

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра прикладної математики
(повна назва)

КВАЛІФІКАЦІЙ РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Кластеризація фрактальних реалізацій
на основі машинного навчання
(тема)

Виконав:
студент 2 курсу, групи ПМм-21-1
Мельникова П. О.
(прізвище, ініціали)

Спеціальність 113 Прикладна математика
(код і повна назва спеціальності)

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

Освітня програма Прикладна математика
(повна назва освітньої програми)

Керівник проф. Кіріченко Л. О.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ПМ

Сидоров М.В.
(підпис) (прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет інформаційно-аналітичних технологій та менеджменту

Кафедра прикладної математики

Рівень вищої освіти другий (магістерський)

Спеціальність 113 Прикладна математика

(код і повна назва)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Прикладна математика

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри ПМ _____

(підпис)

“ 07 ” листопада 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Мельниковій Поліні Олексіївні

(прізвище, ім'я, по батькові)

1. Тема роботи Кластеризація фрактальних реалізацій на основі машинного навчання

затверджена наказом по університету від 25 жовтня 2022 р. № 1412 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 7 грудня 2022 р.

3. Вихідні дані до роботи математична модель фрактальних реалізацій

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Аналіз предметної області

2. Вибір і обґрунтування методу розв'язання

3. Програмна реалізація та результати обчислювального експерименту

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій _____

1. Актуальність теми роботи _____

2. Постановка задачі _____

3. Аналіз предметної області _____

4. Метод чисельного аналізу _____

5. Результати обчислювального експерименту _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Підбір та вивчення технічної літератури за темою роботи	7 – 13 листопада 2022 р.	виконано
2	Вибір та обґрунтування методу	14 – 20 листопада 2022 р.	виконано
3	Розробка алгоритму і програми	21 – 27 листопада 2022 р.	виконано
4	Проведення аналітичних досліджень та розрахунків	28 листопада – 4 грудня 2022 р.	виконано
5	Робота над текстом пояснювальної записки	28 листопада – 6 грудня 2022 р.	виконано
6	Представлення роботи на рецензію в ЕК	7 грудня 2022 р.	виконано

Дата видачі завдання 7 листопада 2022 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Кіріченко Л.О.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 54 с., 1 табл., 27 рис., 1 дод., 21 джерело.

КЛАСТЕРИЗАЦІЯ, МАШИННЕ НАВЧАННЯ, ФРАКТАЛЬНИЙ БРОУНІВСЬКИЙ РУХ, КЛАСИЧНИЙ БРОУНІВСЬКИЙ РУХ, РЕКУРЕНТНА МАТРИЦЯ, ТРАЄКТОРІЯ, ПАСТКА.

Об'єкт дослідження – процес фрактального броунівського руху частинок.

Мета роботи – аналіз і застосування методів кластеризації та подальша розробка програмного забезпечення для виявлення пасток в траєкторіях руху броуновської частинки за допомогою машинного навчання.

Методи дослідження – моделювання траєкторій фрактального броунівського руху і методи кластеризації машинного навчання.

Кваліфікаційна робота присвячена дослідженню застосування методів машинного навчання для пошуку пасток у траєкторії руху частинки. Для обчислюваного експерименту була згенерована траєкторія руху частинки, яка складається з вільних відрізків та пасток. Для пошуку пасток були застосовані такі методи кластеризації як K-means та DBSCAN.

ABSTRACT

Introductory note: 54 pages, 1 table, 27 figures, 1 appendix, 21 sources.

CLUSTERIZATION, MACHINE LEARNING, FRACTAL BROWNIAN MOTION, CLASSICAL BROWNIAN MOTION, RECURRENCE MATRIX, TRAJECTORY, TRAP.

Object of research – the process of fractal Brownian motion of particles.

Purpose of work – analysis and application of clustering methods and further development of software for detecting traps in the trajectories of Brownian particle motion using machine learning.

Methods of research – modeling of fractal Brownian motion trajectories and machine learning clustering methods.

The qualification work is devoted to the study of the application of machine learning methods to search for traps in the trajectory of a particle. For the calculated experiment, a particle motion trajectory consisting of free segments and traps was generated. Such clustering methods as K-means and DBSCAN were used to search for traps.

ЗМІСТ

	С.
Вступ	7
1 Аналіз предметної області та постановка задач дослідження	10
1.1 Аналіз предметної області	10
1.2 Змістовна та формальна постановки задачі	13
1.2.1 Змістовна постановка задачі	13
1.2.2 Формальна постановка задачі	14
1.3 Постановка задач дослідження	17
2 Вибір та обґрунтування методу розв’язання	19
2.1 Моделювання фрактального броунівського руху методом Холецького...	19
2.2 Рекурентна матриця	22
2.3 Метод DBSCAN	25
2.4 Метод K-means	28
3 Програмна реалізація та результати обчислювального експерименту	31
3.1 Python та машинне навчання	31
3.2 Опис роботи програми та результати обчислень	32
4 Результати обчислювального експерименту та їх аналіз	34
Висновки	45
Перелік джерел посилання	46
Додаток А Лістинг програми	48

ВСТУП

Актуальність теми. Класифікація та кластеризація траєкторій частинок являються складними завданнями. До задачі кластеризації відносяться виділення траєкторій, коли частинка попадає у так звані пастки – області простору, де вона уповільнює свій рух. Виявлення пасток є складним завданням і було предметом розслідування кількох авторів. Виявлення пасток та взагалі аномалій у русі білків є важливим завданням біологів. Тому ця тема є ще відкритою для досліджень та має практичне застосування у біології, біохімії та біофізики.

Броунівський рух – невпорядкований, хаотичний рух частинки під дією нерівномірних ударів молекул речовини з різних боків у розчинах. Названий на честь ботаніка Роберта Броуна, який спостерігав це явище під мікроскопом у 1827 р. Теорію броунівського руху сформулював у 1905 р. Альберт Ейнштейн.

Відкриття й пояснення броунівського руху мало велике значення для фізики, оскільки було свідченням теплового руху молекул. Р. Броун відкрив цей хаотичний рух випадково, спостерігаючи за спорами 2 плауна у воді. Рух завислих частинок відбувався внаслідок руху молекул. Молекули рідини зіштовхуються з завислими у ній частинками, отже й передають їм імпульс. Таким же чином рухаються частинки фарби у воді, пилинки в променях світла (хоча на рух пилинок також впливають і мікропотоки в повітрі) тощо.

У броунівському русі вражає одна незвична для нас особливість – рух частинок не припиняється за будь яких обставин, хоча під час дослідження його причин вживалися запобіжні заходи, які виключали можливість різних зовнішніх впливів на броунівські частинки. Характер їх руху не змінювався.

Одиночні методи, які відстежують флуоресцентні молекули з плином часу, дозволяють проводити кількісне визначення біологічних подій з безпрецедентним просторовим і часовим дозволом. У клітинній біології складна організація плазматичної мембрани істотно впливає на бічну дифузію мембранних білків, приводячи до нестационарних моделей руху. Правильна інтерпретація цих складних траєкторій вимагає врахування зміни в основному механізмі руху молекули.

Наприклад, транзиторне уловлювання рецепторів, пов'язаних з G-білком, і G-білків тісно пов'язане з моделлю обмеженого зіткнення [1, 2]. У цій моделі швидкість асоціації молекул на плазматичній мембрані посилюється наявністю обмежених нанодоменів, де рецептори і G-білки частіше стикаються один з одним. Однак використання інструментів аналізу, які припускають один і той же завданням і було предметом розслідування кількох авторів молекулярний рух з плином часу, призводить до неправильних інтерпретацій базової біології. Переривчастий процес, що чергується між вільним броунівським рухом і пастками [3], може бути помилково інтерпретований як випадок аномальної дифузії з аномальним показником $\alpha < 1$.

Мета і завдання кваліфікаційної роботи. Метою кваліфікаційної роботи є аналіз і застосування методів кластеризації та подальша розробка програмного забезпечення для виявлення пасток в траєкторіях руху броунівської частинки за допомогою машинного навчання. Для досягнення поставленої мети необхідно виконати наступні завдання:

- провести огляд і аналіз сучасного стану задачі «кластеризація фрактальних реалізацій на основі машинного навчання»
- вивчити методи машинного навчання і визначити ті, які можуть бути застосовані для розв'язання задачі виявлення пасток;
- вивчити алгоритми кластеризації і вибрати найбільш відповідний для реалізації;
- обрати інструменти програмної розробки і програмно реалізувати обраний метод;
- вибрати математичні моделі фрактальних траєкторій і програмно їх реалізувати;
- на основі реалізованих алгоритмів і модельних траєкторій перевірити метод виявлення пасток.

Об'єктом дослідження є процес фрактального броунівського руху часток.

Предметом дослідження є кластеризація траєкторій часток для пошуку пасток на основі застосування методів машинного навчання.

Методи дослідження. У кваліфікаційній роботі використовуються методи моделювання траєкторій фрактального броунівського руху і методи кластеризації машинного навчання.

Публікації. Результати, отримані у кваліфікаційній роботі, було представлено на XVI Міжнародної науково-практичної конференції «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті» (м. Дніпро, 14-15 грудня, 2022 року) [21].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Аналіз предметної області

Одночастинкові траєкторії, виміряні в експериментах з мікроскопії, містять важливу інформацію про динамічні процеси, що проходять в ряді матеріалів, включаючи живі клітини і тканини. Однак вилучення цієї інформації не є тривіальним завданням через стохастичний характер руху частинок та шум вибірки.

На стороні однотраєкторного аналізу були засновані методики на максимальному зміщенні квадратів, хоча вони, як правило, занадто чутливі до шуму і локальних коливань динаміки траєкторії. Слідом за цим був розроблений ряд методів, серед яких: методи аналізу зображень, модель конкретного оцінювача максимальної ймовірності, випадкові лісові моделі, зворотне поширення нейронних мережових підходів, аналіз спектра масштабування моменту і стандартизована максимальна відстань. Більшість з перерахованих вище прийомів спираються на підходи до часових вікон.

Останні досягнення в технології одномолекулярної мікроскопії та візуалізації зробили одночастинкове відстеження (single-particle tracking, SPT) популярним методом аналізу динамічних процесів у ряді матеріалів [4, 5]. При типовому вимірюванні SPT цікаві молекули (наприклад, білки в живій клітині) позначені флуоресцентними частинками барвника. Після підсвічування лазером молекули виробляють світло і їх положення можна визначити за допомогою мікроскопа. Використання лазерів, які спалахують через короткі проміжки часу, дозволяє відстежувати рух молекул у часі. Записані положення використовуються для реконструкції траєкторій окремих молекул. Потім ці траєкторії аналізуються з метою вилучення локальних фізичних властивостей молекул і їх середовища, таких як швидкість, коефіцієнт дифузії (або тензор) і конфайнмент (локальна щільність перешкод) [6].

Метод SPT має особливе значення для фундаментальної біології. Він долає розрив між біологією, біохімією та біофізикою і дозволяє хоча б частково зрозуміти живі клітини на мікроскопічній основі. Це допомогло вже розкрити деталі руху молекулярних двигунів всередині клітин [7, 8] або цільових механізмів пошуку ядерних білків [9].

Аналіз траєкторій SPT не є тривіальним завданням через стохастичний характер руху молекул. Зазвичай все починається з виявлення відповідного типу руху молекули, оскільки ця інформація вже може дати уявлення про механічні властивості навколишнього молекули. В експериментах SPT спостерігаються чотири основних типи руху: нормальна дифузія (normal diffusion, ND), спрямований рух (directed motion, DM), аномальна дифузія (anomalous diffusion, AD) і обмежена дифузія (confined diffusion, CD). Найбільш поширений метод аналізу використовує криві середнього квадратного зміщення (mean square displacement, MSD) [10]. В рамках цього підходу до даних вписуються теоретичні криві для різних фізичних моделей, а потім вибираються найбільш підходящі для статистичного аналізу. Однак у багатьох випадках фактичні траєкторії занадто короткі для вилучення значущої інформації з усереднених за часом MSD.

Отже, для подолання цих проблем було введено кілька альтернативних методів. Наприклад, повний розподіл переміщень може бути пристосований до змішаної моделі з метою вилучення відмінностей у дифузійній поведінці між підмножинами ансамблів частинок. Метод спектра масштабування моменту також може бути використаний для класифікації різних режимів руху. Розподіл спрямованих змін, метод середньої максимальної екскурсії і дробово-інтегрована структура ковзної середньої (fractionally integrated moving average, FIMA) [11] можуть ефективно замінити оцінювач MSD для цілей класифікації. Приховані марківські моделі (Hidden Markov Models, HMM) були запропоновані для перевірки неоднорідності в межах одиничних траєкторій [12]. Фільтрація частинок також може бути використана для знаходження місць зв'язування процесів з перехідним конфайнментом.

Можливий і альтернативний підхід до аналізу траєкторій, що сягають корінням в інформатику і статистику. Завдяки алгоритмічним досягненням у поєднанні з підвищеною доступністю даних та більш потужними комп'ютерами методи машинного навчання (machine learning, ML) вже можуть перевершити людських експертів у деяких завданнях, включаючи класифікацію, тобто проблему ідентифікації того, до якої категорії належить нове спостереження на основі навчального набору даних, що містить спостереження з відомою категорією приналежності. Оскільки виявлення руху потрапляє в область класифікації, можна спробувати вирішити цю проблему за допомогою алгоритмів машинного навчання. Цей підхід дуже привабливий, оскільки він дозволить автоматизовано аналізувати багато сотень або навіть тисяч траєкторій зі зменшеною кількістю ручного втручання та курації вихідних даних.

Вже проведено кілька спроб аналізу траєкторій STF методами ML. Наприклад, Monnier et al [13] використовували байєсівський підхід до класифікації режимів руху на основі MSD. Доссет і колеги [14] використовували просту нейронну мережу зворотного поширення для дискримінації різних типів дифузії. Вагнер та інші [15] побудували випадковий класифікатор лісів для нормальної, аномальної, обмеженої та спрямованої дифузії. Хоча кожна з цих спроб використовує різний алгоритм класифікації машинного навчання, всі вони належать до класу методів, заснованих на ознаках. Кожна траєкторія в рамках цього підходу описується набором інженерних людиною ознак, і тільки ці ознаки були надані як вхідні дані до класифікатора модель.

