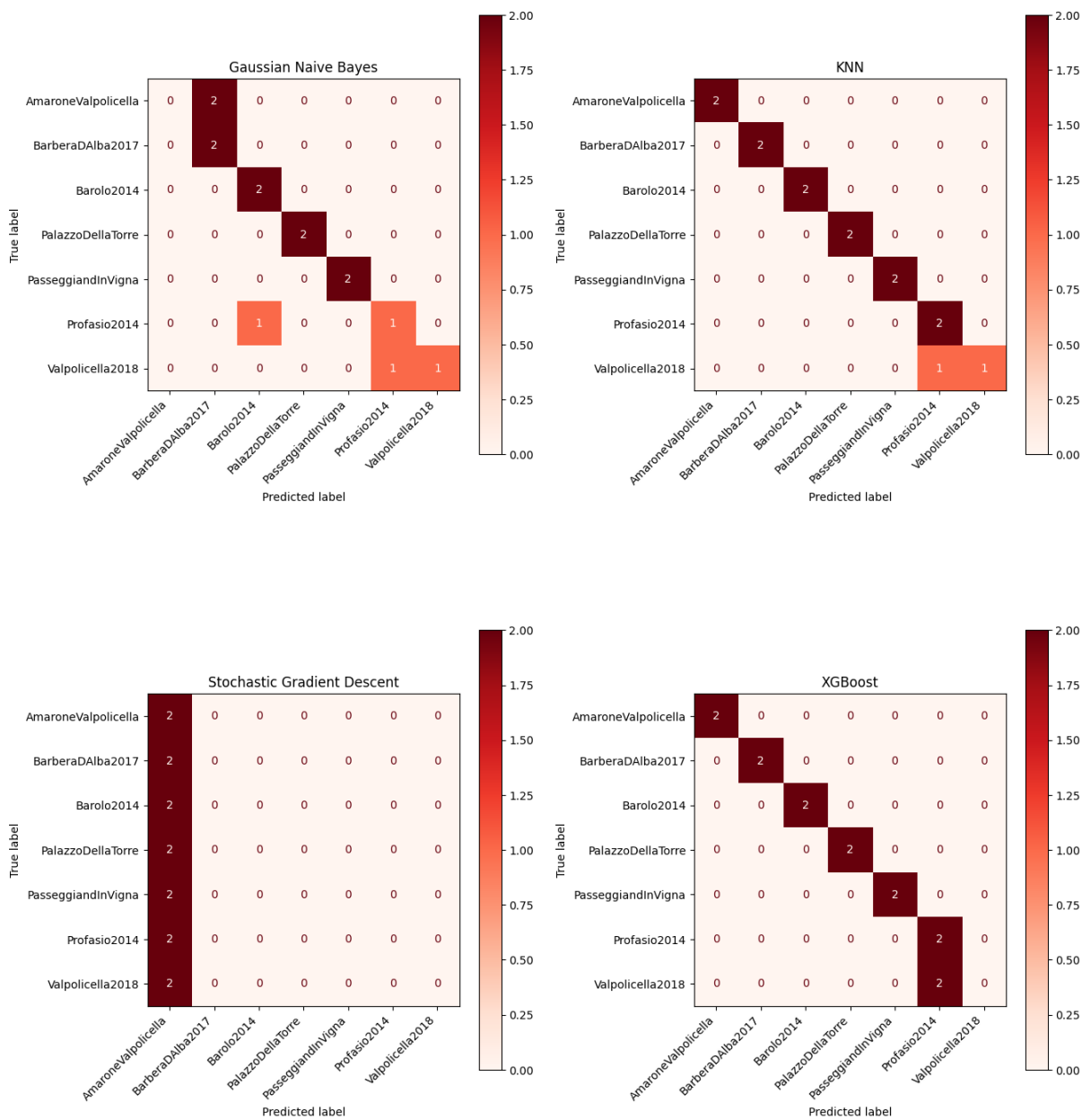


Рисунок 3.15 – Матриця невдач – нерозгорнуті дані, GAF, ResNet-101

vit_base_224

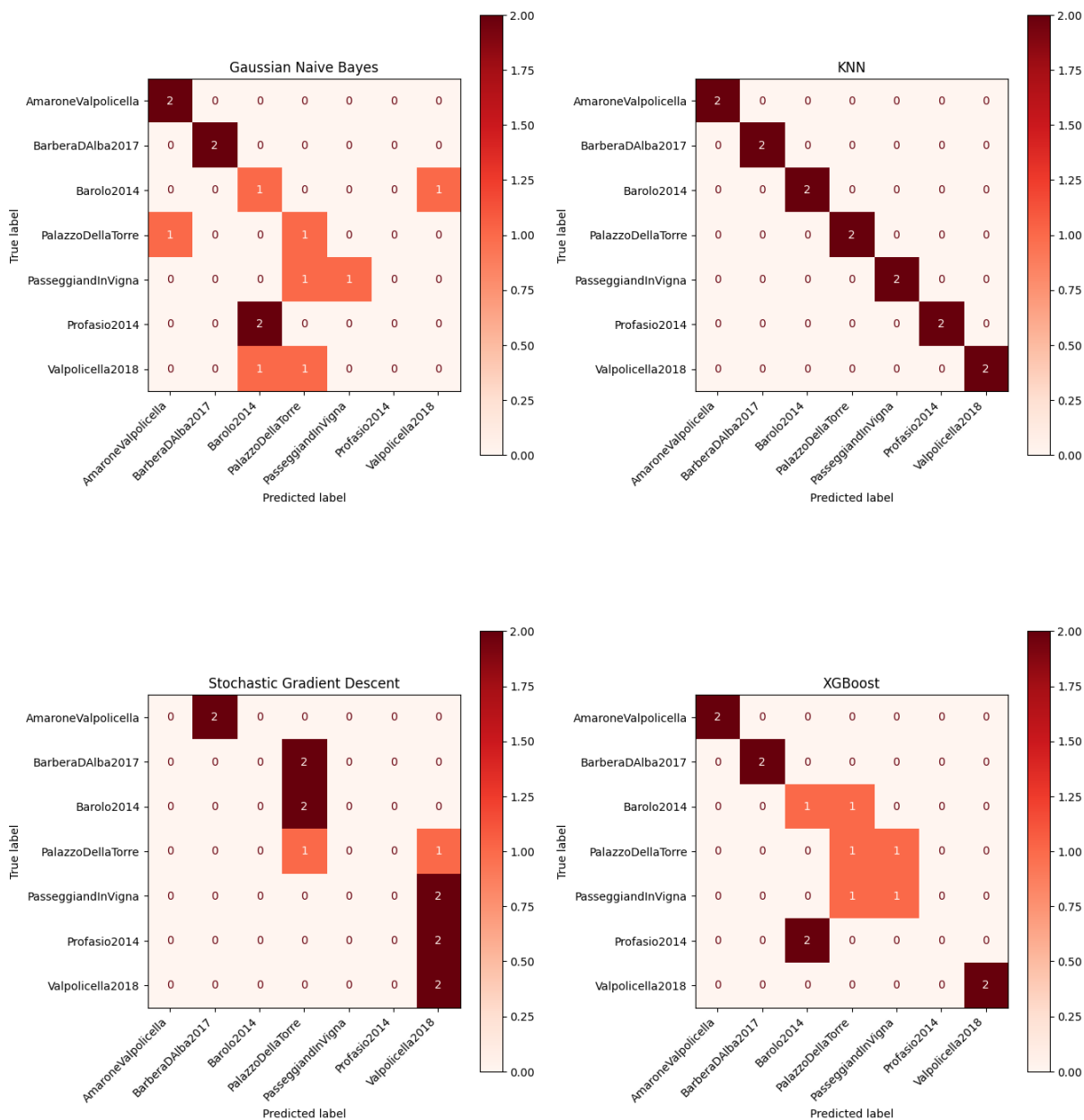


