

УДК 004.94DOI
DOI [https://doi.org/10.15589/znп2026.1\(503\).2.17](https://doi.org/10.15589/znп2026.1(503).2.17)

EVALUATION OF CLUSTERING ALGORITHMS UNDER NOISY AZIMUTH MEASUREMENTS

ОЦІНЮВАННЯ АЛГОРИТМІВ КЛАСТЕРИЗАЦІЇ В УМОВАХ ШУМНИХ АЗИМУТАЛЬНИХ ВИМІРЮВАНЬ

Tetyana A. Farionova
tetyana.farionova@nuos.edu.ua
ORCID: 0000-0003-3384-4712
Artem A. Klieptsov
artem.klieptsov@nuos.edu.ua
ORCID: 0009-0008-3147-0618

Т. А. Фаріонова,
канд. техн. наук, доцент
А. А. Клепцов,
аспірант

Admiral Makarov National University of Shipbuilding, Mykolaiv
Національний університет кораблебудування імені адмірала Макарова, м. Миколаїв

Abstract. Object localization based on azimuth-only (bearing-only) measurements is a challenging task due to high data noise, angular errors, and a large number of spurious ray intersections. The reliability of coordinate estimation largely depends on clustering methods capable of separating true intersection points from noise. This work examines the application of the K-means and DBSCAN algorithms to the clustering of azimuthal intersection points and evaluates their suitability for bearing-only observation tasks.

The purpose of the study is to provide a comprehensive assessment of the effectiveness of the K-means and DBSCAN clustering algorithms in object localization based on noisy azimuth measurements, including an analysis of their ability to distinguish true intersections from spurious ones and to produce accurate position estimates in bearing-only scenarios.

Method. The research was conducted by modeling bearing-only scenes in which several radars generate azimuthal rays, and the intersection points of these rays serve as input data. The K-means and DBSCAN algorithms were applied to these datasets. Their effectiveness was evaluated using three metrics: the number of clusters detected, positional localization error, and angular consistency with measured azimuths. An integral metric, A_{total} , was constructed based on these indicators, and statistical differences between the algorithms were tested using the Wilcoxon signed-rank test.

Scientific novelty. The study proposes a specialized system of metrics for the comprehensive evaluation of clustering algorithms in bearing-only tasks, combining quantitative, positional, and angular components of quality. An integral indicator was developed that enables an objective comparison of algorithms and an assessment of their robustness to noise. The findings demonstrate that the algorithms behave differently depending on data structure and the density of spurious intersections.

The *conclusions* indicate that the results of the study allow for a justified selection of clustering methods for bearing-only localization under high-noise conditions. The proposed metric system and integral indicator can serve as a foundation for developing more robust methods of azimuth data processing and for optimizing multi-radar configurations. The experiments showed that DBSCAN consistently outperforms K-means according to the integral metric, particularly as the number of targets increases.

Key words: object localization; azimuth measurements; clustering; K-means; DBSCAN; quality metrics; clustering algorithms; unity; 2D game engine.

Анотація. Локалізація об'єктів за азимутальними (bearing-only) вимірюваннями є складною задачею через високу шумність даних, кутові похибки та значну кількість хибних перетинів променів. Надійність визначення координат суттєво залежить від методів та алгоритмів кластеризації, які здатні відокремлювати істинні точки від шумових. У роботі розглянуто порівняльний аналіз алгоритмів K-means та DBSCAN для кластеризації азимутальних перетинів з подальшим їх застосуванням в задач bearing-only спостереження.

Метою роботи є оцінювання ефективності алгоритмів кластеризації K-means та DBSCAN в умовах шумних азимутальних вимірювань у bearing-only задачах.

Методика. В дослідженні використано методи статистичного аналізу та імітаційного моделювання.

Наукова новизна. У роботі набуло подальший розвиток метод оцінки алгоритмів кластеризації в bearing-only задачах на основі спеціалізованої системи метрик, що дозволить об'єктивно порівнювати різні алгоритми кластеризації та оцінити їхню робастність до шуму.

У висновках зазначено, що результати роботи дозволяють обґрунтовано вибирати алгоритми кластеризації для задач bearing-only локалізації в умовах підвищеного шуму. Запропонована система метрик і інтегральний показник які можуть бути основою для подальшої розробки стійкіших методів оброблення азимутальних вимірювань і оптимізації мультирадарних конфігурацій. Показано, що DBSCAN стабільно перевершує K-means за інтегральною метрикою, особливо зі зростанням кількості цілей.

Ключові слова: локалізація об'єктів; азимутальні вимірювання; кластеризація; K-means; DBSCAN; метрики якості; алгоритми кластеризації; 2D графічні рушії.

ПОСТАНОВКА ЗАДАЧІ

У bearing-only системах спостереження положення об'єкта визначається на основі азимутальних вимірювань, що надходять від кількох радарів. У реальних умовах такі вимірювання містять значний шум, що призводить до появи великої кількості хибних точок перетину азимутальних променів. Основна задача полягає у тому, щоб на основі множини цих перетинів виділити істинні скупчення та отримати найбільш точні оцінки координат об'єктів.

Для цього необхідно дослідити, як алгоритми кластеризації K-means і DBSCAN справляються з групуванням шумових та істинних точок у різних конфігураціях сцени, визначити їх здатність відновлювати правильну кількість об'єктів та забезпечувати мінімальну помилку локалізації. Таким чином, задачею дослідження є оцінювання ефективності кластеризаційних методів у процесі локалізації об'єктів за шумними азимутальними вимірюваннями та визначення їх придатності до практичного використання в bearing-only системах.

АНАЛІЗ ОСТАННІХ ДОСЛІДЖЕНЬ І ПУБЛІКАЦІЙ

Проблематика локалізації об'єктів за азимутальними (bearing-only) вимірюваннями активно досліджується у контексті навігаційних, радіолокаційних та мульти-сенсорних систем. Одним із ключових викликів у цій сфері є велика кількість хибних перетинів ліній спрямування, що виникають через шум, обмежену кількість радарів та складні геометричні конфігурації. Це зумовлює необхідність використання методів кластеризації для відокремлення правдивих оцінок від спуріозних.

У низці робіт [1–4] показано, що алгоритми щільнісної кластеризації, зокрема DBSCAN та його модифікації, є ефективними для видалення хибних точок, та підвищення точності локалізації.

