УДК 004.6:004.275 **DOI:** 10.31673/2412-9070.2025.013274
А. О. ГАШКО, аспірант;
ORCID: 0000-0001-4695-4689
А. А. СТРАЖНІКОВ, аспірант,
ORCID: 0009-0000-3492-2968
Державний університет інформаційно-комунікаційний технологій, Київ

ПОРІВНЯННЯ АЛГОРИТМІВ ПОБУДОВИ КЛАСТЕРНОЇ МОДЕЛІ НА БАЗІ НАБОРУ ДАНИХ (DATASET), ОТРИМАНОГО З BIGDATA

MeanShift — це популярний алгоритм кластеризації, який використовується в широкому діапазоні програм машинного навчання. Його суттєвим недоліком є повільна швидкість алгоритму, пов`язана з необхідністю витрачати параметр квадратичної складності (квадратичний час) на виконання однієї ітерації. Доповнивши алгоритм MeanShift за допомогою методу злиття режимів на основі кластеризації середнього зсуву, обґрунтовуючи даний підхід тим, що він дозволяє інтерпретувати ймовірнісну кластеризацію на основі спорідненості щільності ядер ваги. Також цей вид підключення дозволив принципово оптимізувати ядра ваги і також дозволив використовувати ядра ваги нефіксованого розміру відповідно до локальних структур даних. На цій основі роботу вдається пришвидшити в рази. На відміну від класичного MeanShift, комбінований підхід базується на лінійному часі виконання за кількістю точок та експоненціальний за розміром.

Метою цієї статті є висвітлення читачам для огляду процесу, а саме: як кластеризація середнього зсуву може бути застосована для побудови моделі, а також висвітлення переваг використання не класичного підходу до методики середнього зсуву порівняно з традиційними методами.

Ми намагатимемось створити узагальнений список криптотранзакцій, щоб надати користувачеві аналітику щодо ризиковості криптогаманця або окремої крипто-транзакції. Також проведемо порівняння впливу різних параметрів і функцій на вміст кластерів. Запропонований спосіб знижує витрати на обчислення, зберігаючи прийнятний рівень отриманих результатів кластеризації, як і стандартна процедура середнього зміщення. Продемонструємо ефективність методу на послідовності векторів, що не є сталими та змінюються в часі. Даний експеримент показує, що отримане значення середнього зсуву за допомогою нашої методики розрахунку відстані, перевершує отримані значення середнього зсуву за допомогою класичних методів роботи з неочевидними та неструктурованими значеннями. Для уточнення зав'язків між кластерами та підвищення точності сортування були використані такі параметри: ринкова капіталізація та деякі інші фіатні показники, які можна використовувати у майбутніх дослідженнях.

Ключові слова: кластеризація, машинне навчання, BigData, blockchain, крипто переказ, Mean Shift Clustering, інформаційні системи.

Результати дослідження

Було використано середнє зміщення (Mean-Shift Clustering). Цей алгоритм створений на базі обмеженої області дослідження із загального набору даних, основним завданням якого є знаходження густих областей скупчення точок (вузлів) даних.

Суть алгоритму полягає в тому, щоб знайти центральні точки кожної групи, які працюють шляхом оновлення кандидатів до центральних точок (вузлів) групи, так щоб вони були середніми точками у виділеній області даних.

Для відчуття ефективної роботи алгоритму (Mean-Shift Clustering) дуже важливим є етап після-обробки отриманих даних. На цьому етапі треба приділити увагу на виключення дублів та формування кінцевого набору центральних точок (вузлів) та їх відношення до відповідних

груп. Для того, щоб пояснити середнє зміщення, розглянуто набір точок (вузлів) у двомірному вимірі (рис.1) [1].