Рисунок 3.16 – Матриця невдач – нерозгорнуті дані, GAF, ViT-Base-224

vit_large_224

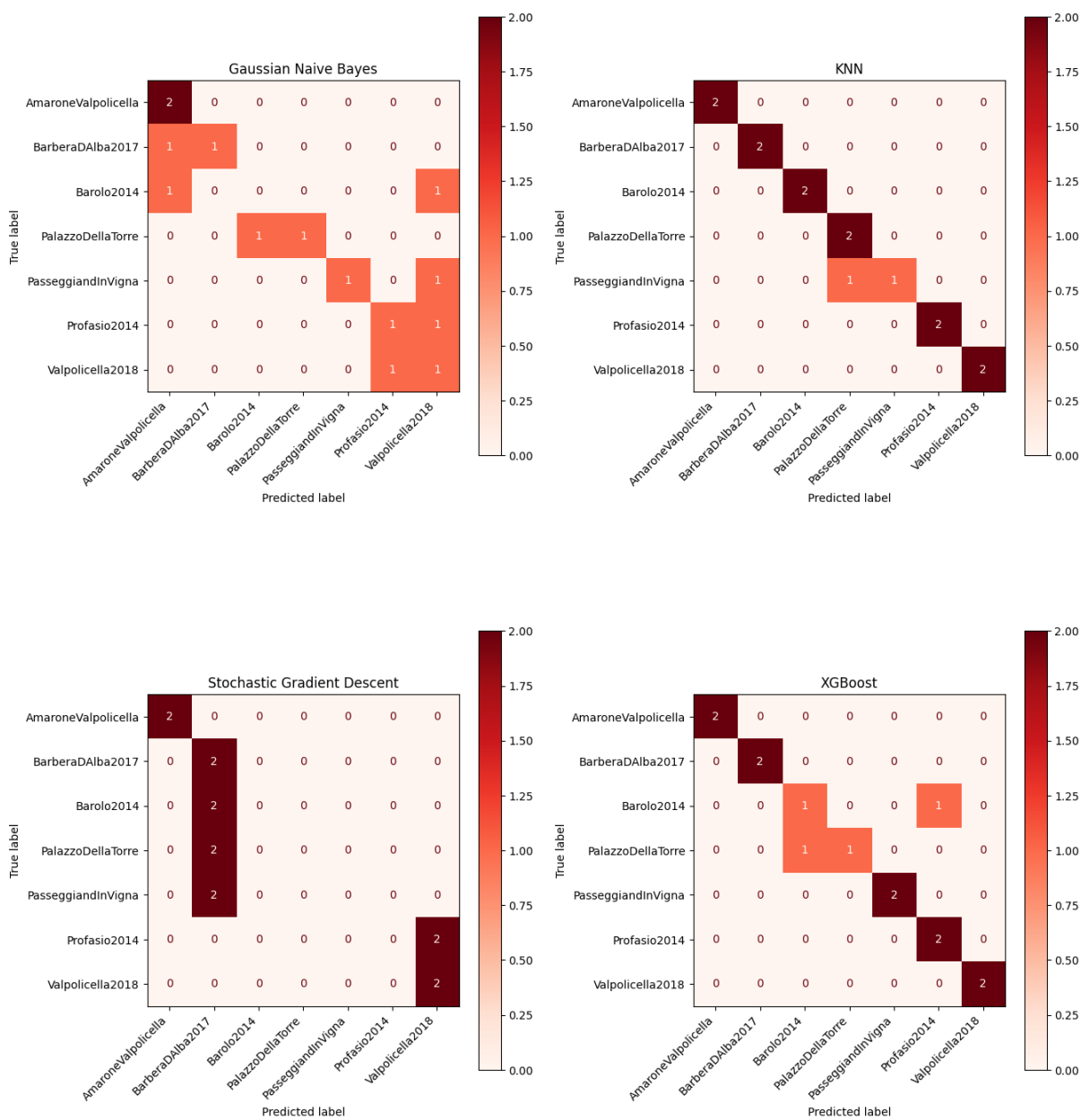


Рисунок 3.17 – Матриця невдач – нерозгорнуті дані, GAF, ViT-Large-224

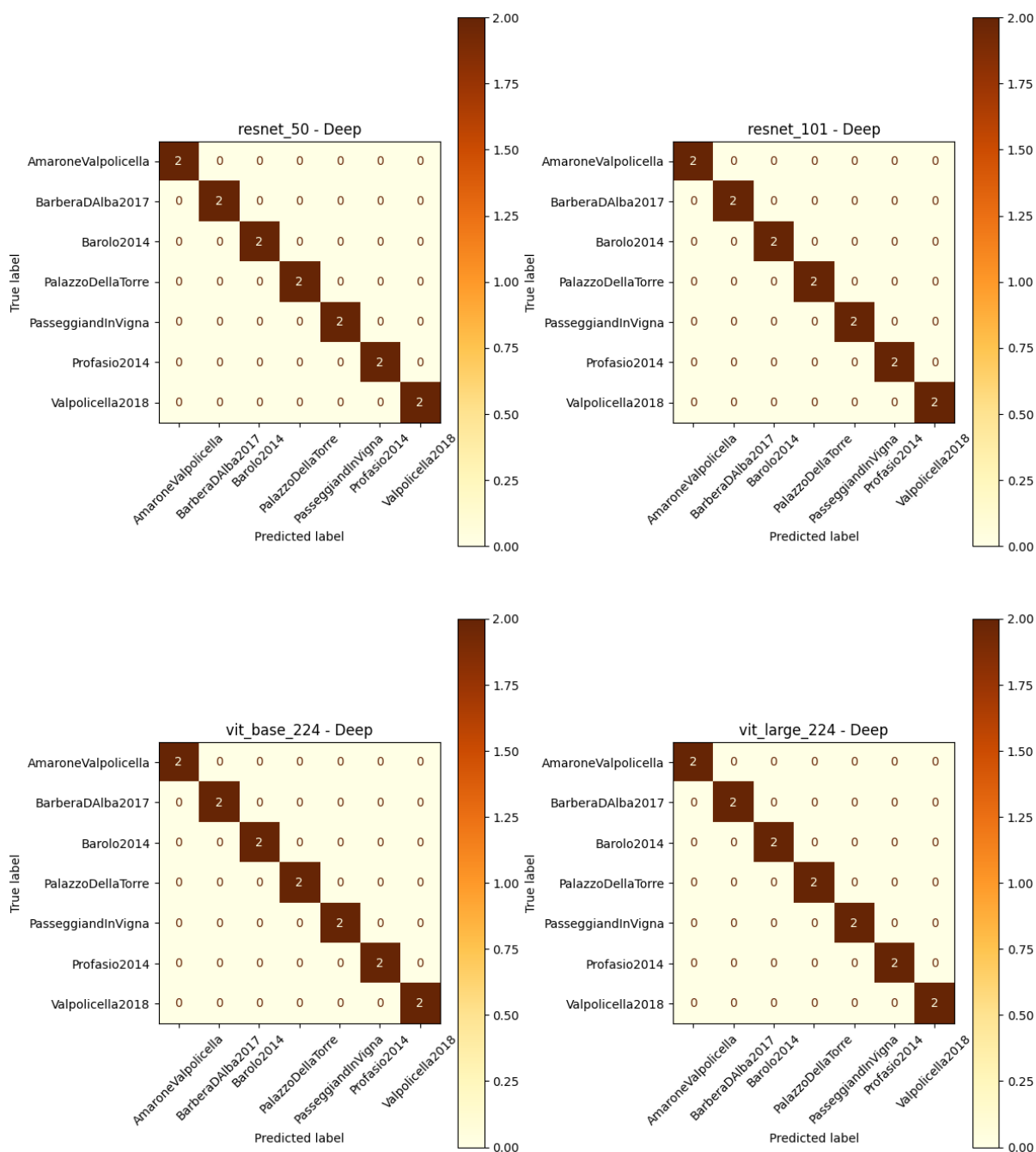


Рисунок 3.18 – Матриця невдач – нерозгорнуті дані, GAF, NN

3.6.2 Результати для розгорнутих матриць

Відповідні експерименти були проведені для даних, що були попередньо розгорнуті у вектор, як це описувалося раніше. Результати внесені у таблиці 3.3.