На відміну від цього, методи глибокого навчання витягують функції самостійно з необроблених даних, без будь-яких зусиль з боку людського експерта. Вони набирають популярність в останні роки і вже успішно застосовуються для комп'ютерного зору, розпізнавання мови і обробки природної мови. Одним з популярних методів є згорткові нейронні мережі (convolutional neural networks, CNN) [16], які визначають місце в класифікації зображень. Вони вже були застосовані для розпізнавання частинок в експериментах з мікроскопії. Однак, хоча деякі спроби аналізу часових рядів за допомогою CNN вже відомі,

наскільки нам відомо, вони ще не застосовані до проблеми класифікації типів руху від необроблених траєкторій.

1.2 Змістовна та формальна постановки задачі

1.2.1 Змістовна постановка задачі

У нас є траєкторія руху частинки у двомірному просторі. За припущення-ми траєкторія руху може містити одну чи більше частин, що можна ідентифікувати як пастки.

Для цього ми збираємося застосувати методи машинного навчання до двох наборів даних, що відповідають одній траєкторії, порівняти результати та на основі цього зробити висновки.

Двома наборами даних для траєкторії є, по-перше, сам по собі набір значень двомірних координат (не має прив'язки до номеру за порядком, а отже не має прив'язки до часу) та, по-друге, матриця рекурентності, що показує силу зв'язку саме між координатами частинки, що близькі не лише за значеннями, а також за моментами у часі.

У цій роботі ми представляємо метод виявлення перехідних подій пасток в рамках однієї траєкторії. Перевагою аналізу перехідних подій пасток є можливість кількісної оцінки кінетики зв'язування молекули через різні клітинні нанодомени. Крім того, цей підхід не вимагає багаторазового аналізу незалежних молекул в один і той же нанодомен для оцінки пастки і не передбачає пасток нанодоменів як довгоживучих. Наша стратегія полягає в тому, щоб ізолювати різні захоплені ділянки траєкторій, розглядаючи просторову самолокалізацію послідовних точок в рамках однієї траєкторії.

Введемо локальні міри, обчислені для кожної точки траєкторії, $n \in [1, N]$, що містять інформацію про сусідні координати траєкторії, як спосіб з'ясування структури траєкторії. Для кожної позиції траєкторії кількість сусідів, які розг-

лядаються для локальної міри, визначається кількістю послідовних координат траєкторії в межах діапазону тестової шкали довжини.

Таким чином у пастках змінюється швидкість, а отже й змінюються міри між точками, що надає можливість використовувати кластеризацію для виявлення цих самих пасток.

1.2.2 Формальна постановка задачі

Випадковість властива всім природним явищам. Тому навіть найближчі до ідеалу системи містять елементи випадковості в реальному стані. Є вагомі підстави вважати, що багато природні явища описуються найкращим чином як фрактали. Саме Роберт Броун вперше зрозумів, що неупорядкований рух мікроскопічних частинок пилку має не біологічну природу, як вважалося до нього, а фізичну.

Розглянемо частинку, яка рухається вздовж прямої лінії – осі x , що переважає на відстань $+\xi$ чи $-\xi$ кожні τ секунд.

Нехай ξ – не фіксована величина. Задаємо її гаусовим, або нормальним, розподіленням вірогідності

$$p(\xi, \tau) = \frac{1}{\sqrt{4\pi D\tau}} \exp\left(-\frac{\xi^2}{4D\tau}\right).$$

Процес випадкового блукання в атомарних масштабах можна описати наступним чином. На кожному інтервалі довжини τ довжина крока ξ обирається випадковим чином, і вірогідність того, що ξ заключена між ξ і $\xi + d\xi$, дорівнює $p(\xi, \tau)d\xi$. Послідовність таких чисел (довжин кроків) $\{\xi_i\}$ являється набором незалежних гаусових випадкових чисел.

Перейдемо до стандартного гаусового процесу, замінив $\frac{\xi}{\sqrt{2D\tau}}$ на ξ , так що знов визначене ξ має нульове середнє и дисперсію, що дорівнює 1. Послідовністю $\{\xi_i\}$ визначається послідовність довжин кроків випадкового блукання, а координата частинки на осі x дорівнює

$$X(t = nt) = \sum_{i=1}^n \xi_i.$$

У межах довільно малої довжини кроку набір випадкових чисел переходить у випадкову функцію $X(t)$.

Гаусовий процес $X(t)$ називаємо одномірним броунівським рухом, або вінеровським процесом на інтервалі $[a, b]$, якщо він має наступні властивості:

- 1) $X(0) = 0$ і $X(t)$ майже завжди неперервна.
- 2) Властивість гаусовості приросту: випадкова величина $\Delta X = X(t_2) - X(t_1)$, $t_2 > t_1$, має гаусовий розподіл з математичним очікуванням 0 і дисперсією $\sigma^2(t_2 - t_1)$, де σ – додатна константа, то є

$$P(\Delta X < x) = \frac{1}{\sqrt{2\pi\sigma^2(t_2 - t_1)}} \int_0^x \exp\left(-\frac{u^2}{2\sigma^2(t_2 - t_1)}\right) du.$$

Броуновновський рух, як і будь-який процес з незалежним приростом, є марківським процесом. Це означає, що умовна вірогідність події $X(t_2)$, де $t_1 < t_2$, залежить тільки від t_1 та t_2 , залежить тільки від поведінки $X(t)$ при $t < t_1$, тобто в процесі випадкового блукання кожен крок робиться без якої не-будь інформації про то, яким образом процес досяг поточного значення.

Класичний броунівський рух представляє собою хорошу модель марківських випадкових фракталів, для яких умовна вірогідність того, що $X(t_2)$ досягне визначеного значення при заданому $X(t_1)$ ($t_1 < t_2$) залежить лише від t_1 та t_2 , а не від поведінки $X(t)$ при $t < t_1$. Очевидно, що є необхідність ввести такий

випадковий процес, який мав би певну пам'ять. Цей процес отримав назву фрактального броунівського руху і був вивчений Мандельбротом і Ван Нессомом в 1968 році. Неявно бачив фрактальний броунівський рух, розглянутий Колмогоровим ще в 1940 році.

Не існує простого методу апроксимації фрактального броунівського руху, такого як підсумовування випадкових величин Гауса, як у випадку з класичним броунівським рухом. З математичної точки зору найбільш логічним є використання апарату Фур'є.

Фрактальний броунівський рух зручно визначити за допомогою параметру $H, 0 < H < 1$. При $H = \frac{1}{2}$ фрактальний броунівський рух співпадає з класичним. Існування фрактального броунівського руху доведено Мандельбротом і Ван Нессом за допомогою стохастичних інтегралів.

Гаусовський процес $X(t)$ називається фрактальним броунівським рухом з параметром $H, 0 < H < 1$, якщо він має наступні властивості:

1) $X(0) = 0$ і $X(t)$ майже завжди неперервна.

2) Властивість гаусовості приросту: випадкова величина $\Delta X = X(t_2) - X(t_1)$, має гаусовий розподіл з математичним очікуванням 0 і дисперсією $\sigma^2(t_2 - t_1)^{2H}$, де $t_2 > t_1$, σ – додатна константа, то є

$$P(\Delta X < x) = \frac{1}{\sqrt{2\pi}\sigma(t_2 - t_1)^H} \int_{-\infty}^x \exp\left(-\frac{1}{2}\left(\frac{u}{\sigma(t_2 - t_1)^H}\right)^2\right) du.$$

На відміну від класичного броунівського руху, прирости якого незалежні, фрактальний броунівський рух з параметром $H \neq \frac{1}{2}$ не володіє цією властивістю.

Нехай $X(t)$ – фрактальний броунівський рух з параметром $H, 0 < H < 1$. Прирости $X(t)$ незалежні тоді і тільки тоді, коли $H \neq \frac{1}{2}$. А при $H > \frac{1}{2}$, то $X(t) - X(0)$ і $X(t+h) - X(t)$, скоріш за все, мають однакові знаки і функція

$X(t)$ зазвичай зростає у майбутньому, якщо вона зростала у минулому. Якщо ж $H < \frac{1}{2}$, то $X(t) - X(0)$ і $X(t+h) - X(t)$ скоріш за все мають різні знаки, і функція $X(t)$ зазвичай зменшується у майбутньому, якщо вона зменшувалася у минулому. Разом ці факти говорять про те, що фрактальний броунівський рух не являється марківським процесом, за винятком випадку $H = \frac{1}{2}$.

Точне моделювання такого процесу викликає великий інтерес. Найбільш часто використовувані підходи можна розділити на дві категорії. Перший з них, пов'язаний з теоретично точними методами, до сих пір складався тільки з методики матричної факторизації, заснованої на холецькому розкладанні коваріаційної матриці. На жаль, ця методика має складність $O(N^2)$ і вимагає високих обчислювальних ресурсів навіть для помірної довжини траєкторії. Інша категорія складається з неточних методів. Всі перераховані вище способи мають свої особливі недоліки і переваги. Вибір між ними зводиться до компромісу між швидкістю і точністю.

Тому формальна постановка задачі це методами машинного навчання виявити пастки у траєкторії руху частинки, тобто виявити ділянки руху, які являють собою приклад класичного броунівського руху, а не фрактального.

1.3 Постановка задач дослідження

Метою кваліфікаційної роботи являється виявлення областей пасток у двомірній траєкторії руху частинки методами кластеризації машинного навчання.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- вивчити методи машинного навчання і визначити ті, які можуть бути застосовані для вирішення задачі кластеризації фрактальних реалізацій;
- вивчити алгоритми кластеризації і вибрати найбільш відповідний для реалізації;

- обрати інструменти програмної розробки і програмно реалізувати обраний метод;
- вибрати математичну модель двомірної траєкторії руху частинок і програмно її реалізувати;
- побудувати матрицю рекурентності для траєкторії;
- на основі реалізованих алгоритмів, модельної траєкторії та побудованої матриці рекурентності перевірити метод виявлення областей пасток;
- на основі результатів досліджень зробити висновки про проведену роботу.

2 ВИБІР ТА ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ'ЯЗАННЯ

2.1 Моделювання фрактального броунівського руху методом Холецького

У даній роботі проведено виявлення пасток у траєкторії руху частинок, моделювання яких складалося з двох частин: моделювання фрактального броунівського руху (що не вважається пасткою) та класичного броунівського руху (що вважається пасткою).

Якщо моделювання класичного броунівського руху є елементарним та не потребує детального опису та пояснень, то фрактальний броунівський рух був промодельований методом Холецького, на якому варто зупинитися докладніше.

Метод Холецького використовує так званий розклад Холецького коваріаційної матриці. Це означає, що коваріаційна матриця $\Gamma(n)$ може бути записана як $L(n)L(n)'$, де $L(n)$ нижня трикутна матриця розміром $(n+1) \times (n+1)$. Позначаючи елемент (i, j) з $L(n)$ як l_{ij} для $i, j = 0, \dots, n$, $L(n)$ є нижньою трикутною матрицею якщо $l_{ij} = 0$ при $i < j$.

Елементи $L(n)$ можуть бути визначені за допомогою визначення елемента (i, j) з $L(n)L(n)'$ та $\Gamma(n)$, що є рівними при $j \leq i$. Таким чином

$$\gamma(i-j) = \sum_{k=0}^j l_{ik} l_{jk}, j \leq i.$$

Ця рівність скорочується до $\gamma(0) = l_{00}^2$ для $i = j = 0$. Для $i = 1$ отримуємо дві рівності

$$\gamma(1) = l_{10} l_{00}, \quad \gamma(0) = l_{10}^2 + l_{11}^2,$$

що визначають l_{10} та l_{11} . Проаналізувавши це помітно, що l_{ij} не можуть бути залежними від n , $L(n+1)$ може бути обчислена з $L(n)$ додаванням рядка, що визначається як

$$\begin{aligned} l_{n+1,0} &= \frac{\gamma(n+1)}{l_{00}}, \\ l_{n+1,j} &= \frac{1}{l_{jj}} \left(\gamma(n+1-j) - \sum_{k=0}^{j-1} l_{n+1,k} l_{j,k} \right), 0 < j \leq n, \\ l_{n+1,n+1}^2 &= \gamma(0) - \sum_{k=0}^n l_{n+1,k}^2. \end{aligned}$$

З цих формул слідує, що $L(n)$ є унікальною за додатковим обмеженням, що на головній діагоналі лише додатні елементи. Якщо $\Gamma(n+1)$ додатно визначена матриця, то невід'ємність $l_{n+1,n+1}^2$ гарантована, і тоді $L(n+1)$ існує.

Позначимо через $V(n) = (V_i)_{i=0,\dots,n}$ $(n+1)$ -вимірний вектор незалежно однаково нормально розподілених випадкових величин, та побудуємо вектор $V(n+1)$ з $V(n)$ приєднанням стандартно розподіленої нормальної величини. Ідея полягає у симуляції $X(n) = L(n)V(n)$ рекурсивно. Для кожного $n \geq 0$, $X(n)$ має нульове середнє значення та коваріаційну матрицю $Cov(L(n)V(n)) = L(n)Cov(V(n))L(n)' = L(n)L(n)' = \Gamma(n)$.

Підсумовуючи, X_{n+1} можна досить легко змодельовати, коли обчислюється $L(n+1)$:

$$X_{n+1} = \sum_{k=0}^{n+1} l_{n+1,k} V_k.$$

На рис. 2.1 – 2.3 представлені реалізації фрактального броунівського руху, що промодельовані методом Холецкого з різними показниками H .