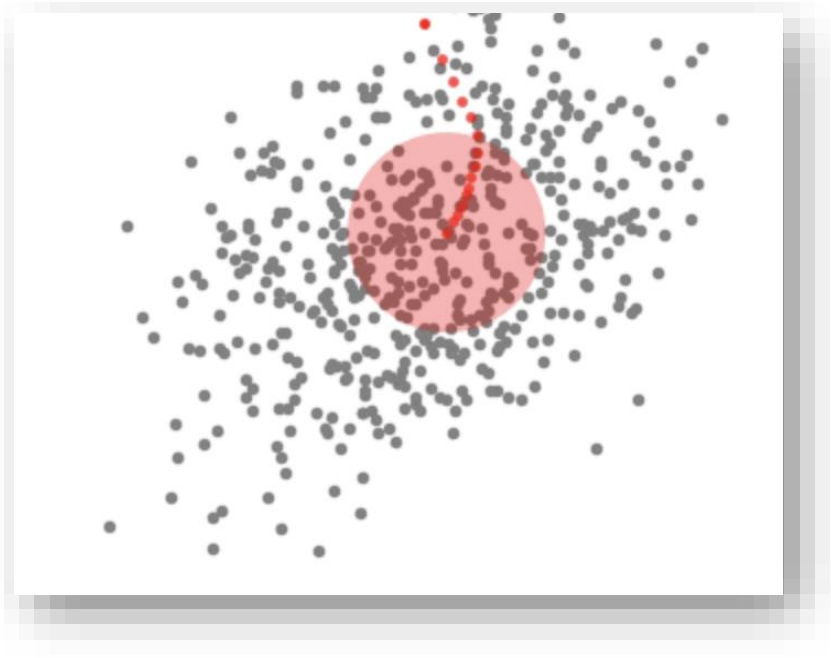

Рис. 1. Середнє зміщення у двомірному вимірі глибоким навчанням

Червоне коло і є виділеною областю даних з центром у точці А (вибраної випадковим чином) та радіусом **r** в якості області даних (ядра). Для того, щоб дослідити весь об'єм отриманих даних, треба переміщувати область вибраних даних **r** та змінювати середні точки даних. Для цього був використаний метод середнього зміщення, який за своєю суттю є алгоритмом Сходження та який включає в себе ітеративне зміщення виділеної області даних (ядра) в область з більшою щільністю точок (вузлів) даних на кожному кроці сходження роботи алгоритму. На кожній наступній ітерації область виділених даних зміщується в сторону області з більш високою щільністю шляхом зміщення центральної точки (вузлу) до середнього значення точки (вузлу) у виділеній області даних. Щільність у виділеній області даних пропорційна кількості точок (вузлів) в середині неї.

При роботі з алгоритмом Mean-Shift Clustering слід враховувати, що область виділених даних поступово зміщується в сторону області з більшою щільність точок (вузлів) даних. Переміщення області виділених даних буде продовжуватись до тих пір, доки не залишиться напряму, в якому зміщення області може розмістити в собі найбільшу кількість точок (вузлів) даних в середині ядра.

Рис.1 ілюструє, що область виділених даних продовжує переміщення, доки не перестане збільшуватись щільність точок (вузлів). Цей процес виконується з багатьма областями до тих пір, доки всі точки (вузли) не попадуть до одної області. Коли декілька виділених областей даних перекриваються, то область яка має найбільшу кількість точок (вузлів), зберігається. Після збереження точки (вузли) даних групуються в залежності з плаваючими областями виділених даних, в яких вони знаходяться.

Для визначення ядра ваги доцільно використати функцію Гауса (1) на прикладі отриманого ядра ваги за допомогою формули:

$$K(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}}, \qquad (1)$$

де σ — це параметр стандартного відхилення.

Після визначення ядра ваги ми можемо провести розрахунок оцінки щільності даних у кожній точці за формулою (2):

$$f(x) = \frac{x^2}{nh^{\mathrm{d}}} \sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right), \qquad (2)$$

де n – кількість точок даних;
h — параметр вікна пропускної здатності;
d - розмір простору;
x_i – точки (вузли) даних.