Таблиця 3.3 – Результати для розгорнутих матриць при використанні MTF

Model	Classifier	Accuracy	F1 Score
ResNet-50	Gaussian Naive Bayes	1.000	1.000
	KNN	0.929	0.933
	Stochastic Gradient Descent	0.286	0.443
	XGBoost	0.929	0.933
	Neural Network	1	1
ResNet-101	Gaussian Naive Bayes	0.643	0.690
	KNN	0.857	0.862
	Stochastic Gradient Descent	0.286	0.388
	XGBoost	0.429	0.467
	Neural Network	1	1
ViT-Base-224	Gaussian Naive Bayes	0.714	0.714
	KNN	0.786	0.795
	Stochastic Gradient Descent	0.286	0.405
	XGBoost	0.857	0.862
	Neural Network	1	1
ViT-Large-224	Gaussian Naive Bayes	0.500	0.548
	KNN	0.857	0.867
	Stochastic Gradient Descent	0.143	0.250
	XGBoost	0.786	0.814
	Neural Network	1	1

Таблиця 3.4 – Результати для розгорнутих матриць при використанні GAF

Model	Classifier	Accuracy	F1 Score
ResNet-50	Gaussian Naive Bayes	0.929	0.933
	KNN	1.000	1.000
	Stochastic Gradient Descent	0.286	0.388
	XGBoost	0.929	0.933
	Neural Network	1	1
ResNet-101	Gaussian Naive Bayes	0.786	0.790
	KNN	0.929	0.933
	Stochastic Gradient Descent	0.143	0.250
	XGBoost	0.643	0.695
	Neural Network	0.857	0.81
ViT-Base-224	Gaussian Naive Bayes	0.929	0.933
	KNN	1.000	1.000
	Stochastic Gradient Descent	0.286	0.443
	XGBoost	1.000	1.000
	Neural Network	1	1
ViT-Large-224	Gaussian Naive Bayes	0.786	0.790
	KNN	0.929	0.933
	Stochastic Gradient Descent	0.429	0.524
	XGBoost	0.786	0.823
	Neural Network	1	1

Як можна помітити, класифікація картинок, що були утворені з розгорнутих даних, є набагато більш ефективною. Це пов'язано з розмірністю картинок, що генеруються, а отже і з їх інформативністю. При цьому, моделі, навчені на картинках, що були утворені шляхом перетворення за допомогою GAF алгоритму, продовжують демонструвати кращі результати. Також цікаво, що більшої точності вдалося досягнути при навчанні класифікаторів на ознаках, що вилучалися меншими моделями. Це твердження є справедливим як і для згорткової мережі, так і для трансформера.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи були досліджена сучасні методи використання штучних нейронних мереж при обробці даних хімічного аналізу, обговорені можливі використання систем, їхні потенційні можливості та перспективи. Зокрема, була детально розкрита тема використання електронного язика – набору електрохімічних сенсорів, що в поєднанні з обчислювальним модулем та штучними нейронними мережами може імітувати роботу справжнього органу живих істот – органу смаку.

Робота в подробицях розкриває принципи роботи штучних електронних язиків, а також детально описує необхідні алгоритми та засоби навчання, що необхідні для швидкої та зручної обробки вихідних даних сенсорів, з яких складаються ці язики. Було розглянуто статті, а також опорні дані для дослідження – промарковані дані з сенсорів, що послідовно занурюються у винні напої.

