

Харківський національний університет радіоелектроніки

Факультет Навчально-науковий центр заочної форми навчання

Кафедра Штучного інтелекту

Рівень вищої освіти перший (бакалаврський)

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

Тип програми освітньо-професійна

Освітня програма Штучний інтелект
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«_____» _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Плотницькому Олександру Михайловичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка та аналіз адаптивного метода нечіткої кластеризації даних за умов апріорної та поточної невизначеності

затверджена наказом університету від 18 травня 2026 р. № 76Стз

2. Термін подання здобувачем роботи до екзаменаційної комісії 10 липня 2026 р.

3. Вихідні дані до роботи Науково-технічні публікації, монографії та матеріали конференцій з нечіткої кластеризації даних, алгоритму Gustafson-Kessel, потокового оброблення даних, адаптивної нормалізації та виявлення аномалій; дані відкритих наукових наборів даних і публічних Інтернет-джерел; документація мов програмування та програмних бібліотек для реалізації й експериментального дослідження алгоритмів кластеризації.

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Аналіз предметної галузі та постановка задачі

2) Розробка метода нечіткої кластеризації даних за умов апріорної та поточної невизначеності

3) Програмна реалізація

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / термін виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	18.05.2026	виконано
2	Аналіз предметної області та постановка задачі	31.05.2026	виконано
3	Дослідження алгоритмів та обґрунтування вибору базового алгоритму	07.06.2026	виконано
4	Розроблення адаптивного методу потокової нечіткої кластеризації	16.06.2026	виконано
5	Розробка методів вирішення проблем апріорної та поточної невизначеності	23.06.2026	виконано
6	Програмна реалізація методу та експериментальне моделювання на синтетичних наборах даних	25.06.2026	виконано
7	Аналіз результатів експериментів	27.06.2026	виконано
8	Написання пояснювальної записки	02.07.2026	виконано
9	Нормоконтроль	04.07.2026	виконано
10	Перевірка на академічний плагіат	05.07.2026	виконано
11	Підготовка презентації та доповіді	06.07.2026	виконано
12	Попередній захист	07.07.2026	виконано
13	Рецензування	08.07.2026	виконано
14	Захист перед ЕК	10.07.2026	

Дата видачі завдання 18 травня 20 26 р.

Здобувач 
(підпис)

Керівник роботи _____
(підпис)

ас. Максим ПОЛІТ
(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 76 с., 11 рис., 2 дод., 15 джерел.

АДАПТАЦІЯ КІЛЬКОСТІ КЛАСТЕРІВ, АДАПТИВНА МЕТРИКА, АЛГОРИТМ GUSTAFSON-KESSEL, АНОМАЛЬНІ СПОСТЕРЕЖЕННЯ, БУФЕР КАНДИДАТІВ, НЕЧІТКА КЛАСТЕРИЗАЦІЯ, ОНЛАЙН-ОБРОБЛЕННЯ, ПОТОКОВА НОРМАЛІЗАЦІЯ, ПОТОКОВІ ДАНІ, РЕКУРСИВНЕ ОНОВЛЕННЯ.

Об'єкт дослідження – нечітка кластеризація поточкових даних.

Предмет дослідження – алгоритми нечіткої кластеризації поточкових даних на основі алгоритму Gustafson-Kessel та методи їх адаптації до апіорної і поточної невизначеності.

Мета роботи – дослідження наявних підходів до нечіткої кластеризації, розробка та програмна реалізація алгоритму на основі алгоритму Gustafson-Kessel для послідовного оброблення спостережень.

Методи дослідження – аналіз наукових публікацій і технічної документації, дослідження методів нечіткої кластеризації, математичного моделювання, вивчення існуючих рішень, що використовуються в інших алгоритмах.

У ході кваліфікаційної роботи було проаналізовано підходи до нечіткої кластеризації, зокрема алгоритм Gustafson-Kessel. Розроблено метод, що виконує послідовне оброблення даних і рекурсивне оновлення параметрів кластерів. Реалізовано програмний засіб для проведення експериментальних досліджень з поточковими та синтетичними наборами даних. Проведено оцінювання результатів роботи розробленого методу.

ABSTRACT

Bachelor's thesis contains: 76 pp., 11 fig., 2 ann., 15 references.