Кожна точка (вузол) даних потребує обчислення вектору, який вказує на області з максимальною щільністю точок (вузлів) даних. Для обчислення щільності доцільно використати ітераційний алгоритм градієнтного спуску (3).

$$m(x) = \frac{\sum_{i=1}^{n} K'\left(\frac{x-x_i}{h}\right)(x-x_i)}{\sum_{i=1}^{n} K'\left(\frac{x-x_i}{h}\right)}, \qquad (3)$$

де K' — похідна ядра по осі x.

На рис. 2 продемонстровано процес групування даних за допомогою довільного графіка. Вбачається, що сірі точки представляють собою вузли даних (точки), які гуртуються навколо кожної чорної точки і які являються центрами ваги (ядра).

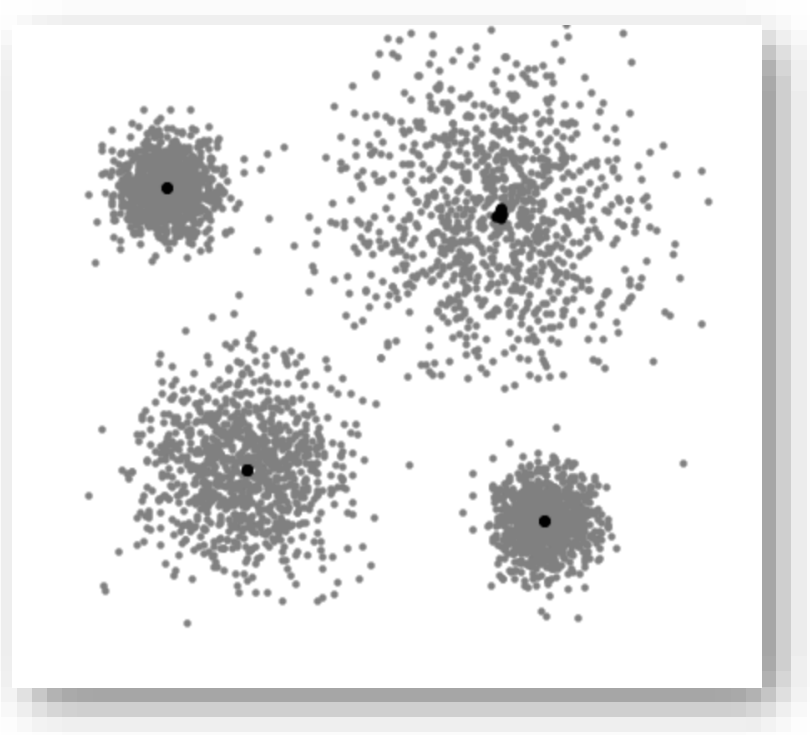

Рис. 2. Групування точок (вузлів) даних навколо ядер ваги

Прикладним завданням дослідження було відсортувати дані (транзакції) по трьом основним групам: мінімальний ризик, середній ризик та високий ризик. Створено та доповнено модель четвертою групою для того, щоб сортувати нульові транзакцій та інший пил для подальшого зменшення ваги та впливу на кінцевий результат. Результат дослідження проілюстровано на рис. 3.

Для оновлення точок (вузлів) даних, отриманих з Big Data, було переміщено кожну точку з набору даних, що досліджується, у напрямку попередньо розрахованого вектору середнього зміщення (4):

$$x_{t+1} = x_t + m(x_t) \qquad (4)$$

Однозначною перевагою алгоритму Mean-Shift Clustering є те, що не потрібно вказувати кількість кластерів, так як алгоритм середнього зміщення виявляє цей параметр автоматично. Також хочу виділити інтуїтивне розуміння, так як центри кластерів (ядра) сходяться з точками (вузлами) максимальної щільності.