Питання асоціації даних та вибору коректних перетинів розглянуто у класичній праці Hero & Blatt [5], де автори демонструють, що попередня кластеризація є критичним кроком у задачах мультиджерельної локалізації за азимутами. Дослідження підкреслює, що геометрична нестійкість bearing-only задачі може бути суттєво компенсована застосуванням методів групування та статистичної перевірки достовірності оцінок.

Крім того, у роботі Daniel Carvalho de Ramos, Lucas Reksua Ferreira та ін. [6] з радарної аналітики порівнюються K-means, DBSCAN, Mean-Shift та інші алгоритми в задачах просторової кластеризації точок сенсорних спостережень. Показано, що K-means є ефективним у випадку, коли кількість об'єктів відома наперед, тоді як DBSCAN краще працює у шумних середовищах та при складних формах кластерів.

Українські дослідники також активно аналізують властивості алгоритмів кластеризації, зокрема акцентуючи увагу на їх чутливості до параметрів, стійкості до шуму та здатності до адаптації під різні типи просторових даних [7, 8].

Однак, слід зазначити, що попри значну кількість робіт, як вітчизняних так й закордонних авторів, специфічний аспект практичної оцінки K-means та DBSCAN у задачі локалізації об'єктів саме за перетинами азимутальних променів залишається недостатньо висвітленим. Тому оцінювання ефективності алгоритмів кластеризації K-means та DBSCAN в умовах шумних азимутальних вимірювань є актуальною науково-практичною задачею.

ВІДОКРЕМЛЕННЯ НЕВИРІШЕНИХ РАНІШЕ ЧАСТИН ЗАГАЛЬНОЇ ПРОБЛЕМИ

Попри активні дослідження у сфері локалізації за азимутальними вимірюваннями, низка ключових аспектів залишається недостатньо вивченою. Більшість робіт зосереджені на теоретичних моделях або абстрактних вибірках, без урахування реальної геометрії спуріозних перетинів, що виникають через шум, похибки кутових вимірювань та складні конфігурації радарів. Недостатньо опрацьованою є проблема комплексної оцінки якості кластеризації: у наявних підходах метрики зазвичай охоплюють лише просторову точність або кількість кластерів, без інтеграції позиційної, кутової та кількісної складових в єдину систему. Бракує практичного порівняння K-means і DBSCAN саме в bearing-only задачах, де кількість цілей може бути невідомою, а результат суттєво залежить від рівня шуму та конфігурації сцени.

Отже, відсутній комплексний підхід, що охоплює моделювання bearing-only сцен, кластеризацію перетинів, багатокритеріальну оцінку та статистичне порівняння алгоритмів – ці прогалини і визначають актуальність та новизну даної роботи.

МЕТА ДОСЛІДЖЕННЯ

Метою роботи є оцінювання ефективності алгоритмів кластеризації K-means та DBSCAN в умовах шумних азимутальних вимірювань у bearing-only задачах.

МЕТОДИ, ОБ'ЄКТ

ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Об'єкт дослідження – процес оцінювання локалізації об'єктів за азимутальними вимірюваннями в багаторадарних bearing-only системах у присутності шуму та спуріозних перетинів променів.

Предмет дослідження є моделі та алгоритми для оцінювання локалізації об'єктів за азимутальними вимірюваннями в багаторадарних bearing-only системах у присутності шуму та спуріозних перетинів променів.

Методи дослідження. В дослідженні використано методи статистичного аналізу та імітаційного моделювання.

ОСНОВНИЙ МАТЕРІАЛ

Для оцінювання алгоритмів кластеризації в умовах шумних азимутальних вимірювань було запропоновано модель багаторадарної bearing-only сцени. Ця модель реалізовано у середовищі Unity, що дає можливість створити керовану 2D симуляцію радарів та об'єктів із точною геометричною взаємодією. Використання C# дозволяє програмно задавати конфігурацію радарів, автоматично генерувати позиції цілей та формувати азимутальні промені від кожного радара до об'єктів. Перетини цих променів слугують вхідними даними для подальшої кластеризації. Unity також забезпечує зручні інструменти для візуалізації хмар точок та реалізації алгоритмів K-means і DBSCAN у режимі моделювання, що дозволяє досліджувати їхню роботу за різних рівнів шуму та просторових конфігурацій. Авторами запропоновано модель для кластеризації та локалізації об'єктів за азимутальними даними, яка має наступну конфігурацію що наведена нижче. (рис. 1).

Радари і їх конфігурація. У моделі використано п'ять радарів (рис. 2): один основний та чотири додаткові. Усі радари представлені у вигляді кіл радіусом 10 одиниць. Радар має скрипт CircleArea. Він відповідає за геометрію та візуалізацію 2D-кола, а також надає функцію отримання випадкової точки всередині нього.

Основний радар розташований у центрі сцени за координатами (0, 0). Чотири додаткові радари розміщено симетрично відносно центру: R₂: (1.4, -5, 0); R₃: (-1.4, -5, 0); R₄: (1.4, 5, 0); R₅: (-1.4, 5, 0).

Таке розташування створює сцену, де додаткові радари частково перекривають область основного, забезпечуючи різноманітні кути огляду та підвищуючи ймовірність отримання коректних перетинів азимутальних променів.

За генерацію цілей усередині заданої круглої області відповідає скрипт TargetSpawner, котрий

є тільки на основному колі. Вхідними параметрами для цього скрипта є кількість об'єктів, які треба створити, та їх префаб. Під час старту він очищає старі цілі та створює початкову кількість нових об'єктів, а також дозволяє спавнити їх по одному або пачками. Візуально точки виглядають як малі зелені кола.

У моделі використовується один центральний генератор цілей, який розміщує об'єкти лише всередині головного кола. Решта радарів (усього п'ять) не мають власних локальних цілей, а натомість обчислюють азимут до тих самих глобальних об'єктів, використовуючи власні координати як точки спостереження. Це робиться за допомогою окремого скрипта RadarAzimuthProvider. Таким чином, кожен радар формує унікальний набір азимутальних напрямків на однаковий набір цілей, що відповідає класичній геометричній постановці задачі bearing-only локалізації, у якій різні спостерігачі реєструють тільки напрямки на об'єкт, а не відстань до нього.