ADAPTIVE METRIC, ANOMALOUS OBSERVATIONS, CANDIDATE BUFFER, CLUSTER NUMBER ADAPTATION, DATA STREAMS, FUZZY CLUSTERING, GUSTAFSON-KESSEL ALGORITHM, ONLINE PROCESSING, RECURSIVE UPDATING, STREAMING NORMALIZATION.

The object of research is fuzzy clustering of data streams.

The subject of research is fuzzy data stream clustering algorithms based on the Gustafson-Kessel algorithm and methods for their adaptation to a priori and streaming uncertainty.

The purpose of the work is to study existing approaches to fuzzy clustering and to develop and implement an algorithm based on the Gustafson-Kessel algorithm for the sequential processing of observations.

Research methods include the analysis of scientific publications and technical documentation, the study of fuzzy clustering and mathematical modelling methods, and the examination of existing solutions applied in related algorithms.

In the course of the qualification work, approaches to fuzzy clustering were analyzed, including the Gustafson-Kessel algorithm. An algorithm for sequential data processing and recursive updating of cluster parameters was developed. A software tool for conducting experiments on streaming and synthetic datasets was implemented. The performance of the developed algorithm was evaluated.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	7
Вступ.....	8
1 Аналіз предметної галузі та постановка задачі.....	9
1.1 Теоретичні основи адаптивної нечіткої кластеризації потоків даних.	9
1.1.1 Кластеризація даних та подання об'єктів у просторі ознак	10
1.1.2 Чітка та нечітка кластеризація.....	12
1.1.3 Особливості поточкових даних	14
1.1.4 Апріорна та поточна невизначеність	17
1.2 Огляд класичних, нечітких методів кластеризації	19
1.2.1 Алгоритм K-means	19
1.2.2 Метод Fuzzy C-Means	20
1.2.3 Алгоритм Gustafson-Kessel	22
1.3 Постановка задачі.....	27
2 Розробка метода нечіткої кластеризації даних за умов апріорної та поточної невизначеності.....	29
2.1 Вимоги до розроблюваного методу та обґрунтування вибору базового алгоритму	29
2.2 Принцип потокового оновлення параметрів моделі	31
2.3 Формування початкової моделі на стартовому фрагменті потоку	34
2.4 Потокова нормалізація та переведення параметрів кластерів у нову систему координат	35
2.5 Адаптація кількості кластерів і оброблення потенційно аномальних спостережень	38
2.6 Узагальнена схема роботи розробленого методу	44
3 Програмна реалізація	48
Висновки	63
Перелік джерел посилення	65
Додаток А Детальні результати серій експериментів	67
Додаток Б Відомість кваліфікаційної роботи.....	76

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ІІІ – штучний інтелект;

ACI – Adjusted Concordance Index – скоригований індекс узгодженості для нечітких розбиттів;

AI – Artificial Intelligence – штучний інтелект;

ARI – Adjusted Rand Index – скоригований індекс Ренда;

FCM – Fuzzy C-Means – алгоритм нечітких с-середніх;

GK – Gustafson-Kessel – алгоритм нечіткої кластеризації з адаптивною метрикою;

Warm-up – початковий фрагмент потоку, що використовується для формування стартової моделі.

ВСТУП

Сучасний розвиток інформаційних технологій супроводжується постійним зростанням обсягів даних, що надходять із різних джерел: сенсорних систем, мережевих сервісів, фінансових операцій, виробничих процесів та інших інформаційних середовищ. Значна частина таких даних надходить не у вигляді фіксованих наборів, а у формі потоків, характеристики яких можуть змінюватися з часом. Для їх обробки необхідно використовувати методи аналізу, що здатні поступово оновлювати модель у процесі надходження нових спостережень і враховувати обмеженість в пам'яті та часі необхідних для обробки [5], [6].

Однією з важливих задач інтелектуального аналізу даних є кластеризація, тобто виявлення груп подібних об'єктів без заздалегідь заданих міток класів. У реальних задачах межі між такими групами часто є нечіткими. У таких умовах доцільним є використання нечіткої кластеризації, яка дозволяє описувати ступінь належності об'єкта до різних кластерів [1], [2].