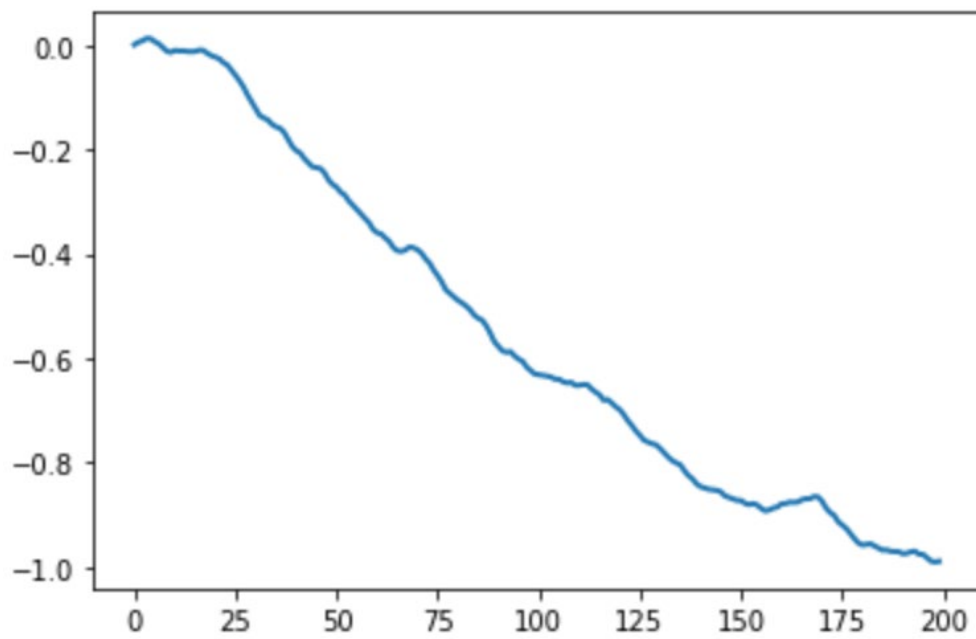


Рисунок 2.1 – Реалізація фрактального броунівського руху, що промодельовані методом Холецького з параметром $H = 0,94$

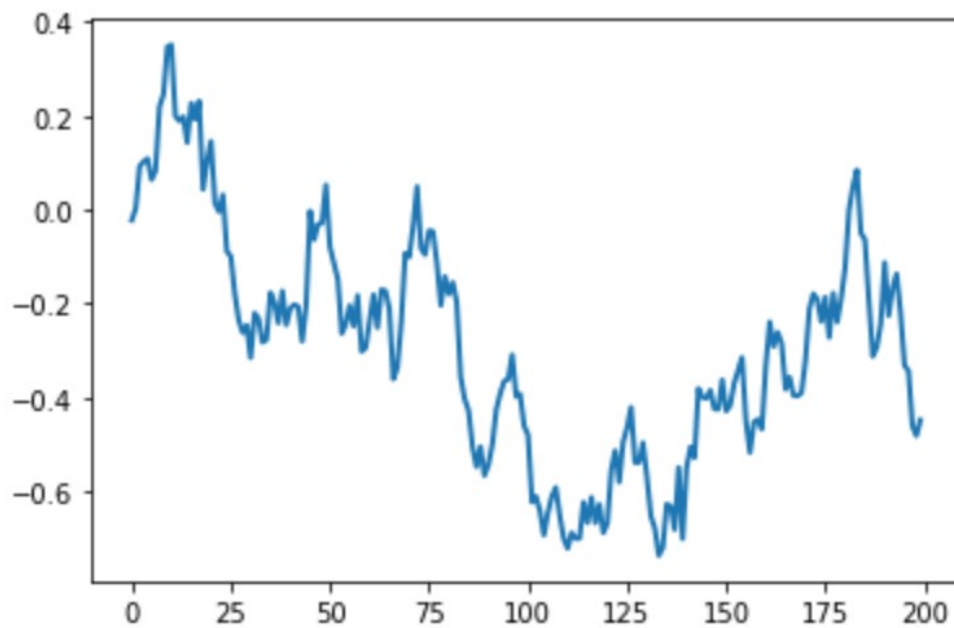


Рисунок 2.2 – Реалізація фрактального броунівського руху, що промодельовані методом Холецького з параметром $H = 0,5$

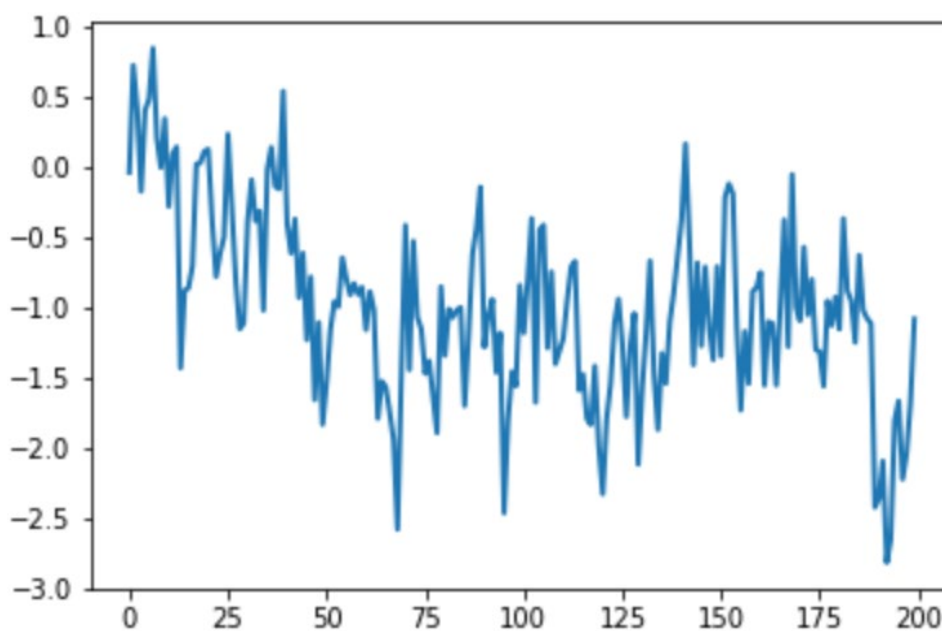


Рисунок 2.3 – Реалізація фрактального броунівського руху, що промодельовані методом Холецького з параметром $H = 0,1$

2.2 Рекурентна матриця

Рекурентні матриці використовуються в різних областях науки. Вони можуть бути використані для реконструкції структури білка і навіть використовуються для виявлення структурних змін реакційних систем. В цілому вони використовуються для кількісної оцінки нелінійних часових рядів, отриманих з динамічних систем, таких як виявлення змін конформації білків в молекулярній динаміці або для кількісної оцінки фізіологічних вимірювань. У контексті динамічних систем показано, що можна реконструювати хаотичний атрактор, пов'язаний з часовим рядом. Крім того, раніше було досліджено вплив спостережних шумів на рекурентні ділянки, і також, що для перевірки стаціонарності часових рядів використовуються рекурентні ділянки.

Хоча концепція рекурентної матриці не є новою, ми будемо її модифікованим способом, який значно обмежує ефект винятків та помилки локалізації. Наша центральна гіпотеза полягає в тому, що пастка в межах траєкторії перекладається в матрицю рецидивів у вигляді квадратної блокової структури по діа-

гоналі матриці рекурентності.

Нехай $r(t_i) = (x(t_i), y(t_i), z(t_i)) \in \mathbb{R}^3$ позначає положення частинки в момент часу t_i , $i = 0, \dots, N$. Припустимо, що частинка змінює динаміку в залежності від речовини, в якій вона рухається. Тоді в спостережуваній траєкторії ми можемо розрізнити проміжки часу, де частинка рухається повільніше (є у пастці) або швидше (вільна). Ми прагнемо розробити метод, який дозволяє визначати ці проміжки часу.

Почнемо з побудови матриці рекурентності $W \in \mathbb{R}^{N+1 \times N+1}$, яка містить інформацію про зв'язок положень частинки на заданій траєкторії. Визначимо елементи цієї матриці як:

$$w_{ij} := \begin{cases} \exp \left(-\frac{\|r(t_i) - r(t_j)\|^2}{\sigma} \right) & |t_i - t_j| < \delta \\ 0 & \text{інакше} \end{cases}$$

для $i, j = 0, \dots, N$. Враховуючи матрицю рекурентності W , вводимо вектор $f \in \mathbb{R}^{N+1}$ компоненти якого визначені як $f_i := \sum_{j=0}^N w_{ij}$. Компонента f_i вектору f містить інформації про з'єднання положення конкретної компоненти в момент часу t_i зі своїм положенням часом t_j , $j = 0, \dots, N$. Більша цінність f_i , тим вище ймовірність того, що i -те положення належить фрагменту траєкторії захопленої частинки [17].

Без обмеження $|t_i - t_j| < \delta$ матриця рекурентності W показувала б близькість за мірою відстані між усіма точками траєкторії, що не є доцільним для визначення пастки. Оскільки ми вважаємо за пастку ділянку траєкторії руху частинки, де вона зменшує швидкість, і як результат, звісно, зменшує переміщення, а тобто збільшує показник близькості з найближчими за часом сусідніми координатами. Тобто нас цікавить лише ті ділянки руху, де частинка знахо-

диться увесь відрізок часу пастки, без навіть тимчасового виходу з неї [18].

Звісно, що δ та σ є параметрами, і їх підбір залежить від довжини траєкторії за часом, від параметрів розподілу, що відповідають цій траєкторії, та від суб'єктивної оцінки дослідника.

Оскільки значення рекурентної матриці знаходяться між 0 та 1, доцільно використовувати для її показу псевдоколірний графік. Такий графік відображає матричні дані як масив кольорових клітин.

Для простої ілюстрації принципу використання псевдоколірного графіку,

відобразимо матрицю $A = \begin{pmatrix} 1 & 0,42 & 0,41 & 0,62 & 0,11 \\ 0,42 & 1 & 0,84 & 0,26 & 0,05 \\ 0,41 & 0,84 & 1 & 0,26 & 0,04 \\ 0,62 & 0,26 & 0,26 & 1 & 0,17 \\ 0,11 & 0,05 & 0,04 & 0,17 & 1 \end{pmatrix}$ у такий спосіб, як

показано на рис. 2.4:

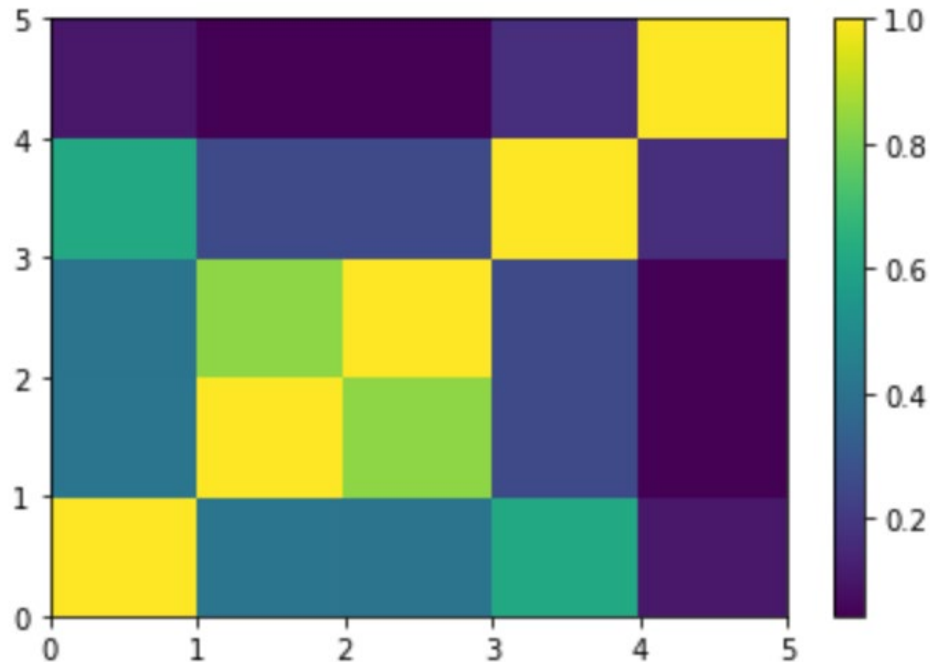


Рисунок 2.4 – Псевдоколірний графік матриці A

На рис. 2.5 представлений зразок псевдоколірного графіку.

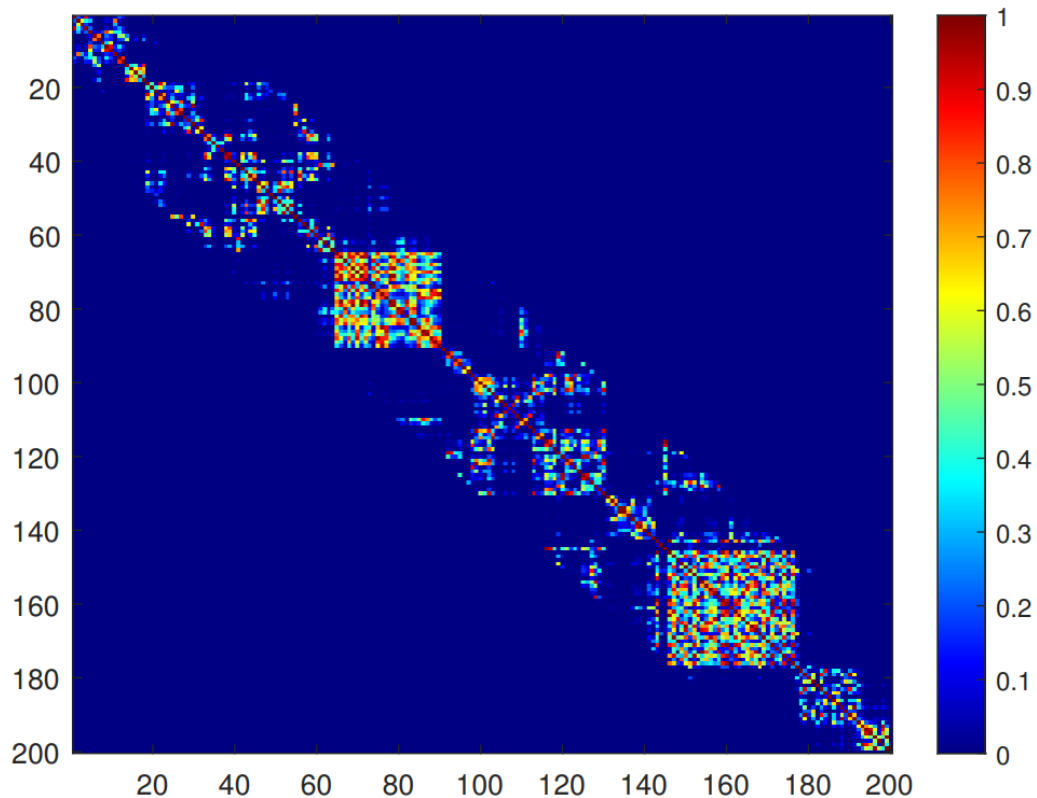


Рисунок 2.5 – Зразок псевдоколірного графіку

2.3 Метод DBSCAN

Алгоритм DBSCAN розглядає кластери як області високої щільності, розділені областями низької щільності. Завдяки цьому досить загальному погляду, кластери, знайдені DBSCAN, можуть мати будь-яку форму, на відміну від K-means, які припускають, що кластери мають опуклу форму.

Центральним компонентом DBSCAN є концепція корових проб, тобто проб, що знаходяться в зонах високої щільності. Таким чином, кластер – це набір основних зразків, кожен з яких знаходиться близько один до одного і набір неосновних зразків, які близькі до основного.