Обчислення азимуту здійснюється за допомогою нормалізованого вектора $\vec{d}$:

$$\vec{d} = \frac{(targetPosition - center)}{|targetPosition - center|}$$

де targetPosition – вектор положення цілі (координати об'єкта, створеного TargetSpawner-ом у центральному колі); center – вектор положення радара-спостерігача, відносно якого формується азимут; $(targetPosition - center)$ – вектор з центра радара до цілі, що задає необроблений напрямок; $|targetPosition - center|$ – евклідова норма (довжина вектора), яка використовується як дільник під час нормалізації.

Такий підхід дозволяє перетворити вектор будь-якої довжини на одиничний вектор, зберігаючи тільки його напрямок. Отриманий одиничний вектор описує азимутальний промінь і є основою для створення структури AzimuthRay(center, dir), де фіксуються як точка початку (центр радара), так і напрямок променя до цілі.

За потреби азимутальні промені додатково візуалізуються у середовищі Unity засобами Debug.DrawLine, що дає змогу контролювати коректність геометричної моделі. У підсумку кожен радар утворює набір променів, спрямованих до глобальних цілей, а їхня сукупність формує множину геометричних перетинів. Саме ці перетини, справжні та спуріозні, і становлять вхідні дані для подальшої кластеризації, необхідної для відновлення реальних положень об'єктів.

Система обробки азимутальних даних. У bearing-only локалізації азимутальні промені визначають лише напрямок на об'єкт без дистанції, тому перетини променів від різних радарів слугують єдиним джерелом оцінки місцезнаходження цілі. У системі за це відповідає компонент AzimuthIntersectionSystem, який реалізує логіку тріангуляції та візуалізації

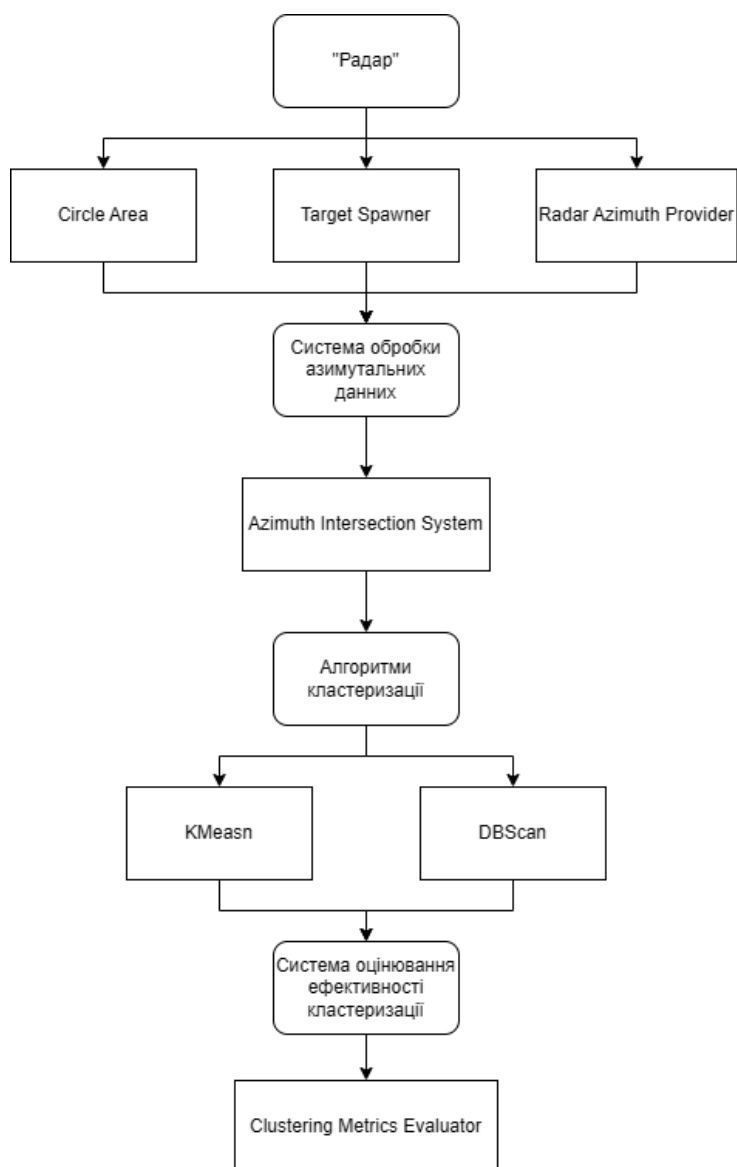


Рис. 1. Модель для кластеризації та локалізації об'єктів за азимутальними даними

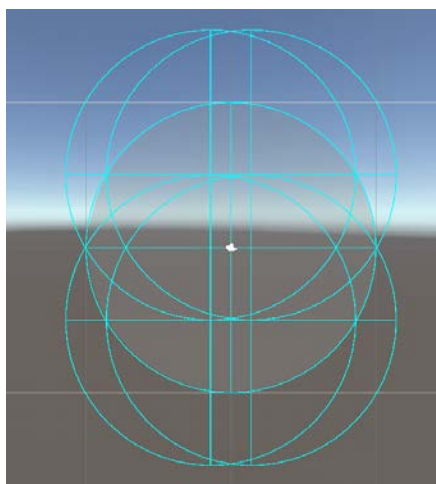


Рис. 2. Візуалізація перетину радарів

кандидатних точок. Візуально перетини виглядають як малі червоні кола (рис. 3).

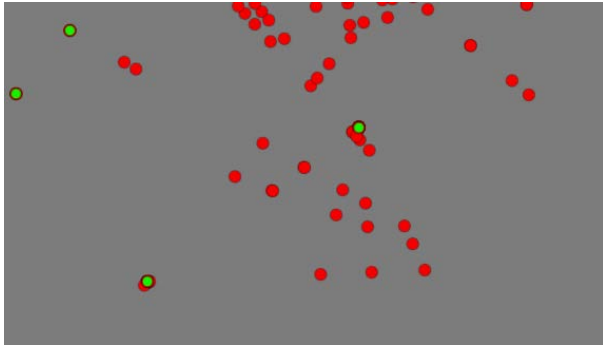


Рис. 3. Візуалізація цілей

Проте в реальних умовах таких перетинів виникає значно більше, ніж є об'єктів, що призводить до появи спуріозних (хибних) точок. Основною причиною надмірної кількості точок є те, що кожен радар створює окремі промінь на кожен об'єкт у радіусі своєї дії. При наявності декількох цілей та кількох радарів кількість пар променів швидко зростає, і більшість їхніх перетинів не мають відношення до реальних об'єктів. Це є джерелом природного шуму та створює складну хмару даних для подальшого тестування алгоритмів кластеризації K-means та DBSCAN.