Особливу складність становить кластеризація поточкових даних за умов апріорної та поточної невизначеності. Апріорна невизначеність пов'язана з відсутністю повної інформації про структуру даних до початку обробки, зокрема про кількість кластерів, їх форму, щільність та наявність аномалій. Поточна невизначеність виникає під час обробки потоку і проявляється у зміні характеристик даних, появі нових кластерів, зникненні старих або зміщенні меж між ними.

Актуальність роботи полягає в необхідності розробки та аналізу адаптивного методу нечіткої кластеризації потоків даних, здатного працювати в умовах невідомої або змінної структури кластерів, наявності аномалій. Такий метод має поєднувати переваги нечіткого представлення кластерної структури з можливістю інкрементального оновлення моделі в онлайн-режимі.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Теоретичні основи адаптивної нечіткої кластеризації потоків даних

Однією із найчастіших проблем інтелектуального аналізу можна вважати необхідність виявлення прихованих патернів та структур. Також ця задача може бути ускладнена як розміром самого масиву інформації, так і відсутністю чітких, відомих наперед міток класів.

Кластеризація використовується для неконтрольованої класифікації цих даних і їх групування на основі подібності у кластери. Нечіткий підхід дозволяє коректніше опрацьовувати такі набори даних, де деякі об'єкти мають властивості, що притаманні для різних класів. При вирішенні таких проблем, де межі між групами не є чіткими, належність елементу до класу задається ступенем належності в інтервалі від 0 до 1. Що робить систему гнучкішою ніж при використанні бінарних значень [2].

Ще одним ускладненням стає обробка поточкових даних. На відміну від статичних наборів, характеристики таких даних можуть змінюватись з часом. Ці дані, що надходять послідовно зумовлюють необхідність використання адаптивних методів, що працюють з урахуванням обмежень в пам'яті та часу.

Таким чином, задача адаптивної нечіткої кластеризації потоків даних поєднує в собі такі напрями як: методи кластерного аналізу, теорію нечітких множин та методи онлайн-обробки потокової інформації. Для послідовного розкриття теоретичних основ цієї теми доцільно спочатку розглянути загальне поняття кластеризації даних та основи нечіткого підходу, далі розглянути особливості опрацювання поточкових даних і проблему невизначеності.

1.1.1 Кластеризація даних та подання об'єктів у просторі ознак

Кластеризація є методом інтелектуального аналізу даних і машинного навчання, з класичним прикладом навчання без учителя, призначеним для виявлення внутрішньої структури набору спостережень. Її метою є поділ об'єктів на групи – кластери таким чином, щоб об'єкти одного кластера були більш подібними між собою за обраними ознаками, ніж до об'єктів інших кластерів. На відміну від задач класифікації, під час кластеризації вихідні дані не містять наперед заданих міток класів. Тому алгоритм має самостійно визначити можливу структуру даних на основі відстаней, подібності, густини розподілу або статистичних властивостей об'єктів.

У загальному вигляді для кластеризації задано набір об'єктів в такому виді ($X = \{x_1, x_2, \dots, x_n\}$), де кожний об'єкт описується набором числових або категоріальних ознак. А результатом роботи алгоритму є множина кластерів ($C = \{C_1, C_2, \dots, C_k\}$). У жорсткій кластеризації кожний об'єкт належить лише одному кластеру. У нечіткій кластеризації об'єкт може одночасно належати кільком кластерам із різним ступенем належності, що дає змогу точніше описувати об'єкти, розташовані поблизу меж між групами.

Класифікація, навпаки, є задачею навчання з учителем. Для неї використовується навчальна вибірка виду $((x_i, y_i))$, у якій для кожного об'єкта (x_i) уже відома правильна мітка класу (y_i). На основі цих даних будується модель, що надалі визначає клас нових об'єктів. Отже, класифікація відтворює відому структуру класів, тоді як кластеризація виявляє структуру, яка до початку аналізу невідома.

Важливо, що номер кластера не є змістовною міткою класу: наприклад, кластери з номерами 1, 2 і 3 лише позначають про належність об'єктів до різних груп. Їх інтерпретація здійснюється після отримання результату кластеризації.

Зазвичай, кластеризація працює з об'єктами як з наборами ознак. Одним з найзручніших способів такого представлення об'єкту – вектор ознак [2].

$$x_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{ip}), \quad (1.1)$$

де p – кількість ознак.