Будь-який базовий зразок за визначенням є частиною кластера. Алгоритм вважає будь-який зразок, який не є основним зразком і знаходиться на відстані принаймні ϵ від будь-якого основного зразка, базовим.

У той час як параметр `min_samples` головним чином контролює, наскільки

алгоритм толерантний до шуму (на галасливих і великих наборах даних може бути бажаним збільшити цей параметр), параметр ϵ має вирішальне значення для правильного вибору для набору даних і функції відстані, і зазвичай його не можна залишати за значенням за замовчуванням. Він контролює локальне оточення точок. Якщо вибрати занадто малий розмір, більшість даних взагалі не буде кластеризовано (і позначено як -1 для «шуму»). Якщо вибрано надто великий, це призводить до об'єднання близьких кластерів в один кластер і, зрештою, весь набір даних повертається як один кластер.

Алгоритм DBSCAN є детермінованим, він завжди генерує однакові кластери, коли їм надаються однакові дані в тому самому порядку. Однак результати можуть відрізнятися, якщо дані надаються в іншому порядку. По-перше, незважаючи на те, що основні зразки завжди призначатимуться тим самим кластерам, мітки цих кластерів залежатимуть від порядку, у якому ці зразки зустрічаються в даних. По-друге, і що більш важливо, кластери, до яких призначаються неосновні зразки, можуть відрізнятися залежно від порядку даних. Це станеться, якщо неосновний зразок має відстань меншу за ϵ до двох основних зразків у різних кластерах. Відповідно до трикутної нерівності, ці два зразки ядра повинні бути більш віддалені один від одного, ніж ϵ , інакше вони будуть в одному кластері. Неосновна вибірка призначається тому кластеру, який буде згенерований першим під час проходження даних, тому результати залежатимуть від порядку даних [19].

Алгоритм кластеризації DBSCAN виконується покроково:

а) випадковим чином обирається зразок з набору даних, який ще не відноситься до будь якого кластеру або вважається відхиленням, при цьому радіусом ϵ визначається окіл;

б) якщо окіл цього зразка містить хоча б min_samples інших зразків, тоді його починаємо вважати коровим, та починається формування кластеру, в іншому випадку цей зразок позначається як шум;

в) усі зразки, що входять в окіл ϵ корового, стають елементи кластеру, який сформував цей зразок;

г) при потраплянні нового зразка у кластер, від нього будується окіл ϵ , і нові зразки стають також елементами кластеру, якщо потрапили в цей окіл;

д) пролонгуємо додавання нових зразків до кластеру, поки знаходяться нові корові зразки;

е) пролонгуємо обирання зразків та формування нових кластерів, доки всі зразки не будуть визначенні.

На рис. 2.6 проілюстрована робота алгоритму.

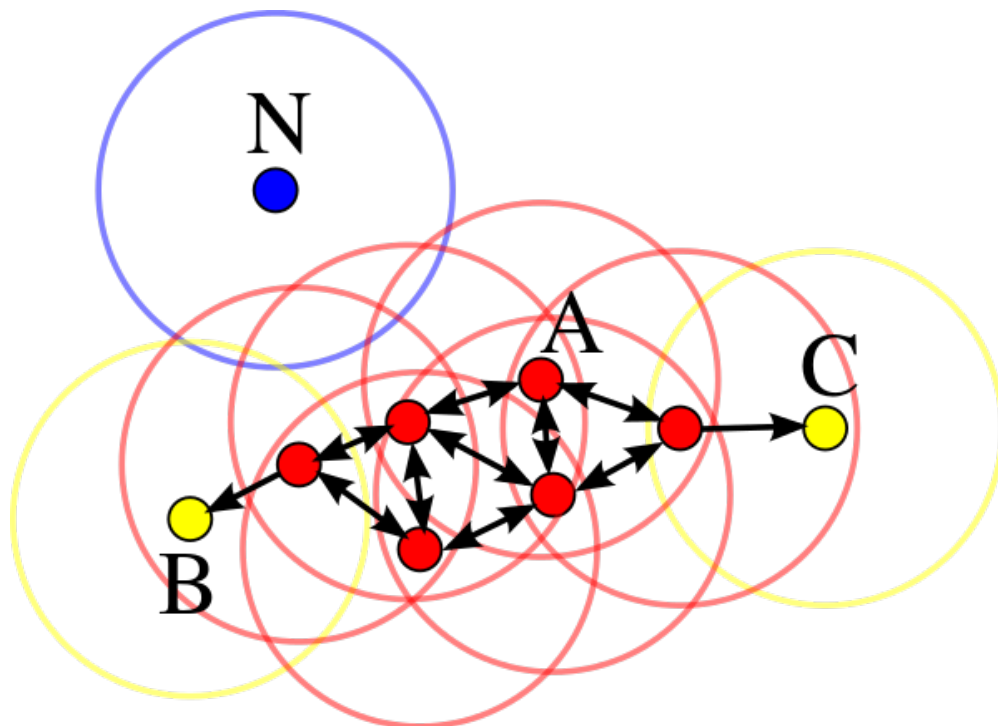


Рисунок 2.6 – Ілюстрація роботи алгоритму

З головних переваг методу DBSCAN можна визначити те, що не потрібно вказувати апріорну кількість кластерів, і знаходження кластерів довільної форми.

Щодо недоліків, то DBSCAN не є детерміністичним (точки на межі, які досяжні з декількох кластерів, можуть належати одному або іншому кластеру в залежності від порядку обробки даних) та неможливість кластеризувати набори даних з великим перепадом щільностей.

На рис. 2.7 представлені приклади використання алгоритму DBSCAN на різних наборах даних.

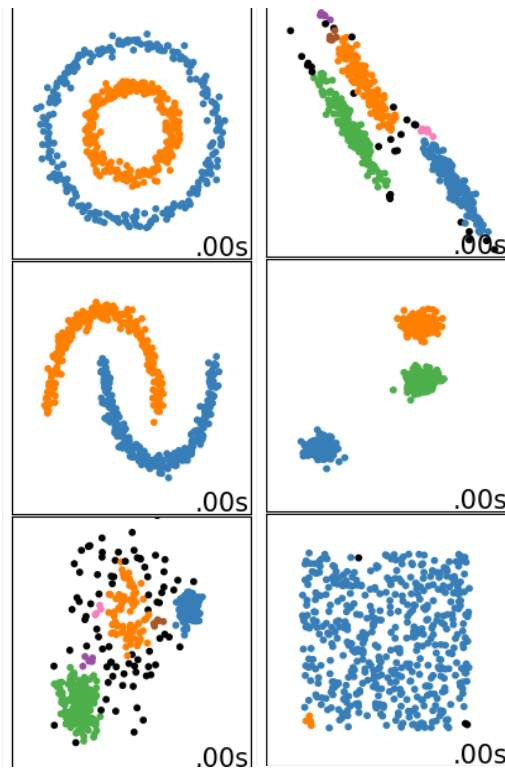


Рисунок 2.7 – Приклади використання алгоритму DBSCAN
на різних наборах даних

2.4 Метод K-means

K-means – метод кластеризації даних, що базується на розподіленні даних на кілька груп однакової дисперсії. Він добре масштабується для великої кількості зразків і використовується в широкому діапазоні областей застосування в багатьох різних галузях.

Алгоритм K-means розбиває набір X з N вибірок на K непересічних кластерів C , кожен з яких описується середнім значенням вибірок у кластері. Середні зазвичай називають «центроїдами» кластера; зауважте, що вони, загалом, не є точками з X , хоча вони живуть в одному просторі.

K-means має на меті вибрати центроїди, які мінімізують інерцію або критерій суми квадратів у кластері.

На першому кроці вибираються початкові центроїди, а найпростішим методом є вибір зразків із набору даних. Після ініціалізації K-means складається з

циклу між двома іншими кроками. Перший крок призначає кожному зразку найближчий центроїд. На другому етапі створюються нові центроїди, беручи середнє значення всіх вибірок, призначених кожному попередньому центроїду.

Різниця між старим і новим центроїдами обчислюється, і алгоритм повторює останні два кроки, доки це значення не стане меншим за порогове значення [20].

Основний тип задач, які розв'язує алгоритм K-means, це наявність припущень (гіпотез) щодо кількості кластерів, при цьому вони мають бути різні настільки, наскільки це можливо. Вибір числа k може базуватися на результатах попередніх досліджень, теоретичних міркуваннях або інтуїції.

Алгоритм кластеризації K-means виконується покроково:

- а) задається кількість кластерів, що необхідно утворити;
- б) випадковим чином обирається k зразків, які на цьому кроці вважаються центроїдами кластерів;
- в) кожен зразок починає відноситися до одного з кластерів — того, міра відстані до якого найменша;
- г) розраховується нова центроїда кожного кластера як зразок, міра якого розраховуються як середнє арифметичне відстані зразків, що входять у цей кластер;
- д) пролонгуємо кроки 3-4, поки центроїди кластерів стануть стійкими (при кожній ітерації в кожному кластері опинятимуться ті самі зразки), дисперсія всередині кластера буде мінімізована, а між кластерами — максимізована.

Перевагами цього методу є простота використання, швидкість використання, зрозумілість і прозорість алгоритму. Недоліки: алгоритм занадто чутливий до викидів, повільна робота з великими обсягами даних, необхідно задавати кількість кластерів. На рис. 2.8 представлені приклади використання алгоритму K-means на різних наборах даних.

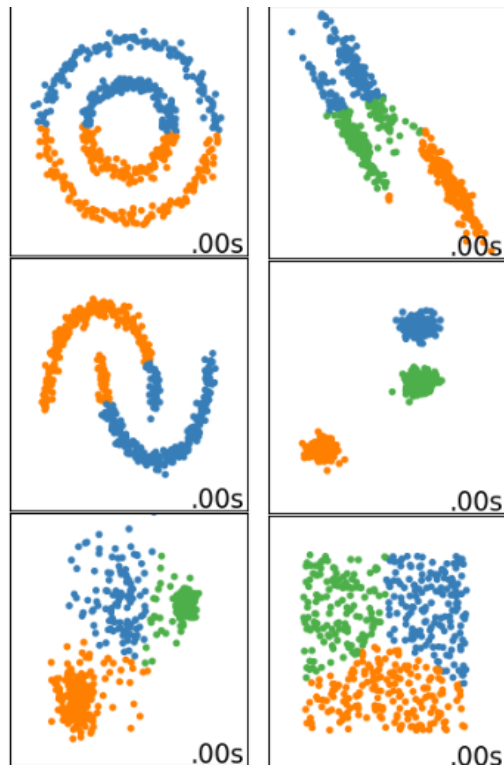


Рисунок 2.8 – Приклади використання алгоритму
K-means на різних наборах даних

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНОГО ЕКСПЕРИМЕНТУ

3.1 Python та машинне навчання

Python – інтерпретована об'єктно-орієнтована мова програмування високого рівня.

Розробка Python розпочалася в кінці 1980-х років, а перша повноцінна версія з'явилася у 1991 році. Звісно, за цей час мова широко розвинулася, і найголовнішою перевагою є велика кількість модулів, використання яких дає розробникам незлічену кількість можливостей для реалізації проєктів.

Під час виконання кваліфікаційної роботи було використано декілька модулів, на короткому описі та використанні яких є сенс зупинитися. Це такі модулі:

а) Numpy – модуль мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих математичних функцій для операцій з цими масивами. Також важливо відмітити, що саме цей модуль дозволяє генерувати масив випадкових значень за заданим законом розподілення, а це, в свою чергу, є базою цього дослідження;

б) Matplotlib – модуль мови програмування Python для візуалізації даних двовимірною 2D графікою та 3D графікою. Звісно, цей модуль використовувався для показу графіків траєкторій руху частинок та для відображення псевдоколірних графіків;

в) Math – один з найважливіших модулів у Python, оскільки дає можливість реалізовувати обширну роботу з числами та математичними функціями. Безпосередньо у цій роботі він був використаний для розрахунку значень рекурентної матриці;

г) Sklearn – модуль мови програмування Python, який надає функціонал для створення та тренування алгоритмів машинного навчання як кластеризації, так і інших видів (класифікації, регресії, модельної селекції тощо). Цей модуль

можна вважати досить «молодим», оскільки перша версія була написана у 2007 році Девідом Корнапе. Наприкінці січня 2010 року було опубліковано перший публічний реліз (бета-версія 0.1). У наш час, з травня 2022 року, діє версія `scikit-learn 1.1.0`.

Тому, виходячи з цих фактів, для програмної реалізації задачі кластеризації фрактальних реалізацій на основі машинного навчання була обрана мова програмування Python, оскільки її інструментарій повністю задовольняє цим потребам.

3.2 Опис роботи програми

Робота програми починається з генерації двовірного броунівського руху частинки. Для наочності за більш зрозумілої інтерпретації результатів була обрана така структура траєкторії: перші 100 кроків моделюються як фрактальний броунівський рух методом Холецького, наступні 50 моделюються як класичний броунівський рух і так далі, поки траєкторія не сягне 400 кроків.

При моделюванні частин фрактального руху обирається показник Херста та використовуються однакові параметри усі три рази, для моделювання двох пасток – також параметри були однакові.

Наступним кроком вираховується рекурентна W матриця розміром 400×400 , де близькість за мірою відстані має не нульове значення для тих, точок траєкторії, відстань у часі для яких менша за 50, тобто $\delta = 50$, а параметр $\sigma = 1$.

Далі необхідно побудувати набір даних, який буде кластеризуватись. Таким набором даних є матриця M розміром $k \times 2$, яка будується за таким принципом: перебираємо всі елементи рекурентної матриці W , якщо цей елемент більший за значення 0,6 та менший за 1, тоді в матрицю M заносимо його положення у матриці W . Інакше кажучи, у матриці M зберігаємо пари координат траєкторії за номером по порядку (тобто за номером у часі), міра зв'язку між

якими більша за значення 0,6, а умова, що ця міра має бути меншою за 1, відсікає елементи, що показують зв'язок координати з самою собою, що не має ніякого сенсу.

Тепер проведемо кластеризацію даних матриці M методом K-means, число кластерів дорівнює 5, так як змодельована траєкторія складається з п'яти частин.

Далі проводимо кластеризацію цих самих даних методом DBSCAN, обираючи $\text{min_samples} = 40$, $\text{eps} = 15$. Параметр min_samples був обраний з міркувань доцільності, оскільки ми не вважаємо доцільним зараховувати за пастку частину траєкторії, де частинка блукала менше 20 кроків (значення 40, бо матриці M і W є симетричними), параметр eps був обраний емпіричним шляхом.