Робота скрипту AzimuthIntersectionSystem починається з автоматичного пошуку всіх активних компонентів RadarAzimuthProvider у сцені. Основна логіка зосереджена в методі RebuildIntersections, який спочатку очищує попередні результати, видаляючи старі маркери з пам'яті та внутрішнього списку. Наступний кроком алгоритм визначає усі перетини променів і на їх місці створює потенційні точки, які візуально репрезентовані як червоні кола (рис. 3).

Алгоритми кластеризації. У роботі застосовано два алгоритми кластеризації – K-means та DBSCAN, які працюють над множиною точок перетину азимутальних променів. Обидва алгоритми реалізовані в середовищі Unity та використовуються для виявлення груп точок, що відповідають реальним об'єктам.

Перший алгоритм базується на алгоритмі K-means, який використовується у випадках, коли кількість цілей вважається відомою заздалегідь. У поточному моделюванні параметр кількості кластерів встановлено як $K = 10$ (з можливістю динамічної зміни в ході експерименту), а максимальна кількість ітерацій оновлення центроїдів обмежена значенням Max Iterations = 50. Алгоритм ітеративно мінімізує сумарне квадратичне відхилення цілей від центрів кластерів, де кожен фінальний центроїд виступає математичною оцінкою координат потенційного об'єкта. У середовищі Unity результати роботи K-means візуалізуються у вигляді салатових гексагональних маркерів, що дозволяє наочно оцінити

точність суміщення обчислених центрів із реальними цілями (рис. 4).

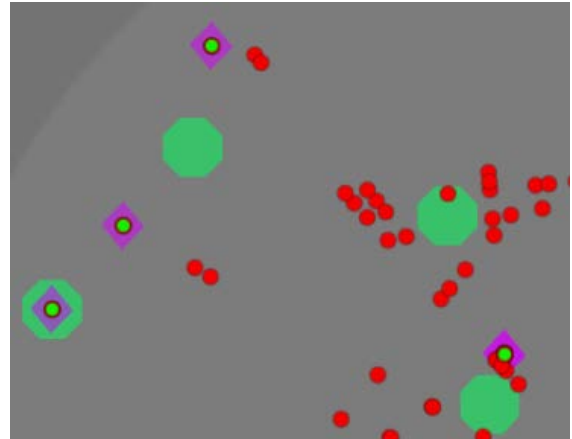


Рис. 4. Візуалізація кластеризації за алгоритмами K-Means та DBSCAN

Як альтернативний підхід, що не потребує апіорної інформації про кількість цілей, застосовано алгоритм DBSCAN. Цей алгоритм ідентифікує кластери на основі щільності розподілу точок, що є критично важливим для відсікання поодиноких спуріозних перетинів. Для дослідження обрано параметри радіуса околу $\text{eps} = 0.01$ та мінімальної кількості точок для формування ядра кластера $\text{min Points} = 2$. Таке налаштування дозволяє алгоритму ефективно відокремлювати щільні групи перетинів, які з високою ймовірністю відповідають реальним об'єктам, від ізолюваного геометричного шуму. Центри виявлених кластерів відображаються у сцені як фіолетові ромби, що забезпечує чітку візуальну диференціацію між двома методами локалізації на фоні загальної хмари червоних маркерів (рис. 4).

Система оцінювання ефективності кластеризації. Оцінювання проводилося у два етапи:

- розрахунок трьох доменно-специфічних метрик якості;

- статистичне порівняння алгоритмів за допомогою непараметричного критерію Вілкоксона.

Метрики якості:

- localization Error (A_{loc}) визначає метричну точність системи. Розраховується як медіана евклідових відстаней між центроїдами сформованих кластерів та істинними координатами (ground truth) об'єктів.

- azimuth Consistency Score (A_{angle}) специфічна метрика для систем радіолокації, що базуються на методі пеленгації (bearing-only localization). Вона оцінює кутову неузгодженість між розрахованою позицією кластера та первинними азимутальними замірами радарів.

- accuracy of Detected Number of Targets (A_{count}) характеризує здатність алгоритму (наприклад, DBSCAN) до коректної ідентифікації кількості

об'єктів у зоні спостереження. Враховуючи схильність алгоритмів до генерації помилкових кластерів (noise) або об'єднання близько розташованих цілей.

Формування інтегральної метрики A_{total} . Для отримання єдиної оцінки ефективності алгоритму використано метод адитивної згортки. Оскільки кожна метрика має різний ступінь впливу на цільову функцію системи, було введено вектор значень вагових коефіцієнтів параметрів A_{loc} (вага w_1), A_{angle} (вага w_2), A_{count} (вага w_3):

$$W = \{w_1, w_2, w_3\}.$$

Для обґрунтування величин вагових коефіцієнтів, авторами використано Метод аналізу ієрархій (АНР) [9]. Цей підхід дозволяє мінімізувати суб'єктивізм шляхом попарного порівняння важливості метрик за шкалою Т. Сааті.

Пріоритет 1 – A_{loc} . В системах локалізації це «головна» метрика. Якщо система неправильно визначає координати, то всі інші параметри втрачають сенс.

Пріоритет 2 – A_{angle} . Ви зазначили, що працюєте з радаром (bearing-only). У таких системах кут є первинним даним. Помилка в куті безпосередньо призводить до помилки локалізації, тому цей параметр критично важливіший за просту кількість об'єктів.

Пріоритет 3 – A_{count} . Це важлива метрика для оцінки роботи DBSCAN, але вона дискретна і часто є допоміжною. Навіть якщо алгоритм помилився на 1 об'єкт, але інші 10 локалізував ідеально, система вважається ефективною.