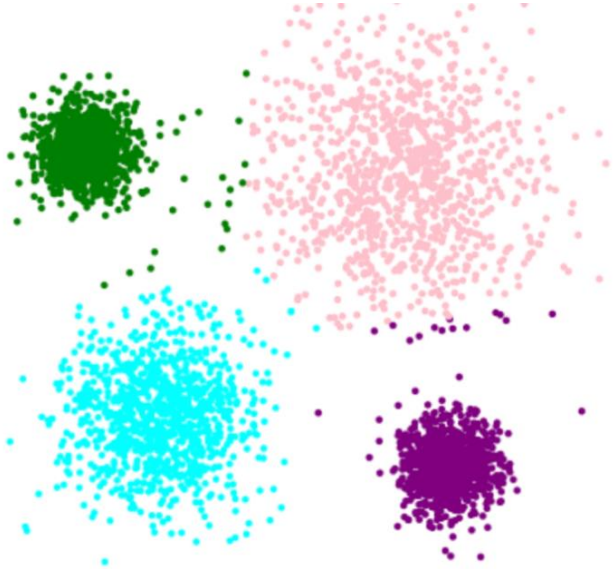

Рис. 3. Сортування транзакцій за рівнями ризиковості

Розрахунок параметру вікна пропускної здатності **h** є одним з найбільш важливих та не простих етапів, але дуже необхідним для отримання якісного результату. Невірний розрахунок параметрів вікна може привести до злиття режимів та створенню додаткових «площин» режимів. У бага-

тьох випадках доцільним є використання адаптивного розміру вікна, але конкретно для завдання цей метод не підходив, так як дистанція між точками (вузлами) даних була досить великою, щільність розміщення точок не пропорційною, а велика кількість шуму впливала на загальну похибку.

Тому, використання цього алгоритму потребує дуже ретельного та глибокого аналізу та розуміння тих даних, до яких алгоритм **Mean-Shift Clustering** буде застосовуватись.

Для вирішення конкретної задачі, середнє значення для чотирьох зафіксованих груп вираховується шляхом визначення середньоарифметичного значення, проводячи розрахунки на основі значень ваги для всіх точок (5):

$$M_A = \frac{1}{n}\sum_{n=1}^{n} x_i \ , \qquad (5)$$

де M – середнє значення;
n – розмір вибраних даних;
x_i – функція точок (вузлів) даних.

У даному випадку було недоцільно надавати рівні значення ваги всім точкам, оскільки частину датасету складали нульові транзакції та велика кількість шуму. Тому, одним із рішень є скористатись формулою (6):

$$M_w = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i} \ , \qquad (6)$$

де w_i – це вага для x_i, яка пов'язана з еквівалентною дистанцією між точками (вузлами) даних.

Інший алгоритм, який аналізується у цьому дослідженні - це алгоритм DBSCAN (просторова кластеризація додатків на основі щільності).

DBSCAN – це кластерний алгоритм, який за своєю суттю дуже схожий на алгоритм Mean-Shift Clustering, тому що обидва алгоритми створені на базі щільності, але все ж таки з різницею, яка і стане предметом дослідження.

Отримані переваги після використання **DBSCAN** алгоритму.

По-перше, алгоритм **DBSCAN** на відміну від **Mean-Shift Clustering** виявляє кластери будь-якої форми. Для роботи алгоритму **DBSCAN** ключовими є 2 параметри: поріг відстані та мінімальна кількість точок (вузлів даних). Для аналізу порогу відстані проведемо розрахунок відповідно трьох довільних точок даних. Даний метод передбачає обчислення середньої відстані між усіма точками та розташованими поруч сусідами. Індекс найближчих точок виразимо як відношення попередньої відстані, поділеної на прогнозовану відстань. Отриманий параметр є еквівалентною відстанню, представленою у рівнянні (7).

$$d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{i2} - x_{j2})^2} \qquad (7)$$

Отриманий індекс виразимо як R, і вказуємо розрахунок в рівняння (8):

$$R = \frac{d_{obs}}{d_{ran}} \ , \qquad (8)$$

де d_{obs} – середня попередня відстань між найближчими сусідніми точками;
d_{ran} – прогнозована середня відстань для заданих точок у випадковому порядку.