Останнім кроком проводиться оцінка правильності визначення пасток методом DBSCAN за простою формулою $\frac{n}{m}$, де m — кількість положень в траєкторії руху, які алгоритм визначив як пастки, n — кількість положень з m , що були змодельовані як пастки.

4 РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНОГО ЕКСПЕРИМЕНТУ ТА ЇХ АНАЛІЗ

На рис. 4.1 наведена змодельована траєкторія руху частинки з параметрами $H = 0,7$ для фрактального руху і $\sigma = 0,5$ для класичного. Синім кольором позначені ділянки, що відповідають фрактальному броунівському руху (тобто частинка вільною в ці моменти), а зеленим та жовтим – ділянки класичного броунівського руху (дві пастки). На рис. 4.2 та 4.3 наведені траєкторії з параметрами $H = 0,4$ та $H = 0,9$.

На рис. 4.4 та 4.5 зображені одновимірні траєкторії руху для $H = 0,7$ (по осі x та осі y відповідно), позначення у кольорах залишається таким, як і на рисунку 4.1. Для $H = 0,4$ та $H = 0,9$ аналогічні дані показані на рис. 4.6-4.7 та 4.8-4.9.

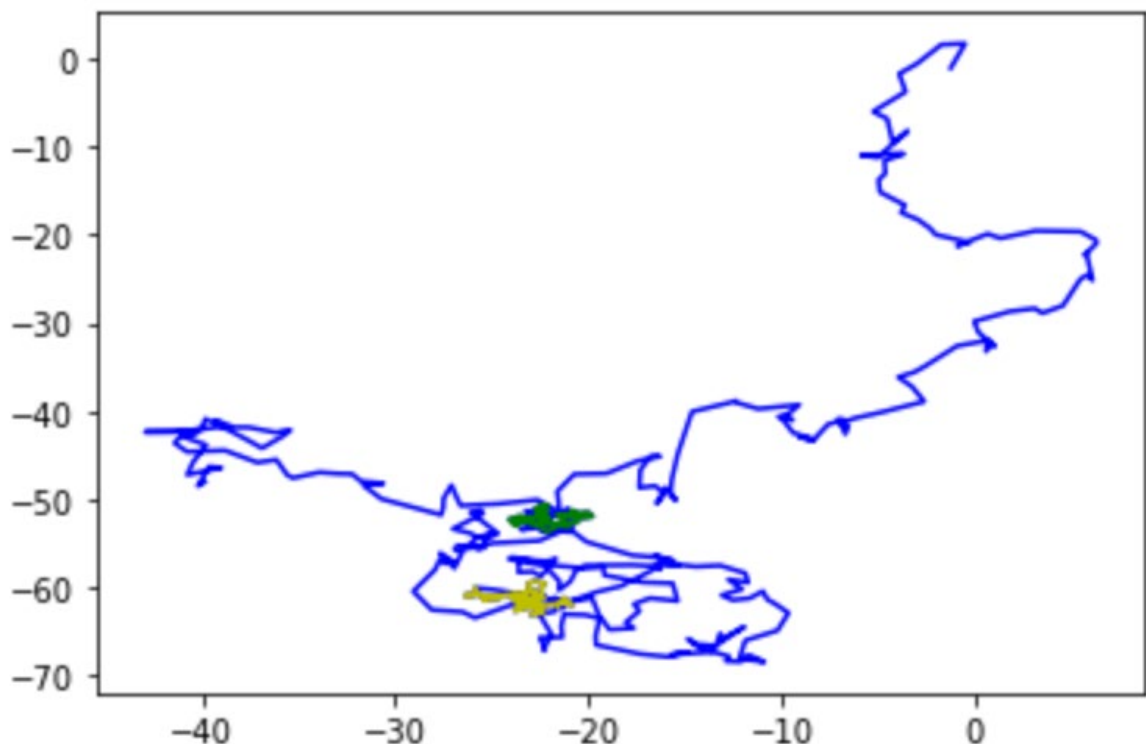


Рисунок 4.1 – Змодельована траєкторія руху частинки $H = 0,7$