Формалізуємо викладене в такому вигляді:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \Rightarrow \begin{cases} \sqrt[3]{a_{11} * a_{12} * a_{13}} = p_1 \\ \sqrt[3]{a_{21} * a_{22} * a_{23}} = p_2 \\ \sqrt[3]{a_{31} * a_{32} * a_{33}} = p_3 \end{cases} \Rightarrow$$

$$\Rightarrow (p_1 + p_2 + p_3) = P \Rightarrow$$

$$\Rightarrow \left\{ \frac{p_1}{P} = w_1, \frac{p_2}{P} = w_2, \frac{p_3}{P} = w_3 \right\} \Rightarrow$$

$$\Rightarrow (w_1, w_2, w_3) - \text{вектор пріоритетів.}$$

Тут A – зворотно-симетрична матриця попарних порівнянь елементів a ; p – власний вектор; P – оператор нормування; w_i – вагові коефіцієнти пріоритетів. Порівняння виконуються з використанням фундаментальної шкали [9] (табл. 1).

$$V_{sum} = 2.154 + 1.145 + 0.404 = 3.703$$

Далі виконується перевірка узгодженості елементів матриці A за допомогою підрахунку відношення узгодженості. λ_{max} визначається як середнє арифметичне відношення елементів вектора зважених сум до відповідних вагових коефіцієнтів:

$$\lambda_{max} = \frac{1}{n} \sum_{i=1}^n \frac{\sum_{j=1}^n a_{ij} * w_j}{w_i} = \frac{1}{3} \left(\frac{1.745}{0.582} + \frac{0.927}{0.309} + \frac{0.327}{0.109} \right) \approx 3.003$$

λ_{max} дуже близьке до ідеальних 3 (кількість параметрів).

$$\text{Індекс узгодженості (CI)} : (3.003 - 3) / 2 = 0.0015$$

$$\text{Відношення узгодженості (CR): } 0.0015 / 0.58 = 0.0026$$

$CR < 0.1$, тому можна зробити висновок, що матриця ідеально узгоджена.

Згідно з результатами розрахунку, інтегральна метрика якості A_{total} набуває вигляду:

$$A_{total} = 0.58A_{loc} + 0.31A_{angle} + 0.11A_{count}$$

Статистична перевірка відмінностей між алгоритмами. Для визначення, чи є різниця між алгоритмами статистично значущою, застосовано критерій Вілкоксона (Wilcoxon signed-rank test) [10].

Обґрунтування вибору критерію:

- метрики не гарантують нормального розподілу;
- вибірки парні (K-means та DBSCAN обробляють одні й ті ж сцени);

Таблиця 1. Фундаментальна шкала

Ступінь переваги	Визначення	Коментар
1	Рівна перевага	Дві альтернативи, однакові з позиції мети
2	Слабкий ступінь переваги	Проміжна градація між рівною і середньою перевагами
3	Середній ступінь переваги	Досвід експерта дозволяє вважати одну з альтернатив трохи кращою за іншу
4	Перевага вище середнього	Проміжна градація між середньою і помірно сильною перевагами
5	Помірно сильна перевага	Досвід експерта дозволяє вважати одну з альтернатив явно кращою за іншу
6	Сильна перевага	Проміжна градація між помірно сильною і дуже сильною перевагами
7	Дуже сильна (очевидна) перевага	Досвід експерта дозволяє вважати, що одна з альтернатив домінує над іншою
8	Дуже-дуже сильна перевага	Проміжна

Таблиця 2. Розрахунок вектору пріоритетів

	A_{loc}	A_{angle}	A_{count}	P	V	w
A_{loc}	1	2	5	10	2.154	0.582
A_{angle}	0.5	1	3	1.5	1.145	0.309
A_{count}	0.2	0.33	1	0.066	0.404	0.109

– обсяг даних у кожній серії тестів – 30 повторів, що ідеально підходить для непараметричного підходу.

Тому t-тест непридатний (вимагає нормальності), а критерій Вілкоксона – оптимальний.

Оцінювання алгоритмів кластеризації K-means і DBSCAN проводилось за такими етапами:

1. Генерація N сценаріїв ($N = 30$);
2. Обчислення метрик A_{count} ; A_{loc} ; A_{angle} ; A_{total} для кожного сценарію;
3. Формування пар вибірок (K-means та DBSCAN);
4. Порівняння за критерієм Вілкоксона кожної пари вибірок (K-means та DBSCAN);
5. На основі критерію Вілкоксона визначається p -value, яке використовується для оцінювання статистичної значущості різниці між вибірками:
 - $p < 0.05$ – різниця значуща;
 - $p \geq 0.05$ – різниця статистично не підтверджена.

Опис процесу моделювання та проведення порівняльного аналізу алгоритмів кластеризації

1. Конфігурації радарів (1+4). У моделі використано п'ять радарів: один основний та чотири додаткові. Усі радари представлені у вигляді кіл радіусом 10 одиниць. R_1 : (0, 0, 0), R_2 : (1.4, -5, 0), R_3 : (-1.4, -5, 0), R_4 : (1.4, 5, 0), R_5 : (-1.4, 5, 0).

2. Генерація наборів цілей. Початкові цілі для локалізації генеруються випадковим чином усередині основного кола. Позиції вибираються рівномірно по площі кола

3. Формування вхідних даних для формування кластерів (перетини азимутів). Після розміщення реальних цілей кожен радар генерує азимутальні промені до об'єктів у своїй зоні. Для кожної пари радарів обчислюються точки перетину відповідних променів. Усі знайдені перетини – як коректні, так і спуріозні – додаються до спільної множини, що утворює «хмару точок».

4. Формування кластерів можливих цілей. Для кластеризації хмари перетинів застосовано два алгоритми K-means і DBSCAN. K-means працює з фіксованою кількістю кластерів ($K = 10$ (буде змінюватись в ході експерименту)), максимальна кількість ітерацій – 50), а результати позначаються салатовими гексагонами (рис. 4). DBSCAN використовує параметри $\epsilon = 0.01$ та $MinPts = 2$, що дозволяє автоматично виділяти як кластери, так і шумові точки; його результати відображаються фіолетовими ромбами (рис. 4). Обидва алгоритми застосовуються до однієї множини перетинів для подальшого порівняльного аналізу.

Експериментальна серія складається з трьох етапів, що відрізняються кількістю реальних цілей у сцені. Конфігурація системи (1 основний та 4 додаткові радари, радіуси 10 одиниць) залишається сталою для всіх випробувань. Параметри алгоритму DBSCAN також фіксовані: $\epsilon = 0.01$, $MinPts = 2$. Для алгоритму K-means параметр K змінюється відповідно до

кількості цілей, тоді як $MaxIterations = 50$ залишається незмінним.