Сукупність компонентів цього вектору x_i визначає положення вектору в просторі ознак, p – вимірному просторі що представляє собою сукупність усіх можливих значень векторів, де кожна координата відповідає окремій ознаці. Якщо об'єкт описується числовими ознаками, то його можна подати як точку в цьому просторі. У двовимірному випадку об'єкти відображаються точками на площині (рисунок 1.1), у тривимірному – точками у просторі. Для більшої кількості ознак геометричне зображення ускладнюється, проте математичний принцип залишається тим самим: близько розташовані точки розглядаються як потенційно подібні об'єкти.

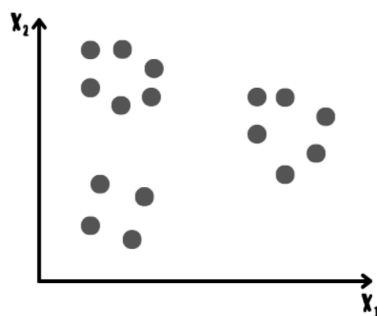


Рисунок 1.1 – Приклад двовірних об'єктів на площині

Для формування кластерів необхідно кількісно оцінити ступінь близькості об'єктів. Для цього використовують міри подібності або міри відстані, це правило, або функція, яка за двома об'єктами повертає число. Міра подібності $s(x_i, x_j)$ набуває більшого значення для більш схожих об'єктів. Відстань $d(x_i, x_j)$, навпаки, є мірою відмінності: чим менше її

значення, тим ближчими є об'єкти за сукупністю ознак. У багатьох алгоритмах кластеризації саме відстань між векторами ознак, а тобто між відповідними точками у p – вимірному просторі є основою для визначення належності об'єкта до кластера [2].

Однією з найпоширеніших мір є евклідова відстань. Для двох об'єктів (x_i) та (x_j) , заданих (p) числовими ознаками, вона обчислюється за формулою [2]:

$$d_E(x_i, x_j) = \sqrt{\sum_{k=1}^p (x_{ik} - x_{jk})^2}. \quad (1.2)$$

У задачах, де необхідно лише порівнювати відстані між об'єктами, часто використовують квадрат евклідової відстані. Це дає змогу уникнути обчислення квадратного кореня, оскільки функція квадратного кореня є монотонно зростаючою.

Отже, евклідова відстань відповідає звичайній геометричній відстані між двома точками у просторі ознак. Вона є доцільною, коли всі ознаки мають числовий характер і порівнювані масштаби. Якщо ознаки вимірюються в різних одиницях або мають суттєво різні діапазони значень, перед обчисленням відстаней необхідно виконати нормалізацію або стандартизацію даних. Інакше ознака з більшим числовим масштабом може непропорційно впливати на результат кластеризації.

1.1.2 Чітка та нечітка кластеризація

Залежно від способу призначення об'єктів до кластерів розрізняють чітку та нечітку кластеризацію. У разі чіткої кластеризації кожен об'єкт може належати тільки одному кластеру [1], [2]. Кожен кластер можна описати як набір об'єктів що алгоритм визначив належними до нього. А кожен об'єкт має бінарний масив довжиною в кількість кластерів. Де 0

означає неналежність об'єкту до кластеру, а 1 належність. І сума всіх об'єктів масиву також дорівнює 1. Це перевірка на випадкову помилку.

Такий підхід є простим для інтерпретації, однак має суттєвий недолік: він примусово відносить об'єкт лише до одного кластера навіть тоді, коли об'єкт розташований поблизу межі між кількома групами. Наприклад, якщо точка має майже однакову відстань до центрів двох кластерів, алгоритм чіткої кластеризації все одно повинен вибрати один із них. У результаті не відображається невизначеність належності такого об'єкта та втрачається інформація про його подібність до іншого кластера.

У нечіткій кластеризації об'єкт може належати одночасно кільком кластерам із різними ступенями належності. У цьому випадку, кожен об'єкт отримує масив довжиною в кількість кластерів заповнений дійсними числами в проміжку від 0 до 1 [1], [2]. Але умова про те що сума має дорівнювати 1 залишилась, оскільки тепер ми розподіляємо нашу одиницю належності, по декільком кластерам. І для того, аби всі точки мали рівний об'єм в цільовій функції ми лишаємо цю умовну перевірку на правильність належності декільком кластерам.