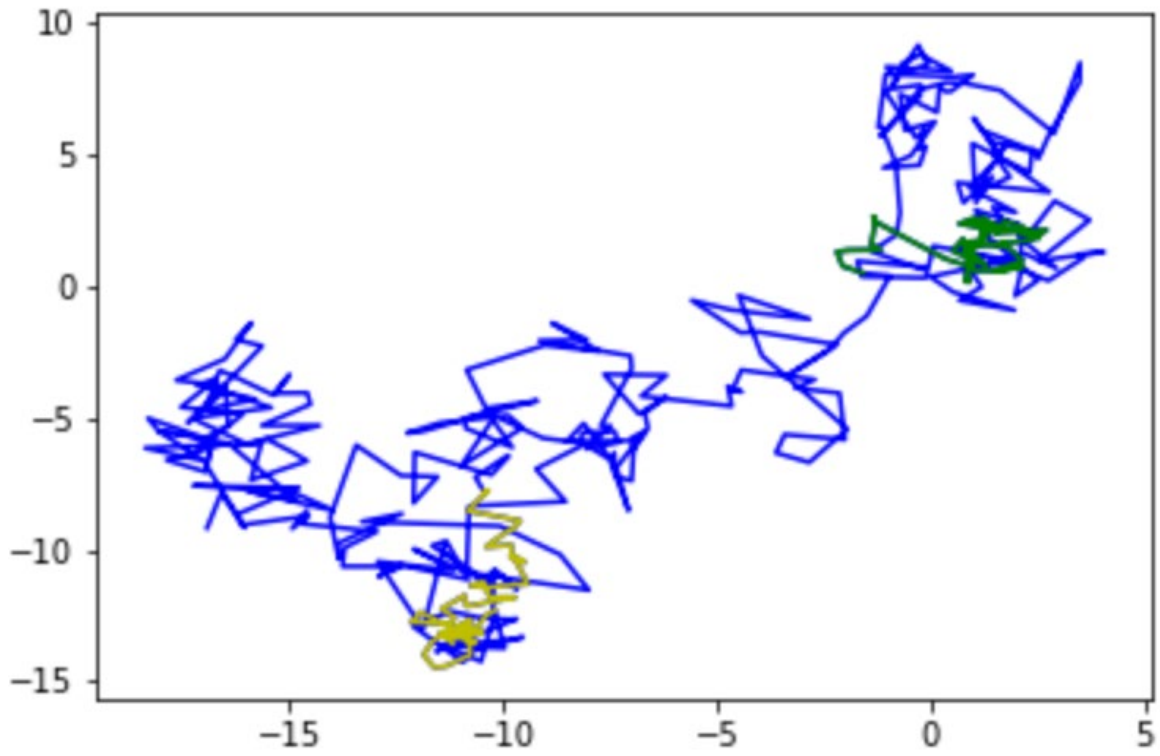


Рисунок 4.2 – Змодельована траєкторія руху частинки $H = 0,4$

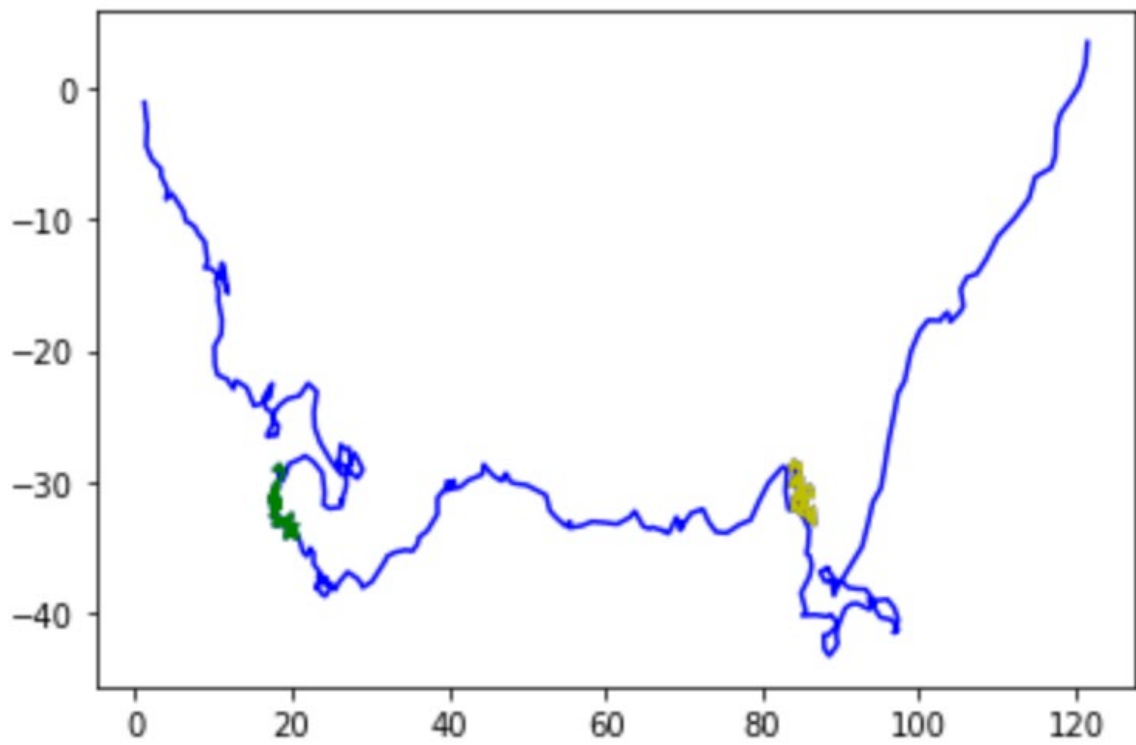


Рисунок 4.3 – Змодельована траєкторія руху частинки $H = 0,9$

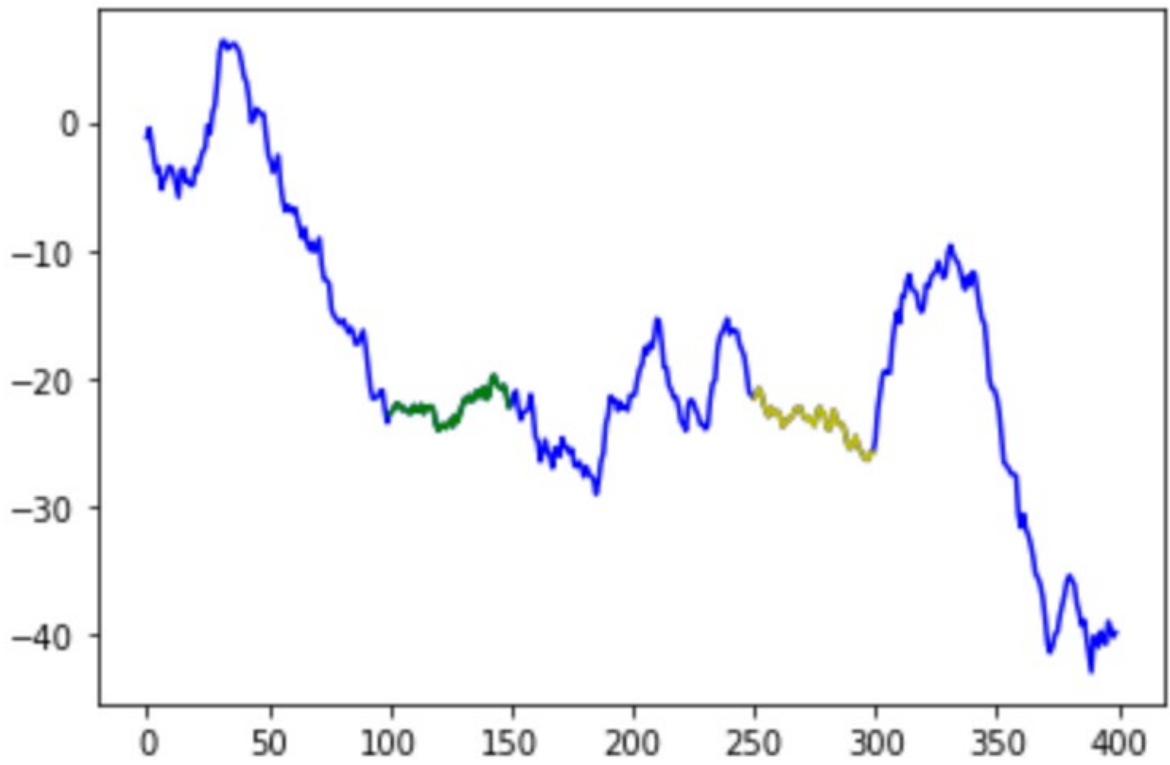


Рисунок 4.4 – Змодельована траєкторія руху
частинки по осі x для $H = 0,7$

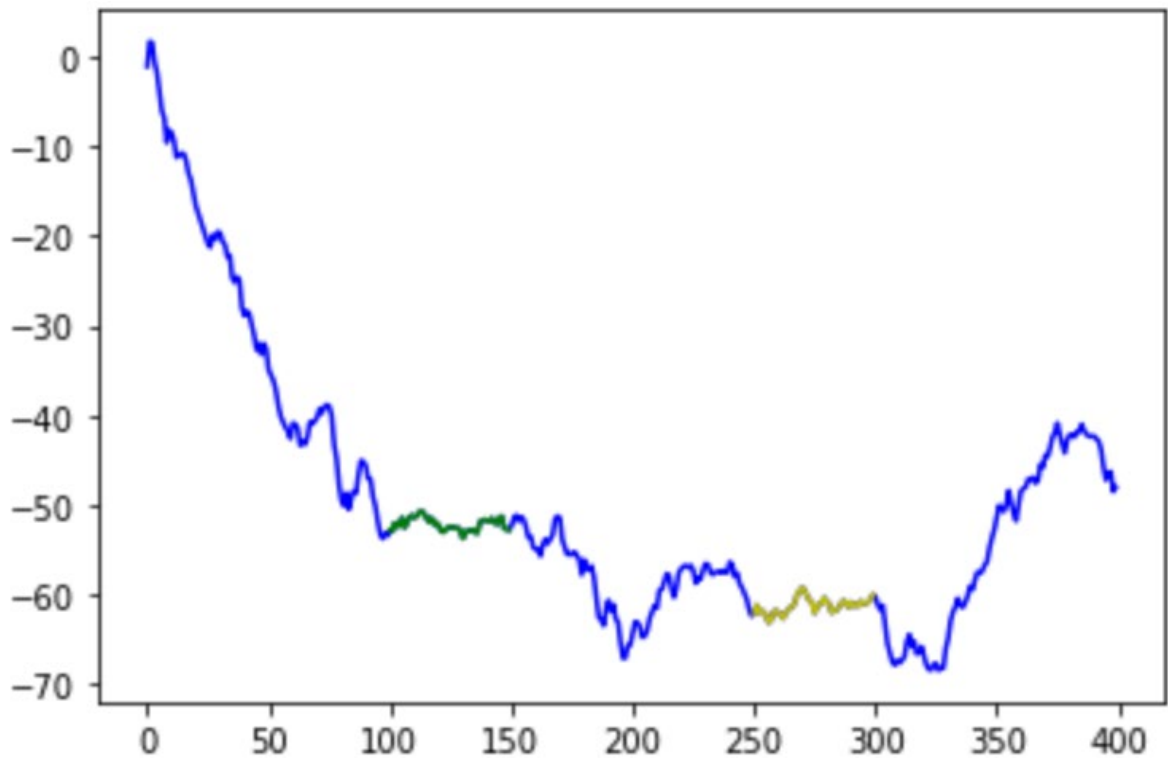


Рисунок 4.5 – Змодельована траєкторія руху
частинки по осі y для $H = 0,7$

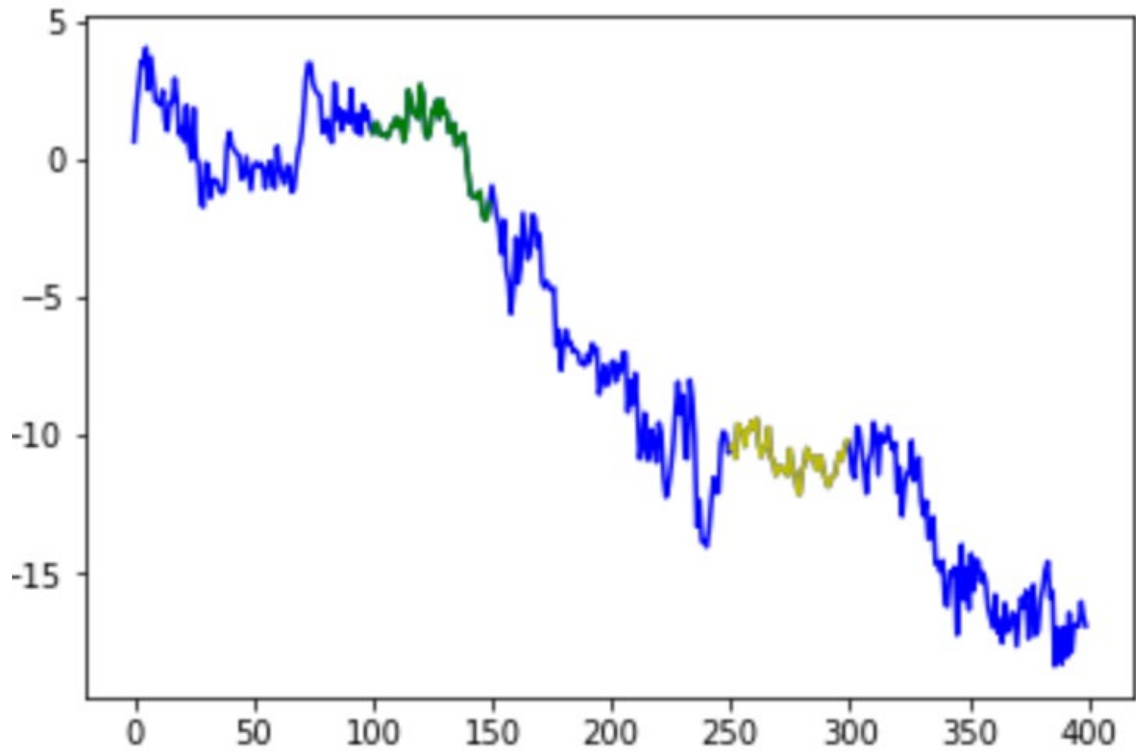


Рисунок 4.6 – Змодельована траєкторія руху
частинки по осі x для $H = 0,4$

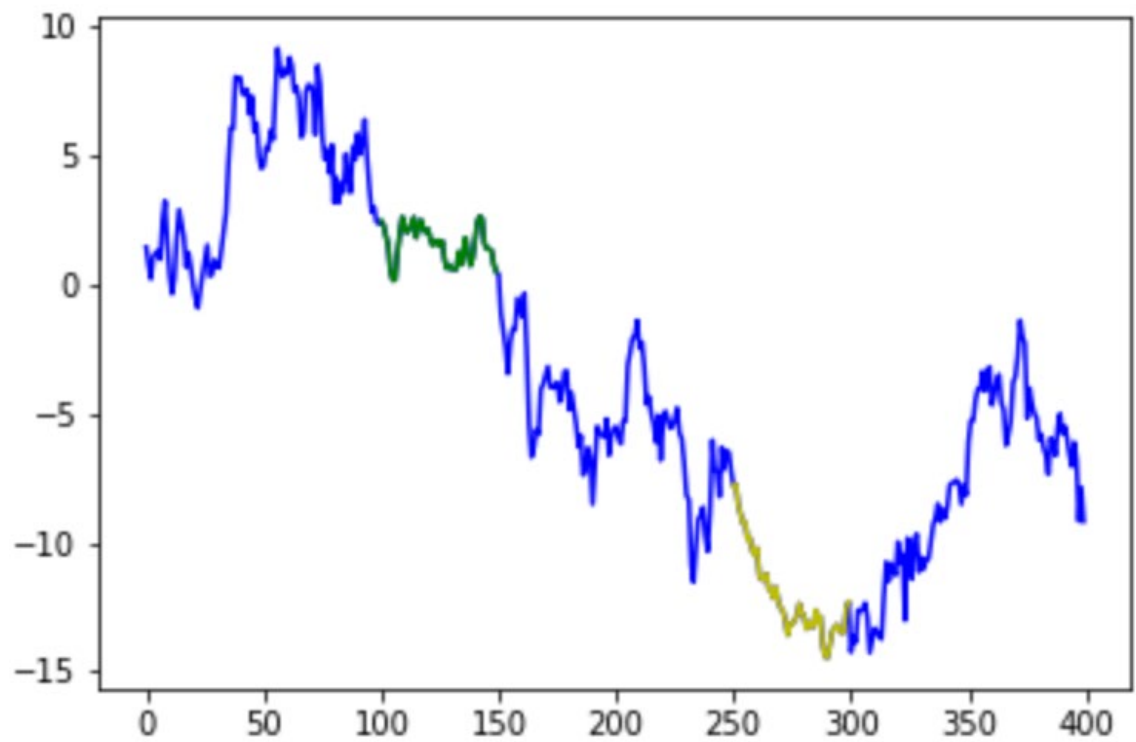


Рисунок 4.7 – Змодельована траєкторія руху
частинки по осі y для $H = 0,4$

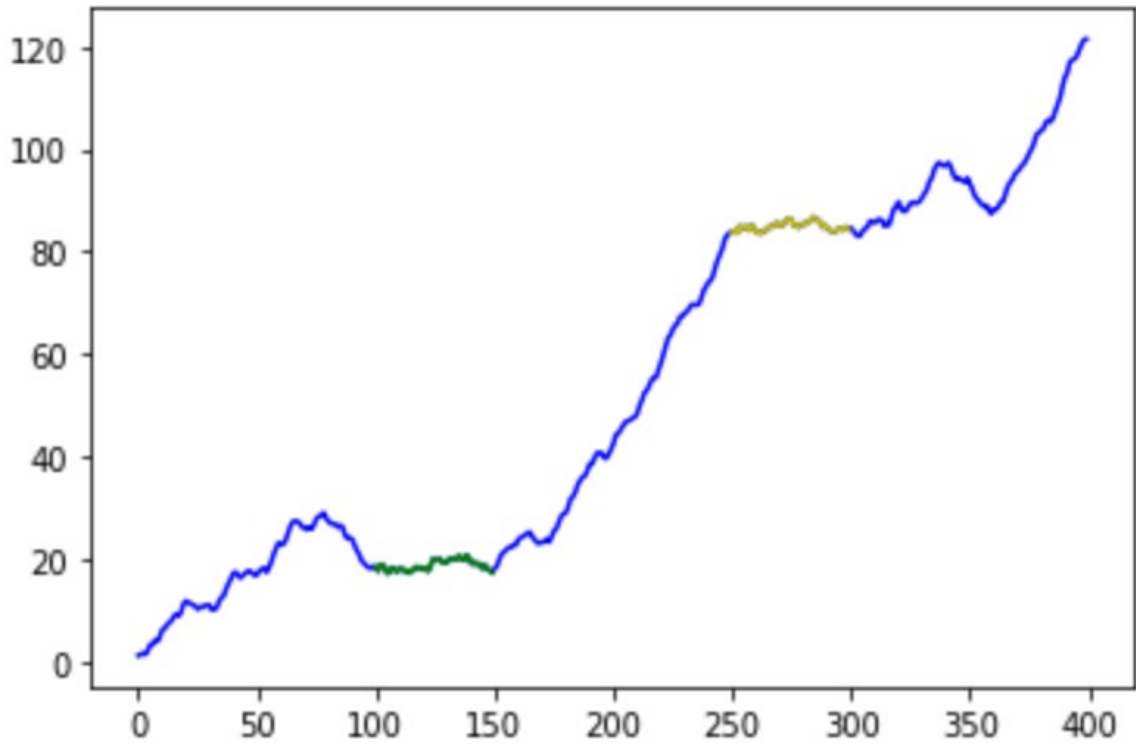


Рисунок 4.8 – Змодельована траєкторія руху
частинки по осі x для $H = 0,9$

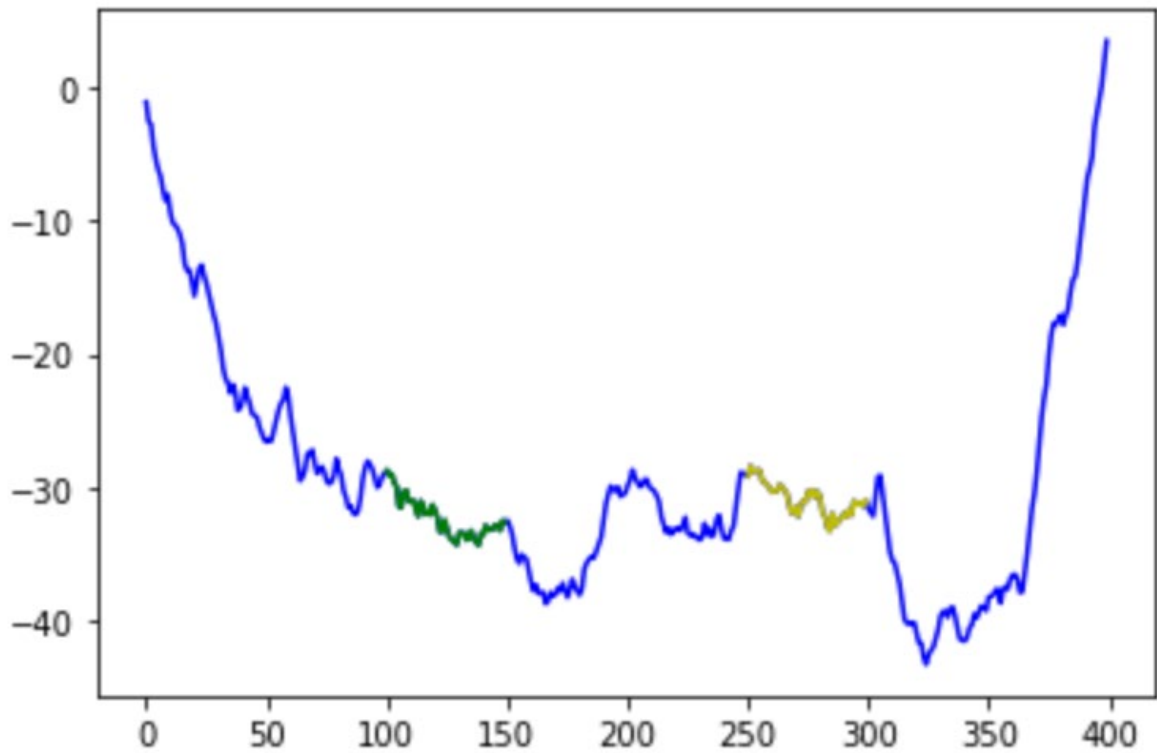


Рисунок 4.9 – Змодельована траєкторія руху
частинки по осі y для $H = 0,9$

Далі псевдоколірно відображаємо матриці рекурентності. На рис. 4.10 зображений таких графік для показника $H = 0,7$, на рис. 4.11 – $H = 0,4$, на рис. 4.12 – $H = 0,9$.