В експериментальній частині роботи детально проаналізовані дані, описані алгоритми для їх попередньої обробки та перетворень. Під час виконання експериментів було протестовано різні класифікатори, що працювали із графічним відображенням часових рядів. Результати експериментів свідчать про те, що застосування штучних нейронних мереж для подібних пристроїв, як електронний язик, є незамінними. Штучні нейронні мережі демонструють свою величезну потужність та ефективність при роботі зі складними кросселективними даними. Також результати експериментів взаємно валідують експерименти з опорної статті [19]. Можна зазначити, що ефективність класифікаторів досить схожа.

Нейромережі в поєднанні з методами машинного навчання відкривають неосяжні простори для можливих способів покращення інструментів хімічного аналізу, зокрема, створення більш досконалих

електронних язиків та подібних до них сенсорів. При розростанні обсягів даних для навчання та класифікації, відкриється можливість для створення більш комплексних систем, що будуть включати більше компонентів, та зможуть значно ближче наблизитися до імітації реального язика, або навіть перевершити його.

Отже, в підсумку можна зазначити, що ці експериментальні дослідження є лише невеличкою часткою демонстрації можливостей штучних нейронних мереж у поєднанні з будь-якими хімічними датчиками, такими, як електронні носи, електронні язики тощо. Подібні системи мають широку область перспективних застосувань та високу конкурентоспроможність продуктів із залученням технологій нейромереж та систем ШІ.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Braun R. D. Chemical analysis | Definition, Methods, Instruments, Examples, & Facts. *Encyclopedia Britannica*. URL: <https://www.britannica.com/science/chemical-analysis> (date of access: 26.03.2024).
2. Nonspecific sensor arrays (“electronic tongue”) for chemical analysis of liquids (IUPAC Technical Report) / Y. Vlasov et al. *Pure and Applied Chemistry*. 2005. Vol. 77, no. 11. P. 1965–1983. URL: <https://doi.org/10.1351/pac200577111965> (date of access: 26.03.2024).
3. Electronic Tongue—A Tool for All Tastes? / M. Podražka et al. *Biosensors*. 2017. Vol. 8, no. 1. P. 3. URL: <https://doi.org/10.3390/bios8010003> (date of access: 27.03.2024).
4. Vlasov Y., Legin A. Non-selective chemical sensors in analytical chemistry: from “electronic nose” to “electronic tongue”. *Fresenius’ Journal of Analytical Chemistry*. 1998. Vol. 361, no. 3. P. 255–260. URL: <https://doi.org/10.1007/s002160050875> (date of access: 29.03.2024).
5. Abdel-Mageed S., Mohamed E. The Electronic Tongue – Basic Principles and Medical Applications. *J. Biophys. Biomed. Sci.* 2010. No. 3. P. 290–295.
6. Exploration of an impedimetric electronic tongue and chemometrics for characterization of black tea from different origins / D. R. K. Raj et al. *Journal of Food Composition and Analysis*. 2023. P. 105535. URL: <https://doi.org/10.1016/j.jfca.2023.105535> (date of access: 26.05.2024).
7. Electronic Tongue for Direct Assessment of SARS-CoV-2-Free and Infected Human Saliva—A Feasibility Study / M. Falk et al. *Biosensors*. 2023. Vol. 13, no. 7. P. 717. URL: <https://doi.org/10.3390/bios13070717> (date of access: 26.05.2024).
8. Electronic tongue for pharmaceutical analytics: quantification of tastes and masking effects / A. Legin et al. *Analytical and Bioanalytical Chemistry*. 2004. Vol. 380, no. 1. URL: <https://doi.org/10.1007/s00216-004-2738-3> (date

of access: 26.05.2024).

9. Cetó X., Valle M. D. Electronic tongue applications for wastewater and soil analysis. *iScience*. 2022. Vol. 25, no. 5. URL: <https://doi.org/10.1016/j.isci.2022.104304> (date of access: 26.05.2024).