Поняття нечіткої кластеризації ґрунтується на теорії нечітких множин. У класичній теорії множин елемент або належить множині, або не належить їй. Тобто належність описується лише двома значеннями: 0 або 1. Нечітка множина дозволяє описувати проміжні випадки, коли елемент має лише часткову належність до певної множини.

Нечітку множину (A) на універсальній множині (X) можна подати у вигляді сукупності пар [1], [2]:

$$A = \{ (x, \mu_A(x) \mid x \in X \}, \quad (1.3)$$

де $\mu_A(x)$ – функція належності елемента (x) до нечіткої множини (A). Значення цієї функції належить інтервалу від 0 до 1.

Якщо $\mu_A(x) = 1$, то елемент повністю належить множині. Якщо 0, то він не належить їй. Проміжні значення, наприклад 0,3, 0,5 або 0,8, означають часткову належність елемента до множини.

Для всього набору даних ступені належності записують у вигляді матриці нечітких належностей [1], [2]:

$$U = \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1c} \\ u_{21} & u_{22} & \dots & u_{2c} \\ \dots & \dots & \dots & \dots \\ u_{n1} & u_{n2} & \dots & u_{nc} \end{pmatrix}. \quad (1.4)$$

де n – кількість об'єктів;

c – кількість кластерів.

Кожний рядок цієї матриці відповідає одному об'єкту, а кожний стовпець – одному кластеру. Елемент (u_{ij}) показує ступінь належності (i) -го об'єкта до (j) -го кластера.

У класичних методах нечіткої кластеризації, зокрема в алгоритмах нечітких FCM і Gustafson-Kessel, для кожного об'єкта виконується умова нормування [1], [2].

Необхідність нечіткої кластеризації особливо помітна для точок, розташованих поблизу межі між кластерами. У разі чіткої кластеризації така точка буде примусово віднесена лише до одного кластера, навіть якщо вона майже однаково близька до двох різних груп. Нечіткий підхід дозволяє зберегти цю невизначеність у вигляді кількох ступенів належності. Завдяки цьому об'єкти проміжного типу описуються точніше, а результат кластеризації краще відображає реальну структуру даних у випадку перекриття або нечітких меж між кластерами.

1.1.3 Особливості потокових даних

На відміну від статичного набору даних, потік даних являє собою послідовність спостережень, що надходять у систему з плином часу. Кожне

нове спостереження може бути подане як вектор ознак x_t , де індекс t відображає момент або порядок його надходження. Джерелами таких потоків можуть бути показники датчиків, журнали подій інформаційних систем, мережевий трафік, фінансові операції або результати безперервних вимірювань.

У межах цього підходу набір даних не розглядається як повністю сформована таблиця, доступна алгоритму до початку обчислень. Натомість алгоритм одержує об'єкти послідовно та повинен формувати поточне уявлення про структуру даних у процесі їх надходження. Потік може тривати невизначений час, а порядок появи об'єктів є частиною умов задачі, оскільки він впливає на стан моделі в кожному конкретний момент [5], [6].

Першою принциповою особливістю потокового режиму є відсутність доступу до майбутніх даних. Під час оброблення об'єкта алгоритм має у своєму розпорядженні лише поточну точку, параметри сформованої моделі та, за потреби, обмежений обсяг збереженої інформації про попередні спостереження. Значення наступних об'єктів ще невідомі. Тому рішення щодо належності точки до кластера, оновлення центру або виявлення аномалії приймається на основі інформації, доступної на цей момент.

Після надходження нових даних це рішення може бути скориговане через оновлення параметрів моделі, але, зазвичай, не через повний повторний аналіз усіх уже отриманих даних [5], [6], [8].

Другою особливістю є обмеження часу на оброблення одного спостереження. З цієї причини поточкові алгоритми зазвичай орієнтовані на один або невелику кількість проходів через дані та уникають складних багатоітераційних перерахунків після надходження кожної точки [5], [6], [8].

Третім обмеженням є обмежений обсяг оперативної пам'яті. Потік може бути дуже великим або потенційно необмеженим, тому зберігати всі його об'єкти, їхні відстані та результати проміжних обчислень неможливо і

недоцільно. Замість повної історії алгоритм має зберігати компактний опис уже обробленої частини потоку. Такий підхід зменшує потребу в пам'яті, однак водночас створює ризик втрати частини інформації про ранні дані. Тому параметри, що зберігаються, мають бути достатніми для наближеного відтворення актуальної структури даних.