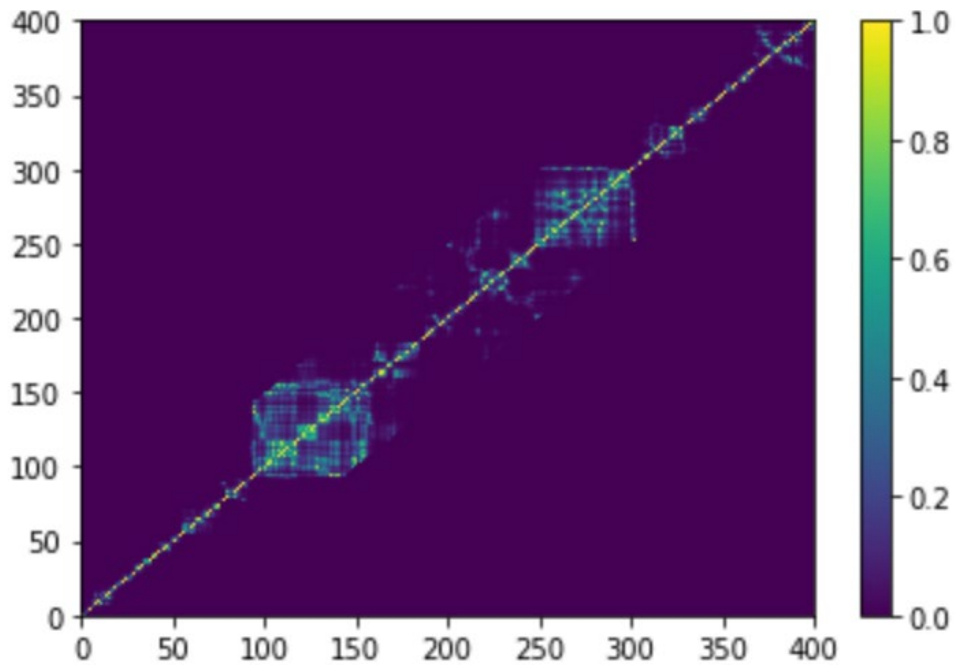


Рисунок 4.10 – Псевдоколірний графік матриці рекурентності для $H = 0,7$

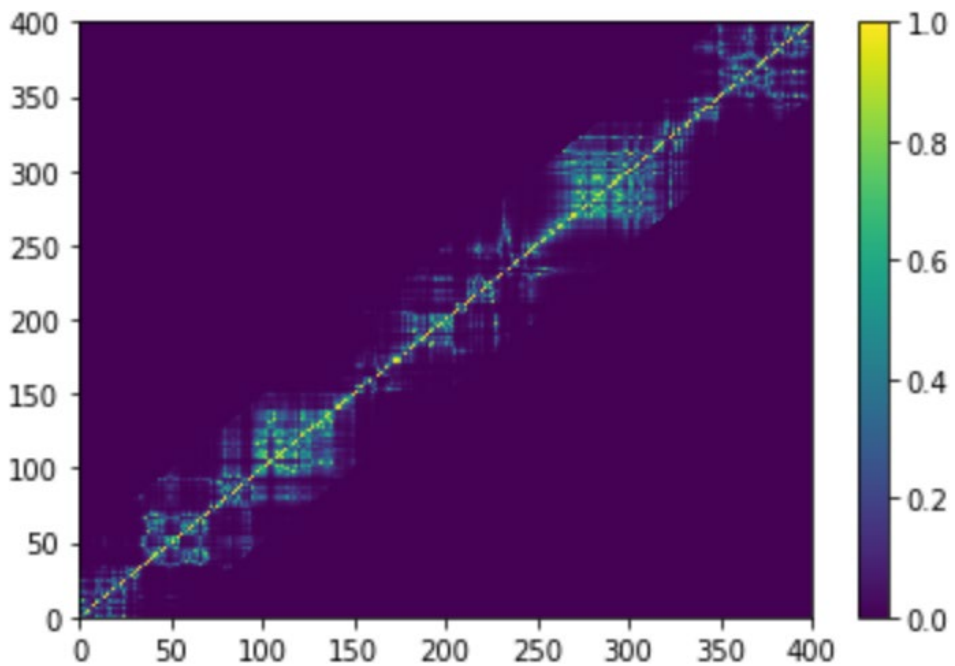


Рисунок 4.11 – Псевдоколірний графік матриці рекурентності для $H = 0,4$

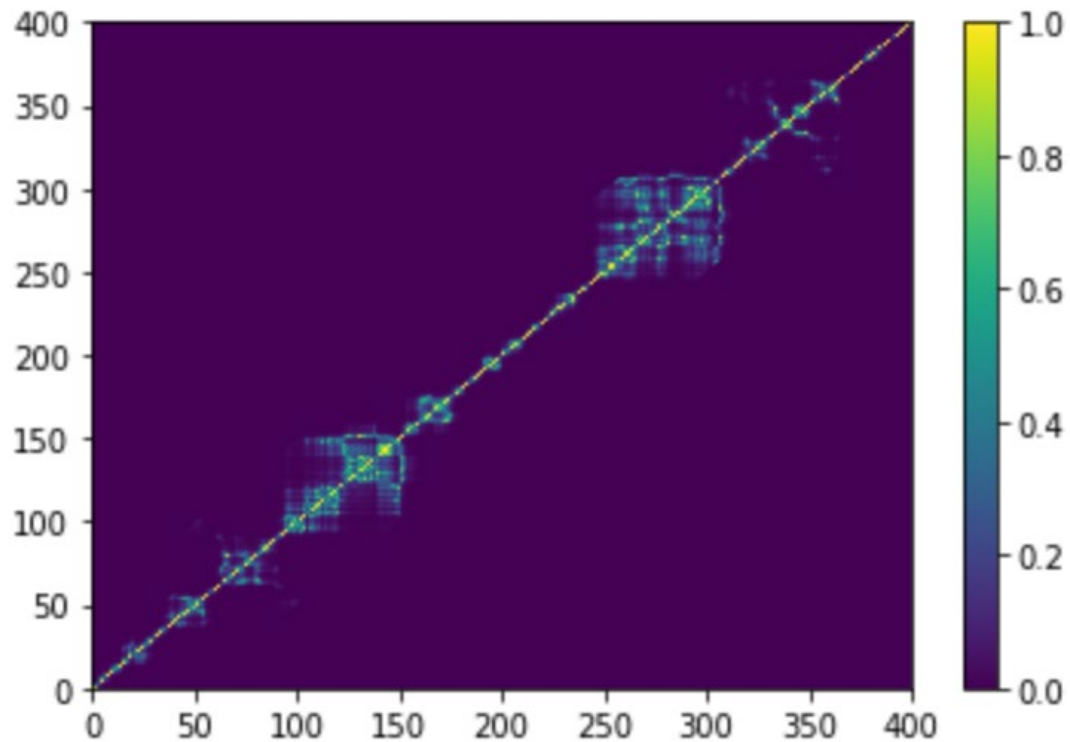


Рисунок 4.12 – Псевдоколірний графік матриці
рекурентності для $H = 0,9$

Вже по цим графікам можна робити передбачення, що пастки є частинами траєкторії руху, що відповідають моментам 100-150 та 250-300 у часі.

Тепер проведемо формуємо матриці M для кожного обраного випадку показника H та кластеризуємо ці дані двома методами (K-means та DBSCAN).

На рис. 4.13-4.15 показані результати кластеризації методом K-means відповідно до різних $H = 0,7$, $H = 0,4$ та $H = 0,9$.

Далі на рис. 4.16-4.18 показані результати кластеризації методом DBSCAN відповідно до вищевказаних значеннях H .

По цим рисункам можна зробити попередні висновки, що DBSCAN досить точно знайшов пастки у траєкторіях.

Таким чином були проведені по 50 моделювань траєкторій для декількох обраних значень параметру H та була вирахована середня оцінка. Результати представлені в таблиці 4.1.