10. del Valle M. Sensor Arrays and Electronic Tongue Systems. *International Journal of Electrochemistry*. 2012. Vol. 2012. P. 1–11. URL: <https://doi.org/10.1155/2012/986025> (date of access: 27.03.2024).

11. del Valle M. Electronic Tongues Employing Electrochemical Sensors. *Electroanalysis*. 2010. P. n/a. URL: <https://doi.org/10.1002/elan.201000013> (date of access: 28.03.2024).

12. Chemical, Gas, and Biosensors for Internet of Things and Related Applications. Elsevier, 2019. URL: <https://doi.org/10.1016/c2017-0-03327-x> (date of access: 29.05.2024).

13. Tasting of beverages using an electronic tongue / A. Legin et al. *Sensors and Actuators B: Chemical*. 1997. Vol. 44, no. 1-3. P. 291–296. URL: [https://doi.org/10.1016/s0925-4005\(97\)00167-6](https://doi.org/10.1016/s0925-4005(97)00167-6) (date of access: 29.03.2024).

14. Horn R. A., Johnson C. R. Matrix Analysis. 2nd ed. *Cambridge: Cambridge University Press*, 2012. 662 p. URL: <https://doi.org/10.1017/CBO9781139020411> (date of access: 29.03.2024).

15. McCulloch W. S., Pitts W. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*. 1943. Vol. 5, no. 4. P. 115–133. URL: <https://doi.org/10.1007/bf02478259> (date of access: 29.03.2024).

16. Grossi E., Buscema M. Introduction to artificial neural networks. *European Journal of Gastroenterology & Hepatology*. 2007. Vol. 19, no. 12. P. 1046–1054. URL: <https://doi.org/10.1097/meg.0b013e3282f198a0> (date of access: 29.03.2024).

17. Matsumoto G. Neurocomputing—neurons as microcomputers. *Future Generation Computer Systems*. 1988. Vol. 4, no. 1. P. 39–51. URL: [https://doi.org/10.1016/0167-739x\(88\)90018-0](https://doi.org/10.1016/0167-739x(88)90018-0) (date of access: 29.03.2024).

18. Combining an Integrated Sensor Array with Machine Learning for the Simultaneous Quantification of Multiple Cations in Aqueous Mixtures / G. Gabrieli et al. *Analytical Chemistry*. 2021. Vol. 93, no. 50. P. 16853–16861. URL: <https://doi.org/10.1021/acs.analchem.1c03709> (date of access: 28.04.2024).
19. Gabrieli G., Muszynski M., Ruch P. W. A reconfigurable integrated electronic tongue and its use in accelerated analysis of juices and wines. 2022 *IEEE International Symposium on Olfaction and Electronic Nose (ISOEN)*, Aveiro, 29 May – 1 June 2022. 2022. P. 1–3. URL: <https://doi.org/10.1109/ISOEN54820.2022.9789630> (date of access: 18.04.2024).
20. What is time series classification? / T. R. Dinger et al. *IBM Developer*. URL: <https://developer.ibm.com/learningpaths/get-started-time-series-classification-api/what-is-time-series-classification/> (date of access: 26.05.2024).
21. Wang Z., Oates T. Encoding Time Series as Images for Visual Inspection and Classification Using Tiled Convolutional Neural Networks. *AAAI-15: Twenty-Ninth Conference on Artificial Intelligence*, Austin, 25–30 January 2015.
22. Wang Z., Oates T. Spatially Encoding Temporal Correlations to Classify Temporal Data Using Convolutional Neural Networks. *arXiv.org*. URL: <https://doi.org/10.48550/arXiv.1509.07481> (date of access: 25.05.2024).
23. Faouzi J., Janati H. pyts: A python package for time series classification. *Journal of Machine Learning Research*. 2020. Vol. 21, no. 46. P. 1–6. URL: <http://jmlr.org/papers/v21/19-763.html> (date of access: 26.05.2024).
24. Markov Transition Fields and Deep Learning-Based Event-Classification and Vibration-Frequency Measurement for ϕ -OTDR / X. Zhao et al. *IEEE Sensors Journal*. 2022. Vol. 22, no. 4. P. 3348–3357. URL: <https://doi.org/10.1109/jsen.2021.3137006> (date of access: 26.05.2024).
25. A semi-supervised deep rule-based classifier for robust finger knuckle-print verification / M. Benmalek et al. *Evolving Systems*. 2022. URL: https://doi.org/10.1007/978-3-031-17111-1_10 (date of access: 26.05.2024).