Кожен з трьох етапів містить серію з 30 незалежних повторів, у межах яких цілі генеруються випадковим чином. Етапи відрізняються таким чином:

- етап 1: 30 ітерацій при кількості цілей $N = 5$;
- етап 2: 30 ітерацій при кількості цілей $N = 10$;
- етап 3: 30 ітерацій при кількості цілей $N = 20$.

Таке багаторазове повторення забезпечує статистичну надійність отриманих результатів і дозволяє коректно застосувати критерій Вілкоксона для подальшого порівняння алгоритмів.

Обчислення метрик якості алгоритмів. Для автоматизованого аналізу результатів кластеризації в моделі реалізовано окремий модуль ClusteringMetricsEvaluator. Його завданням є обчислення обраних метрик якості для двох алгоритмів (K-means та DBSCAN), фіксація результатів кожного запуску експерименту у вигляді таблиці та збереження візуального стану сцени. Процес експерименту повністю автоматизований: скрипт послідовно виконує генерацію перетинів, кластеризацію DBSCAN і K-means, після чого метод EvaluateAll() формує вибірки, обчислює метрики та виводить їх у консоль. Результати кожного запуску дописуються до CSV-файлу clustering_results.csv у Assets/Resources, де фіксуються номер експерименту, кількість істинних цілей і всі метрики для обох алгоритмів. Нумерація експериментів зберігається між перезапусками редактора через PlayerPrefs. Також автоматично зберігається скріншот сцени у файлі screenshot_experiment_<id>.png у тій самій директорії.

ОБГОВОРЕННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

У дослідженні виконано три серії експериментів із різною кількістю цілей (5, 10 та 20), по 30 ітерацій у кожній. Результати наведені у таблицях 3–5 та ілюстровані скріншотами експериментальних сцен [11].

Отримані дані доцільно співвіднести з відомими результатами в літературі. У роботах з bearing-only локалізації. Зокрема, праця «Evaluation of Clustering Algorithms for Radar-Based Object Recognition» [6] де акцент робиться на високій чутливості алгоритмів, що вимагають фіксовану кількість кластерів (таких як K-means), до шумових перетинів і псевдо-цілей. У межах цих досліджень підкреслюється, що методи, які не передбачають наперед заданої кількості кластерів і враховують щільність розташування точок (DBSCAN, OPTICS), демонструють вищу робастність за збільшення рівня шуму. Отримані нами результати повністю підтверджують ці висновки. Порівняння інтегральних метрик якості (A_{total}) для алгоритмів K-means та DBSCAN показує стабільну перевагу DBSCAN у всіх експериментальних режимах. Візуальний аналіз (рис. 3) свідчить, що DBSCAN краще відтворює реальні позиції цілей, формуючи компактні та семантично коректні кластери, тоді як K-means

Таблиця 3. Результати експерименту з 5 реальними цілями

id	true_ targets	kmeans_ Aloc	kmeans_ Aangle	kmeans_ Acount	kmeans_ Atotal	dbscan_ Aloc	dbscan_ Aangle	dbscan_ Acount	dbscan_ Atotal
1	5	0,9997	0,8538	1	0,9545	1	1	1	1
2	5	0,9954	0,2174	1	0,7547	1	1	1	1
3	5	0,9994	0,4814	1	0,8389	1	1	1	1
4	5	0,9989	0,3964	1	0,8122	1	1	1	1
5	5	0,9996	0,8374	1	0,9494	1	1	1	1
6	5	0,9978	0,0025	1	0,6895	1	1	1	1
7	5	0,9983	0,0093	1	0,6919	1	1	1	1
8	5	0,9975	0,0504	1	0,7042	1	1	1	1
9	5	0,9979	0,043	1	0,7021	1	1	0,8	0,978
10	5	0,9986	0,4476	1	0,8279	1	1	0,8	0,978
11	5	0,9996	0,3342	1	0,7933	1	1	1	1
12	5	0,9965	0,0336	1	0,6984	1	1	1	1
13	5	0,9998	0,9119	1	0,9726	1	1	1	1
14	5	0,9978	0,1398	1	0,7321	1	1	1	1
15	5	0,9989	0,3733	1	0,8051	1	1	0,8	0,978
16	5	0,9989	1	1	0,9994	1	1	1	1
17	5	0,9962	0,1332	1	0,7291	1	1	1	1
18	5	0,9993	0,6376	1	0,8873	1	1	1	1
19	5	0,9996	0,6234	1	0,883	1	1	0,6	0,956
20	5	0,9997	0,7455	1	0,9209	1	1	0,8	0,978
21	5	0,9988	0,1923	1	0,7489	1	1	1	1
22	5	0,9996	0,4193	1	0,8198	1	1	0,8	0,978
23	5	1	0,9738	1	0,9919	1	1	1	1
24	5	1	1	1	1	0,9962	0,0049	0,8	0,6673
25	5	0,9995	0,7207	1	0,9131	1	1	1	1
26	5	0,9983	0,1797	1	0,7447	1	1	1	1
27	5	0,9991	0,41	1	0,8166	1	1	1	1
28	5	0,9993	0,0319	1	0,6995	0,996	0,0598	0,8	0,6842
29	5	0,9989	0,3218	1	0,7891	1	1	1	1
30	5	0,9966	0,7476	1	0,9198	1	1	1	1