Розрахунок d_{obs} та d_{ran} представимо формулою (9):

$$d_{obs} = \frac{\sum_{n=1}^{n} d_i}{n} \text{ та } d_{ran} = \frac{1}{2\sqrt{\mathrm{p}}} = \frac{1}{2\sqrt{\mathrm{n/A}}} \ , \qquad (9)$$

де d_i – відстань між i та його найближчими сусідами;
n – загальна кількість точок;
A – мінімально прогнозована площа навколо всіх точок (вузлів) даних.

Коли можливість додавання нових точок (вузлів) до кластеру вичерпана, то алгоритм DBSCAN переходить до наступних, не опрацьованих точок (вузлів) та повторює процедуру поки не будуть опрацьовані всі точки (вузли) (рис. 4).

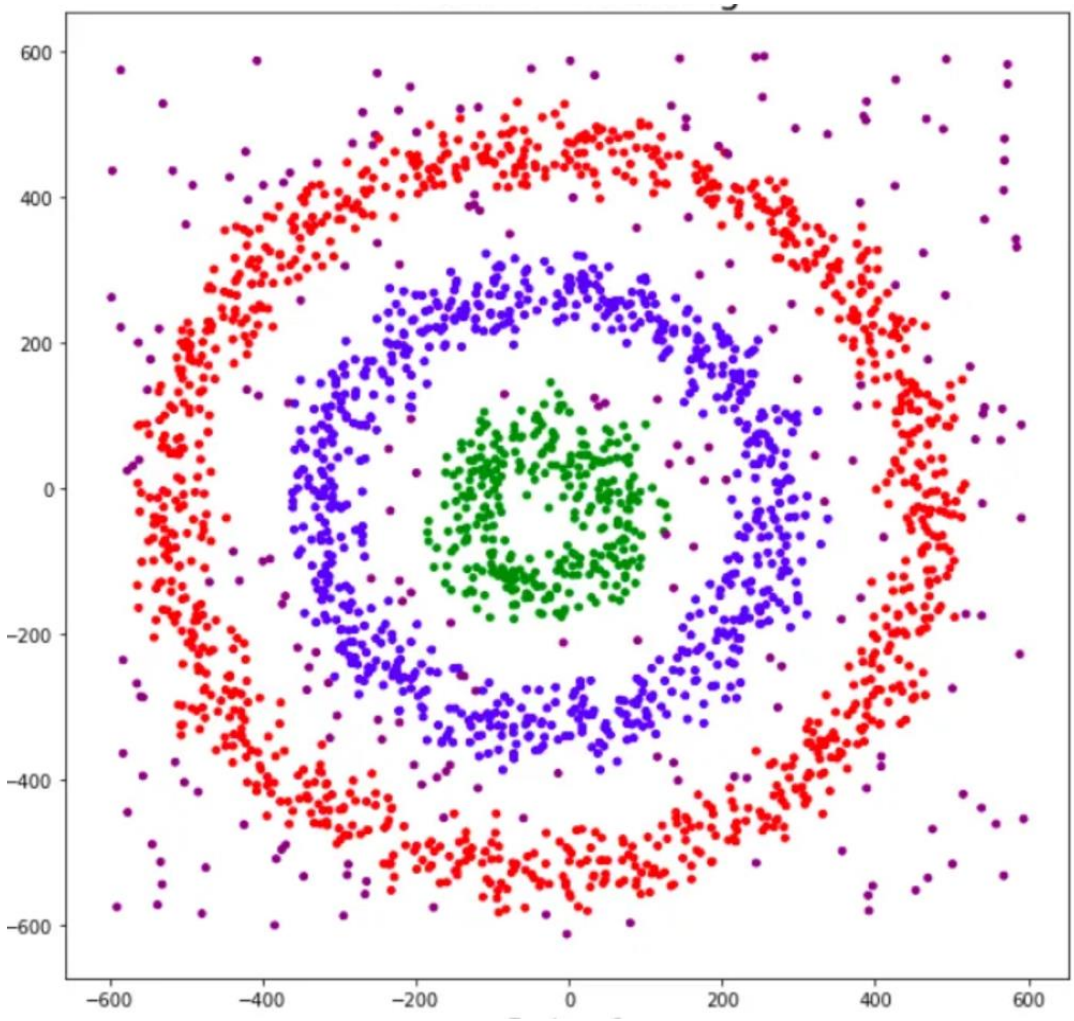

Рис.4. Відстань між точками та найближчими сусідами

Обговорення результатів проведеного дослідження

Об'єднання можливості обох алгоритмів та врахування коефіцієнту похибки, для досягнення результату з групування транзакцій за своєю схожістю навколо заданих ядер ваги, надалі надає нам можливість використовувати отримані точки для встановлення (криптографічних міток розпізнавання) та подальшого аналізу фінального відсотку ризику крипто-переказу або конкретного гаманця.

Висновки

Застосувавши дану методику, досягнуто кратного пришвидшення обробки великого масиву даних за рахунок виділення та групування інформаційних вузлів, що мають для нас цінність. Як наслідок, відчутне збільшення швидкості обчислення загального відсотку ризику. З недоліків даної методики зазначимо, що модель бази не є класичною реляційною моделлю, тому потребує постійного оновлення та аналізу. Серед переваг - це швидкість та глибина пошуку.

Список літератури

1. W. Chen, H. Zhang, and L. Jia, "A novel two-stage method for well-diversified portfolio construction based on stock return prediction using machine learning," The North American Journal of Economics and Finance, vol. 63, p. 101818, Nov. 2022, doi: 10.1016/j.najef.2022.101818.

2. Gupter K.C. Analysis and Design of Planar Microwave Components. – Institute of Electrical and Electronic Engineering, 2021. – 586 p.

3. Gerald C. Alexander, Andreas Weisshaas, Vijai K. Tripathi, Philip C. Magnusson , Philip Cooper . Transmission Lines and Wave Propagation. CRC Press, 2020. – 536 p.

4. Fawwaz T. Ulaby. Fundamentals of Applied Electromagnetics. – Prentice Hall, 2003. – 464 p.