[org/10.1007/s12530-021-09417-x](https://doi.org/10.1007/s12530-021-09417-x) (date of access: 26.05.2024).

26. Lowe D. G. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*. 2004. Vol. 60, no. 2. P. 91–110. URL: <https://doi.org/10.1023/b:visi.0000029664.99615.94> (date of access: 26.05.2024).

27. Dalal N., Triggs B. Histograms of Oriented Gradients for Human Detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA. URL: <https://doi.org/10.1109/cvpr.2005.177> (date of access: 26.05.2024).

28. Deep Learning for Image Classification: A Review / M. Wu et al. *Lecture Notes in Electrical Engineering*. Singapore, 2024. P. 352–362. URL: https://doi.org/10.1007/978-981-97-1335-6_31 (date of access: 26.05.2024).

29. Luo L. Research on Image Classification Algorithm Based on Convolutional Neural Network. *Journal of Physics: Conference Series*. 2021. Vol. 2083, no. 3. P. 032054. URL: <https://doi.org/10.1088/1742-6596/2083/3/032054> (date of access: 26.05.2024).

30. Transfer learning: a friendly introduction / A. Hosna et al. *Journal of Big Data*. 2022. Vol. 9, no. 1. URL: <https://doi.org/10.1186/s40537-022-00652-w> (date of access: 26.05.2024).

31. Yeturu K. Machine learning algorithms, applications, and practices in data science. *Handbook of Statistics*. 2020. P. 81–206. URL: <https://doi.org/10.1016/bs.host.2020.01.002> (date of access: 26.05.2024).

32. Goodfellow I., Bengio Y., Courville A. Convolutional Networks. *Deep Learning*. 2016. P. 326–366. URL: <http://www.deeplearningbook.org> (date of access: 26.05.2024).

33. Rage K. A Study on Different Deep Learning Architectures on Image Captioning. 2022 8th *International Conference on Smart Structures and Systems (ICSSS)*, Chennai, India, 21–22 April 2022. 2022. URL: <https://doi.org/10.1109/icsss54381.2022.9782260> (date of access: 26.05.2024).

34. Deep Residual Learning for Image Recognition / K. He et al. 2015.

URL: <https://doi.org/10.48550/arXiv.1512.03385>.

35. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale / A. Dosovitskiy et al. 2020. URL: <https://doi.org/10.48550/arXiv.2010.11929> (date of access: 26.05.2024).

36. Shape-Based Image Classification and Retrieval / N. Mohanty et al. *Handbook of Statistics - Machine Learning: Theory and Applications*. 2013. P. 249–267. URL: <https://doi.org/10.1016/b978-0-444-53859-8.00010-2> (date of access: 26.05.2024).

37. Cunningham P., Delany S. J. k-Nearest Neighbour Classifiers - A Tutorial. *ACM Computing Surveys*. 2021. Vol. 54, no. 6. P. 1–25. URL: <https://doi.org/10.1145/3459665> (date of access: 26.05.2024).

38. Netrapalli P. Stochastic Gradient Descent and Its Variants in Machine Learning. *Journal of the Indian Institute of Science*. 2019. Vol. 99, no. 2. P. 201–213. URL: <https://doi.org/10.1007/s41745-019-0098-4> (date of access: 26.05.2024).

39. Chen T., Guestrin C. XGBoost. *KDD '16: The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco California USA. New York, NY, USA, 2016. URL: <https://doi.org/10.1145/2939672.2939785> (date of access: 27.05.2024).