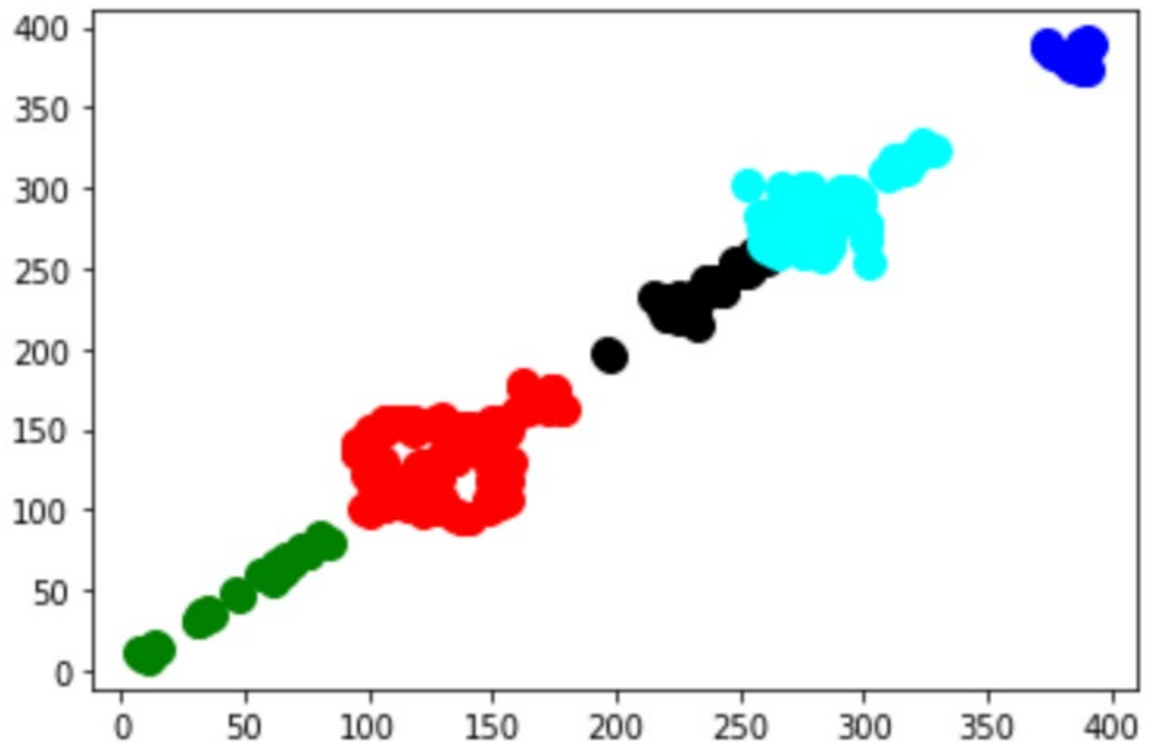


Рисунок 4.13 – Результат кластеризації методом
К-means для $H = 0,7$

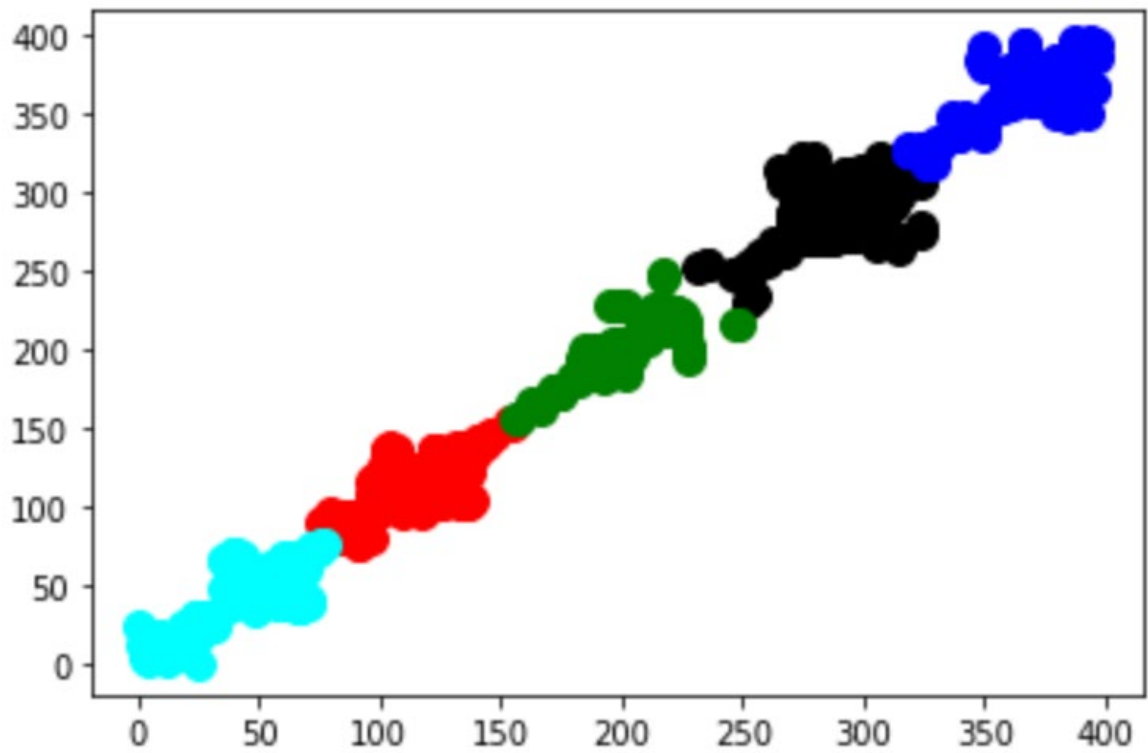


Рисунок 4.14 – Результат кластеризації методом
К-means для $H = 0,4$

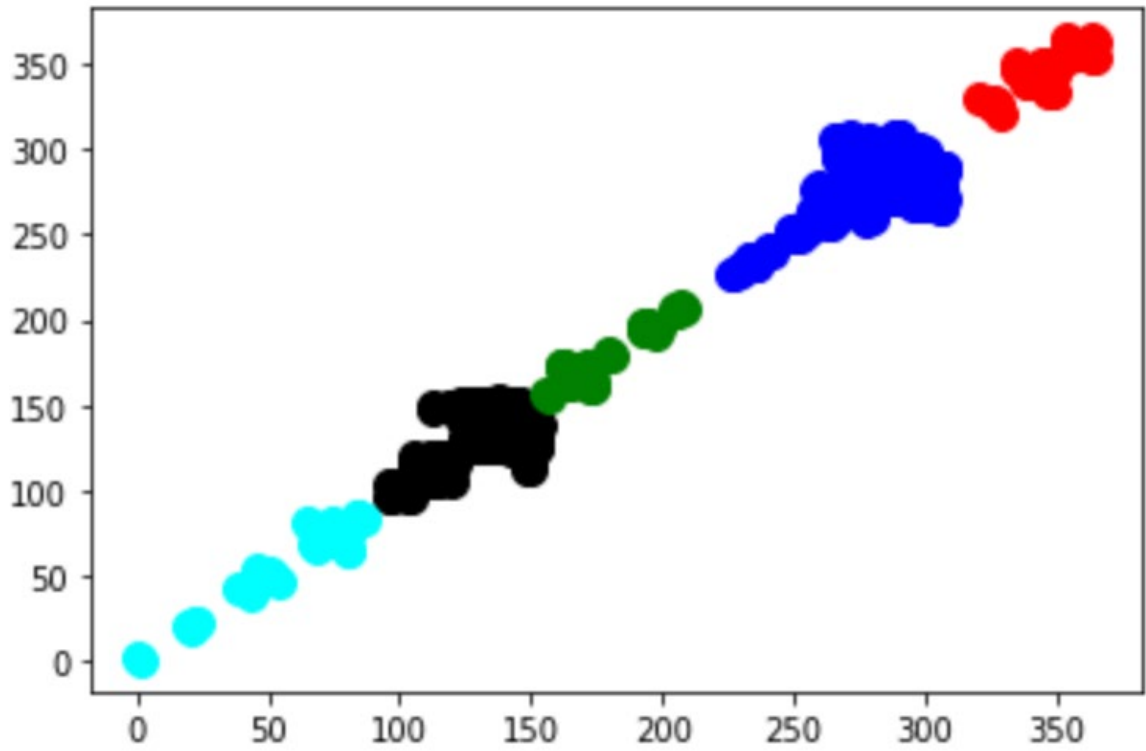


Рисунок 4.15 – Результат кластеризації методом
K-means для $H = 0,9$

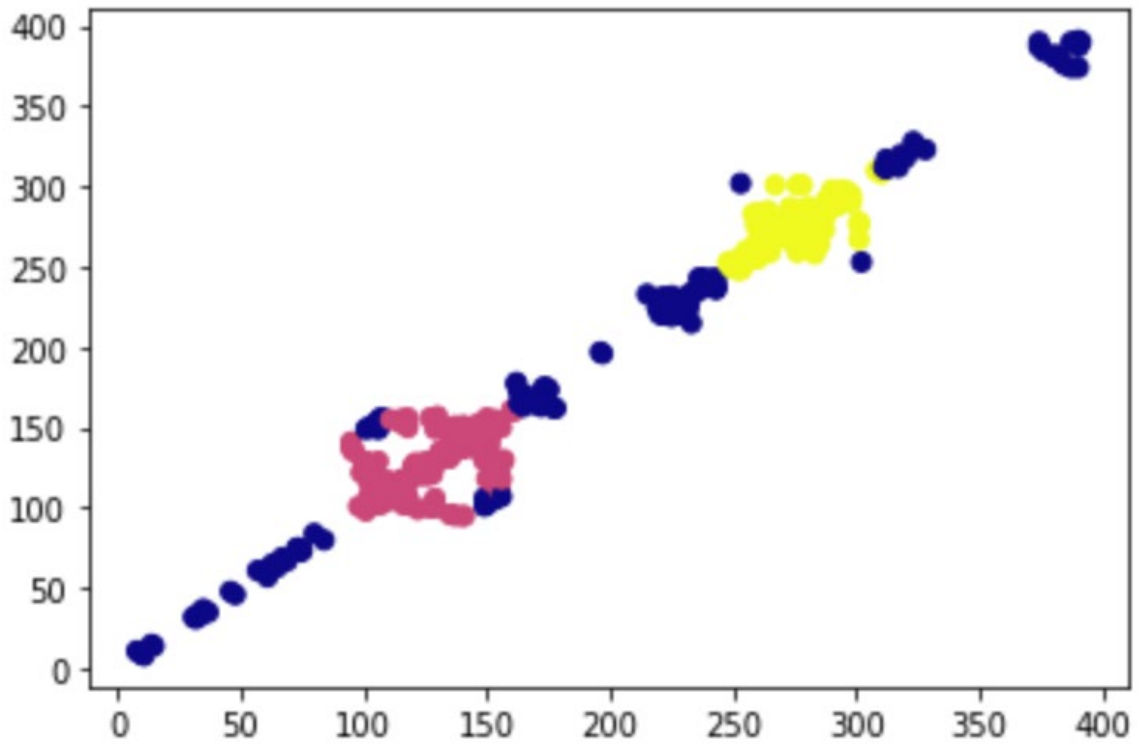


Рисунок 4.16 – Результат цієї кластеризації методом
DBSCAN для $H = 0,7$

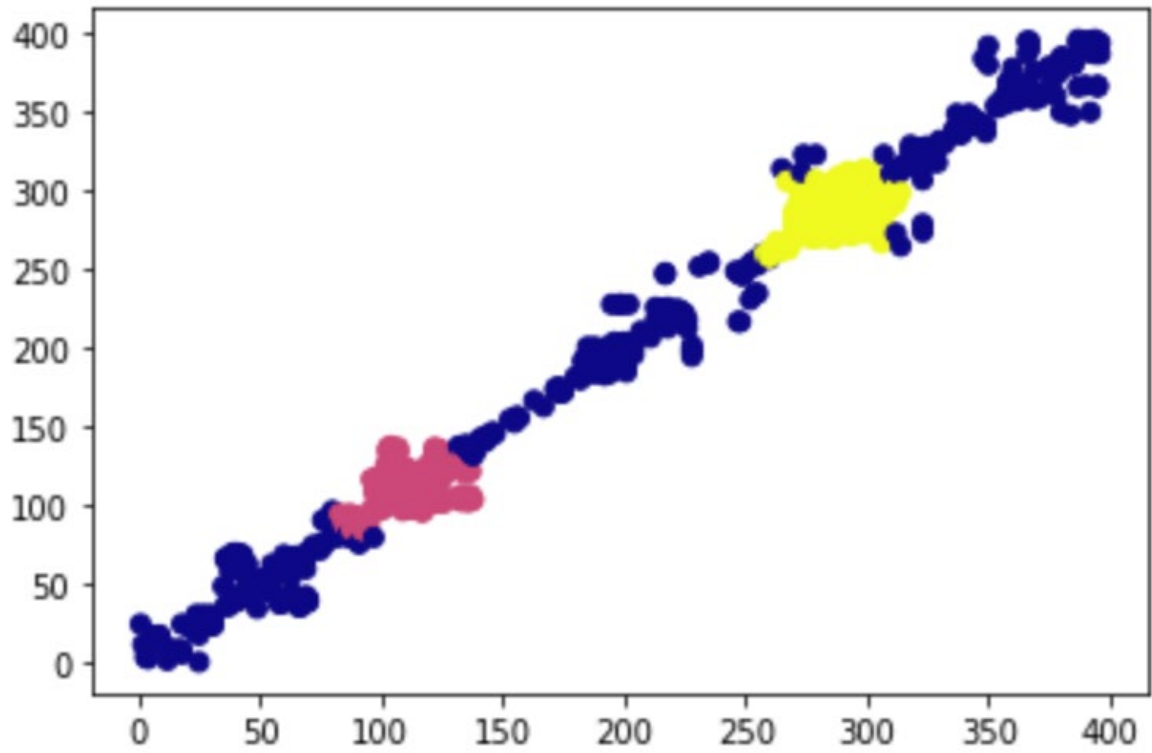


Рисунок 4.17 – Результат цієї кластеризації методом
DBSCAN для $H = 0,4$

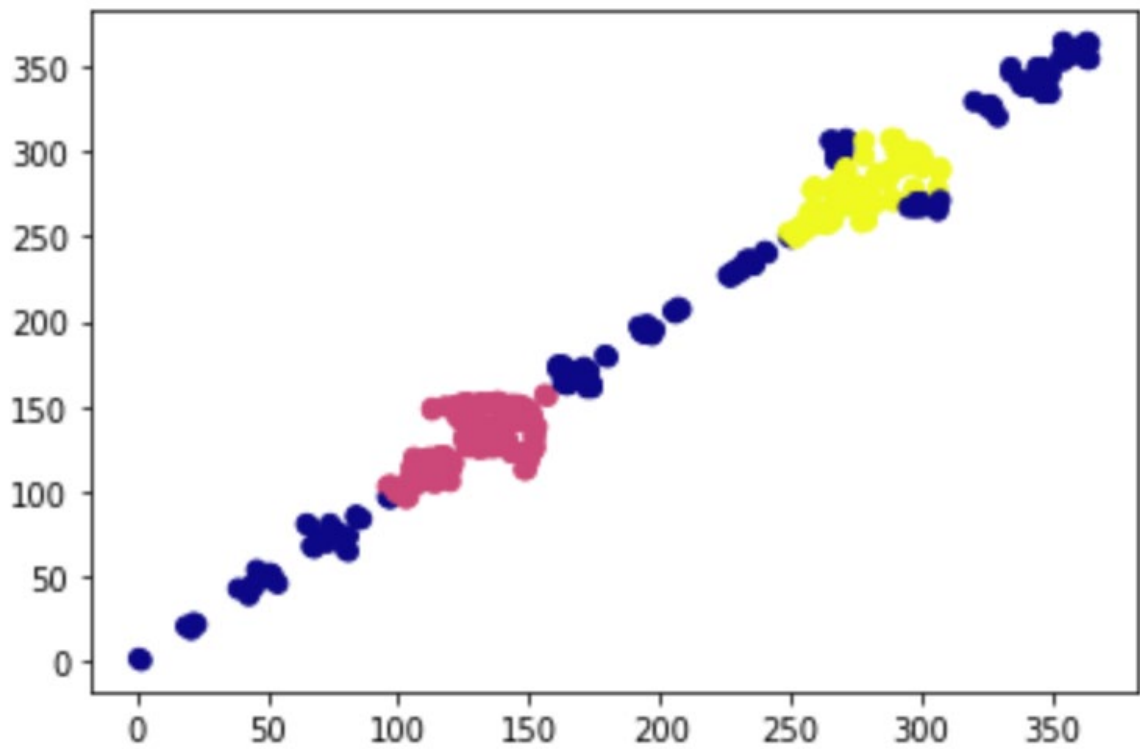


Рисунок 4.18 – Результат цієї кластеризації методом
DBSCAN для $H = 0,9$

Таблиця 4.1 – Середня оцінка правильності кластеризації

Значення параметру H	Відсоткова оцінка правильності кластеризації
0,4	71%
0,6	71%
0,7	72%
0,8	81%
0,7	86%

З усіх вищенаведених результатів можна зробити висновок, що при кластеризації всієї траєкторії руху частинки може використовуватися метод DBSCAN, але лише з емпірично підібраними параметрами.

Тому стає вкрай корисним та доцільним використання кластеризації побудованої за траєкторією рекурентної матриці, оскільки обидва методи (K-means і DBSCAN) показали дуже гарні результати, а аналіз їх разом може застосовуватися для задач у реальному житті.

ВИСНОВКИ

У кваліфікаційній роботі був проведений аналіз існуючих методів пошуку пасток у траєкторіях руху частинок. Були вивчені та розглянуті види броунівського руху та методи його моделювання. Була описана та практично застосована рекурентна матриця, що показує міри зв'язку між кроками траєкторії. Також був проведений огляд методів машинного навчання, а саме кластеризації. При розробці програмної реалізації були отримані навички використання математичних моделей, генерації даних та пошуку пасток.

Для вирішення поставленої мети були виконані наступні задачі:

- генерація броунівської траєкторії руху частинки (фрактальний та класичний);
- обчислення рекурентної матриці;
- реалізація обраних методів машинного навчання;
- перевірка роботи методів кластеризації на двох наборах даних (траєкторія та матриця);
- аналіз результатів.

Таким чином, було розроблено програмне забезпечення, яке шукає та знаходить пастки у траєкторії руху частинки шляхом кластеризації.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Gross W., Lohse M. J. Mechanism of activation of A2 adenosine receptors // Pharmacol. 1991. №39. P. 524–530.
2. Single-molecule imaging reveals receptor-G protein interactions at cell surface hot spots / T. Sungkaworn, M. L. Jobin, K. Burnecki, A. Weron // Nature. 2017. № 550. P. 543–547.
3. Relationship of lipid rafts to transient confinement zones detected by single particle tracking / C. Dietrich, B. Yang, T. Fujiwara, A. Kusumi // Biophys. J. 2002. № 82. P. 274–284.
4. Manzo C., Garcia-Parajo M. F. A review of progress in single particle tracking: from methods to biophysical insights // Reports on Progress in Physics. 2015 № 78. 124 p.
5. Holcman D., Hoze N., Schuss Z. Analysis and interpretation of superresolution single-particle trajectories // Biophysical Journal. 2015. №109. 1761 p.
6. Kinesin and dynein move a peroxisome in vivo: a tug-of-war or coordinated movement / C. Kural, H. Kim, S. Syed, G. Goshima // Science. 2005. №308. P. 149-152.
7. Myosin V walks hand-over-hand: single fluorophore imaging with 1.5-nm localization / A. Yildiz, J. N. Forkey, S. A. McKinney, T. Ha // Science. 2003. №300, 261 p.
8. Machine learning classification of particle trajectories in living cells: machine learning versus statistical testing hypothesis / I. Izeddin, V. R'ecamier, L. Bosanac, I. I. Ciss'e // eLife. 2014. №3, 182 p.
9. Michalet X. Mean square displacement analysis of single-particle trajectories with localization error: Brownian motion in an isotropic medium // Physical Review. 2010. №82. 41 p.
10. Estimating the anomalous diffusion exponent for single particle tracking data with measurement errors / K. Burnecki, E. Kepten, Y. Garini, G. Sikora // Scientific Reports. 2015. №5. P. 234-253.

11. Slaton P. J., Burroughs N. J. A Hidden Markov Model for Detecting Confinement in Single-Particle Tracking Trajectories // *Biophysical Journal*. 2018. №115. 17 p.
12. Bayesian Approach to MSD-Based Analysis of Particle Motion in Live Cells / N. Monnier, M. Mori, J. He, P. L'enfant // *Biophysical Journal*. 2012. №103. 616 p.
13. Automatic detection of diffusion modes within biological membranes using back-propagation neural network / P. Dosset, P. Rassam, L. Fernandez, C. Espenel // *BMC Bioinformatics*. 2016. №17. 197 p.
14. Comparative analysis of pseudorandom number generation in the up-to-date wireless data communication / L. Kirichenko, R. Tsekhmistro, O. Krug, A. Storozhenko // *Telecommunications and Radio Engineering*. 2011. № 70(4). P. 325-333.
15. Calebiro. Detecting transient trapping from a single trajectory: A structural approach, *Entropy* /Lanoiselée Y., Grimes J., Z. Koszegi, D. // Basel. 2021. 98 p.
16. Lions P.L., Mercier B. Splitting algorithms for the sum of two nonlinear operators // *SIAM J. on Numer. Anal.* 1979. P. 964-979.
17. Rudin L.I., Osher S., Fatemi E. Nonlinear total variation based noise removal algorithms // *Physica D*, vol. 60. 1992. P. 259-268.
18. Clason Ch, Jin B., Kunisch K. A duality-based splitting method for L1 -TV image restoration with automatic regularization parameter choice // *Journal of Computational and Applied Mathematics*. 2010. №11. 46 p.
19. Cai X., Chan R., Schönlieb C.B. Linkage between piecewise constant Mumford-Shah model and Rudin-Osher-Fatemi model and its virtue in image segmentation // *SIAM J. Sci. Comput.* 2019. №28. 31 p.
20. Mumford D., Shah J. Optimal approximations by piecewise smooth functions and associated variational problems // *Harvard library*, 1989. 110 p.
21. Кіріченко Л.О., Мельникова П.О. Кластеризація фрактальних реалізацій на основі машинного навчання // Тези XVI Міжнародної науково-практичної конференції «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті» (м. Дніпро, 14-15 грудня 2022 р.). Дніпро : ДІІТ, 2022. С. 76.