5. Pourush R., Jangid A., Tyagi G.S. et al. Magnetically tunable microstrip linearresonator on polycrystalline ferrite // Microwave and Optical Technology Letters. – 2017. – Vol. 49. – № 11. – P. 2868–2870.

A. Hashko, A. Strazhnikov

COMPARISON OF ALGORITHMS FOR BUILDING A CLUSTER MODEL BASED ON A DATASET OBTAINED FROM BIGDATA

MeanShift is a popular clustering algorithm widely used in a range of machine learning applications. A major drawback is the slow speed of the algorithm, as it requires quadratic time for one iteration. By enhancing the MeanShift algorithm with a mode-merging method based on mean-shift clustering, we justify this approach by showing that it allows probabilistic clustering interpretation based on the affinity of kernel density weights. This type of integration also optimizes the weight kernels and enables the use of variable-sized kernels according to local data structures. As a result, we achieved a significant speed improvement. Unlike classical MeanShift, this combined approach is based on linear time with respect to the number of points and exponential with respect to size. The aim of this article is to provide an overview of how mean-shift clustering can be applied to model building and to highlight the advantages of using a non-classical approach to mean-shift methodology compared to traditional methods. We will attempt to create a generalized list of crypto transactions to provide users with risk analytics for a crypto wallet or an individual crypto transaction. We will also compare the influence of different parameters and functions on cluster composition. The proposed method reduces computational costs while maintaining an acceptable level of clustering accuracy, similar to the standard mean-shift procedure. We will demonstrate the method's effectiveness on a sequence of vectors that are non-constant and change over time. This experiment shows that the mean-shift values obtained through our distance calculation method outperform those obtained using classical methods when dealing with non-obvious and unstructured data values. To clarify the relationnnships between clusters and improve sorting accuracy, parameters such as market capitalization and other fiat indicators were used, which can be applied in future studies.

Keywords: clustering, machine learning, Big Data, blockchain, crypto transfer, Mean Shift Clustering.