Таблиця 4. Результати експерименту з 10 реальними цілями

id	true_ targets	kmeans_ Aloc	kmeans_ Aangle	kmeans_ Acount	kmeans_ Atotal	dbscan_ Aloc	dbscan_ Aangle	dbscan_ Acount	dbscan_ Atotal
1	10	0,9981	0,1364	1	0,7312	1	1	0,8	0,978
2	10	0,9994	0,3907	1	0,8107	1	1	1	1
3	10	0,9994	0,5836	1	0,8706	1	1	0,8	0,978
4	10	0,9994	0,579	1	0,8692	1	1	1	1
5	10	0,9974	0,1016	1	0,72	1	1	0,9	0,989
6	10	0,999	0,6701	1	0,8972	1	1	1	1
7	10	0,998	0,0432	1	0,7022	1	1	1	1
8	10	0,9985	0,3498	1	0,7976	1	1	0,9	0,989
9	10	0,9971	0,0503	1	0,7039	1	1	0,8	0,978
10	10	0,9987	0,3	1	0,7822	1	1	0,8	0,978
11	10	0,9971	0,1805	1	0,7443	1	1	1	1
12	10	0,9985	0,1368	1	0,7315	1	1	0,9	0,989
13	10	0,9978	0,0394	1	0,701	1	1	1	1
14	10	0,9982	0,1673	1	0,7408	1	1	0,8	0,978
15	10	0,9979	0,0601	1	0,7074	1	1	0,9	0,989
16	10	0,9976	0,1164	1	0,7247	1	1	0,9	0,989
17	10	0,999	0,0278	1	0,6981	1	1	0,9	0,989
18	10	0,9983	0,3372	1	0,7936	1	1	1	1
19	10	0,9992	0,3717	1	0,8048	1	1	1	1
20	10	0,9986	0,438	1	0,825	1	1	0,9	0,989
21	10	0,9984	0,1282	1	0,7288	1	1	1	1
22	10	0,9994	0,561	1	0,8636	1	1	1	1
23	10	0,9991	0,5313	1	0,8542	1	1	0,9	0,989
24	10	0,9976	0,0445	1	0,7024	1	1	1	1
25	10	0,9961	0,0026	1	0,6886	1	1	0,8	0,978
26	10	0,9987	0,3758	1	0,8058	1	1	0,9	0,989
27	10	0,9986	0,3035	1	0,7833	1	1	0,9	0,989
28	10	0,9992	0,4393	1	0,8257	1	1	0,9	0,989
29	10	0,9993	0,5053	1	0,8462	1	1	0,9	0,989
30	10	0,9974	0,3148	1	0,7861	1	1	0,9	0,989

Таблиця 5. Результати експерименту з 20 реальними цілями

id	true_targets	kmeans_Aloc	kmeans_Aangle	kmeans_Acount	kmeans_Atotal	dbscan_Aloc	dbscan_Aangle	dbscan_Acount	dbscan_Atotal
1	20	0,9975	0,1527	1	0,7359	1	1	1	1
2	20	0,9981	0,1345	1	0,7306	1	1	0,85	0,9835
3	20	0,998	0,2722	1	0,7732	1	1	0,95	0,9945
4	20	0,9984	0,2318	1	0,7609	1	1	0,95	0,9945
5	20	0,9987	0,537	1	0,8557	1	1	0,9	0,989
6	20	0,9981	0,1109	1	0,7233	1	1	0,35	0,9285
7	20	0,9978	0,1038	1	0,7209	1	1	1	1
8	20	0,9978	0,3181	1	0,7874	1	1	0,95	0,9945
9	20	0,9992	0,4508	1	0,8293	1	1	0,8	0,978
10	20	0,9973	0,0876	1	0,7156	1	1	0,9	0,989
11	20	0,9988	0,3306	1	0,7918	1	1	0,85	0,9835
12	20	0,9982	0,5063	1	0,8459	1	1	1	1
13	20	0,9989	0,2539	1	0,7681	1	1	0,7	0,967
14	20	0,9979	0,2559	1	0,7681	1	1	1	1
15	20	0,9978	0,1463	1	0,7341	1	1	0,95	0,9945
16	20	0,9964	0,0173	1	0,6933	1	1	0,85	0,9835
17	20	0,9979	0,2946	1	0,7801	1	1	0,9	0,989
18	20	0,9977	0,1011	1	0,72	1	1	1	1
19	20	0,9988	0,3825	1	0,8079	1	1	0,75	0,9725
20	20	0,9978	0,0883	1	0,7161	1	1	0,85	0,9835
21	20	0,9978	0,2982	1	0,7812	1	1	0,75	0,9725
22	20	0,9978	0,2339	1	0,7612	1	1	0,85	0,9835
23	20	0,9966	0,0401	1	0,7005	1	1	0,95	0,9945
24	20	0,9985	0,321	1	0,7887	1	1	0,75	0,9725
25	20	0,9987	0,1915	1	0,7486	1	1	0,9	0,989
26	20	0,9985	0,6044	1	0,8765	1	1	0,75	0,9725
27	20	0,9986	0,1938	1	0,7492	1	1	0,9	0,989
28	20	0,999	0,2322	1	0,7614	1	1	0,8	0,978
29	20	0,9987	0,2539	1	0,7679	1	1	0,9	0,989
30	20	0,9981	0,1021	1	0,7206	1	1	0,95	0,9945

у ряді випадків утворює хибні або зміщені центри через прив'язку до випадкових спуріозних перетинів.

Для формального підтвердження відмінностей між результатами алгоритмів було застосовано критерій Вілкоксона для зв'язаних вибірок. Цей критерій дозволяє оцінити, чи є різниця між двома методами (K-means та DBSCAN) статистично значущою, не роблячи припущень щодо нормальності розподілу. Для кожного сценарію обчислювалися такі статистики:

- 5 цілей: $n = 30$, $W^+ = 438$, $W^- = 27$, $T = 27$ $p < 0.001$;
- 10 цілей: $n = 30$, $W^+ = 465$, $W^- = 0$, $T = 0$, $p < 0.001$;
- 20 цілей: $n = 30$, $W^+ = 465$, $W^- = 0$, $T = 0$, $p < 0.001$.

Де W^+ – сума рангів позитивних різниць (коли значення DBSCAN перевищує значення K-means); W^- – сума рангів негативних різниць (коли K-means дає кращий результат); $T = \min(W^+, W^-)$ – статистика критерію Вілкоксона, що використовується для визначення p-value; p-value – ймовірність того, що спостережена різниця між алгоритмами могла виникнути випадково. У всіх трьох випадках значення p-value виявилися меншими за 0.001, що свідчить про статистично значущу

різницю між алгоритмами. Це означає, що отримані відмінності не є випадковими та з високою ймовірністю відображають реальну перевагу DBSCAN над K-means у розглянутих умовах.

ВИСНОВКИ

1. В роботі проведено оцінювання ефективності алгоритмів кластеризації K-means та DBSCAN в умовах шумних азимутальних вимірювань у bearing-only задачах за допомогою статистичного критерію Вілкоксона. Результати оцінювання показали, що алгоритм DBSCAN, в порівнянні з K-means, забезпечує кращу та стабільнішу оцінку положення цілей. 2. Практична значимість роботи полягає в тому, що отримані результати дозволяють обґрунтовано обирати методи кластеризації для систем, які працюють з азимутальними вимірюваннями, а запропонована система метрик може бути використана для порівняння та налаштування алгоритмів у мультирадарних конфігураціях. 3. Перспективи подальших досліджень пов'язані з розробкою моделей кластеризації, здатних безпосередньо враховувати негаусівські розподіли похибок азимутальних вимірювань, а також із створенням адаптивних і комбінованих методів, які краще працюватимуть за умов нерівномірної густини, асиметричного шуму та багатоточкових спостережень.

REFERENCES

- [1] Zhang, W., Liu, Y., & Wang, Z. (2023). A novel density-based clustering method for effective removal of spurious intersections in bearings-only localization. *ResearchGate*. Retrieved from <https://www.researchgate.net/publication/368165301>
- [2] Jain, A., Rath, K., Ganguly, Y., Kumar, A., & Bhale, Y. (2025). A comparative analysis of DBSCAN, K-means and agglomerative clustering algorithms for geospatial data. *Atlantis Press*. https://doi.org/10.2991/978-94-6463-716-8_18
- [3] Ma, B., Yang, C., Li, A., Chi, Y., & Chen, L. (2023). A faster DBSCAN algorithm based on self-adaptive parameter determination. *Procedia Computer Science*. <https://doi.org/10.1016/j.procs.2023.07.017>

- [4] Huang, Z., Liang, Z., Zhou, S., & Zhang, S. (2023). An improved adaptive DBSCAN with Sparrow Search Algorithm (SSA-DBSCAN). *Algorithms*, 18(5), 273. <https://doi.org/10.3390/a18050273>
- [5] Hero, A., & Blatt, D. (2005). Multiple-source localization using line-of-bearing measurements: Approaches to the data association problem. *IEEE Transactions on Signal Processing*. Retrieved from <https://www.researchgate.net/publication/224372567>
- [6] Hernandez, J., et al. (2024). Evaluation of clustering algorithms for radar-based object recognition. *Sensors*, 24(22), 7219. <https://doi.org/10.3390/s24227219>
- [7] Kreshtanovych, N. (2023). Porivniannia metodiv klasteryzatsii dlia prostorovykh danykh [Comparison of clustering methods for spatial data]. *Chernivetskyi natsionalnyi universytet – Chernivtsi National University*. Retrieved from <https://archer.chnu.edu.ua/handle/123456789/9948> [in Ukrainian].
- [8] Redko, D., & Desiatko, A. (2024). Ansambl alhorytmiv klasteryzatsii merezhevoho trafiku z riznymi metrykamy [An ensemble of clustering algorithms with various metrics for network traffic analysis]. *Cybersecurity: Education, Science, Technique*, 4(26), 76–92. <https://doi.org/10.28925/2663-4023.2024.26.790> [in Ukrainian].
- [9] Saaty, T. L. (2001). *Decision Making for Leaders: The Analytic Hierarchy Process for Decisions in a Complex World*. Pittsburgh: RWS Publications.
- [10] Hollander, M., Wolfe, D. A., & Chicken, E. (2013). *Nonparametric Statistical Methods* (3rd ed.). Wiley.
- [11] Klieptsov, A. (2024). *DBSCAN_vs_KMeans_BearingOnly_Experiments*. Zenodo. <https://doi.org/10.5281/zenodo.18723019>

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- [1] Zhang W., Liu Y., Wang Z. (2023) A novel density-based clustering method for effective removal of spurious intersections in bearings-only localization. *ResearchGate*. URL: <https://www.researchgate.net/publication/368165301>.
- [2] Jain A., Rath K., Ganguly Y., Kumar A., Bhale Y. (2025) A comparative analysis of DBSCAN, K-means and agglomerative clustering algorithms for geospatial data. *Atlantis Press*. DOI: https://doi.org/10.2991/978-94-6463-716-8_18.
- [3] Ma B., Yang C., Li A., Chi Y., Chen L. (2023) A faster DBSCAN algorithm based on self-adaptive parameter determination. *Procedia Computer Science*. DOI: <https://doi.org/10.1016/j.procs.2023.07.017>.
- [4] Huang Z., Liang Z., Zhou S., Zhang S. (2023) An improved adaptive DBSCAN with Sparrow Search Algorithm (SSA-DBSCAN). *Algorithms*, 18(5), 273. DOI: <https://doi.org/10.3390/a18050273>.
- [5] Hero A., Blatt D. (2005) Multiple-source localization using line-of-bearing measurements: Approaches to the data association problem. *IEEE Transactions*. URL: <https://www.researchgate.net/publication/224372567>.
- [6] Hernandez J., та ін. (2024) Evaluation of clustering algorithms for radar-based object recognition. *Sensors*, 24(22), 7219. URL: <https://www.mdpi.com/1424-8220/24/22/7219>.
- [7] Крештанович Н. (2023) Порівняння методів кластеризації для просторових даних. Чернівецький національний університет. URL: <https://archer.chnu.edu.ua/handle/123456789/9948>.
- [8] Редько Д., Десятко А. (2024) Ансамбль алгоритмів кластеризації мережевого трафіку з різними метриками. *Cybersecurity: Education, Science, Technique*, 4(26). DOI: <https://doi.org/10.28925/2663-4023.2024.26.790>.
- [9] Саати, Т. Л. (2001). *Прийняття рішень для керівників: Метод аналізу ієрархій у складних системах*. Піттсбург: RWS Publications.
- [10] Голландер, М., Вулф, Д. А., Чікен, Е. (2013). *Nonparametric Statistical Methods*. Wiley.
- [11] Klieptsov A. (2024) *DBSCAN_vs_KMeans_BearingOnly_Experiments*. Zenodo. DOI: <https://doi.org/10.5281/zenodo.18723019>.

© Фаріонова Т. А., Клепцов А. А.

Дата першого надходження статті до видання: 13.02.2026

Дата прийняття статті до друку після рецензування: 10.03.2026

Дата публікації (оприлюднення) статті: 22.05.2026



Стаття поширюється на умовах ліцензії відкритого доступу (CC BY 4.0)