У зв'язку з цим модель кластеризації повинна оновлюватися інкрементально. Інкрементальне оновлення означає, що після надходження точки x_t новий стан моделі обчислюється на основі її попереднього стану та характеристик цієї точки. Така схема дозволяє адаптувати модель без зберігання та повторного опрацювання всієї історії потоку.

Потоковий режим принципово відрізняється від пакетного режиму оброблення. У пакетному режимі алгоритм має доступ до всього набору даних до початку навчання. Він може багаторазово переглядати об'єкти, виконувати глобальну нормалізацію, змінювати початкові параметри та повторювати ітерації до досягнення умови збіжності. Наприклад, класичний алгоритм Gustafson-Kessel на статичній вибірці може на кожній ітерації перераховувати ступені належності всіх об'єктів і параметри всіх кластерів [3], [5], [6].

У потоковому режимі така процедура зазвичай є непридатною, оскільки потребує доступу до всіх попередніх даних і значних обчислювальних витрат. Тому замість глобального переобчислення використовують локальні та рекурсивні оновлення параметрів. У результаті потокова кластеризація є компромісом між точністю опису накопичених даних, швидкістю реакції на нові спостереження та обсягом ресурсів, доступних алгоритму.

Ці обмеження створюють передумови для появи потокової невизначеності, адаптивної нормалізації та механізмів виявлення нових кластерів і аномальних об'єктів.

1.1.4 Априорна та поточна невизначеність

У задачах кластеризації невизначеність виникає через те, що структура даних не задається наперед і може змінюватися в процесі надходження нових спостережень. Для задачі адаптивної нечіткої кластеризації потоків даних доцільно розрізняти априорну та поточну невизначеність. Априорна невизначеність характеризує властивості даних, які невідомі до початку оброблення потоку. Поточна невизначеність виникає вже під час роботи алгоритму, коли нові спостереження можуть змінювати актуальне уявлення моделі про сформовані кластери. Такий поділ є робочою схемою, яка пов'язує властивості задачі з механізмами адаптації, передбаченими в методі під час практичного застосування до безперервного потоку даних.

Насамперед априорно невідомою є кількість кластерів. У багатьох класичних алгоритмах, зокрема Fuzzy C-Means і Gustafson-Kessel, число кластерів задається до початку навчання. Проте іноді в реальних даних відсутня достовірна інформація про те, скільки природних груп міститься у вибірці. Вибір занадто малого значення кількості кластерів може призвести до об'єднання різних груп в один кластер, а надто великого – до штучного подрібнення однорідної групи. Для потокового середовища ця проблема є ще складнішою, оскільки на початковому відрізку потоку можуть бути представлені не всі типи спостережень [6], [8], [9].

Також до початку аналізу невідомими є форма, розмір і щільність майбутніх кластерів. Кластери можуть бути близькими до сферичних, витягнутими, орієнтованими під різними кутами у просторі ознак або мати різну дисперсію. Щільність груп також може істотно відрізнятися: одна група може містити багато близько розташованих точок, тоді як інша є більш розсіяною. Через це використання єдиного фіксованого порогу відстані або припущення про однакову форму кластерів може бути некоректним. Для алгоритму Gustafson-Kessel ця особливість є суттєвою,

оскільки він використовує окрему матрицю форми для кожного кластера та може моделювати кластери еліпсоїдної форми [6], [8], [9].

Ще одним проявом апріорної невизначеності є невідомий рівень перекриття кластерів. Дві групи можуть бути добре відокремленими, частково перетинатися або мати значну спільну область у просторі ознак. У зоні перекриття чітке призначення об'єкта лише до одного кластера втрачає частину інформації про його подібність до інших груп. Саме тому нечіткі ступені належності є доцільними: вони дають змогу подати граничний об'єкт як такий, що пов'язаний з кількома кластерами різною мірою.

Окремою проблемою є наявність аномальних спостережень. До початку оброблення невідомо, чи віддалена точка є випадковим шумом, помилкою вимірювання або представником нової закономірної групи. Якщо кожен таку точку одразу включати до найближчого кластера, вона може спотворити його центр і геометричні параметри. Водночас передчасне оголошення точки аномалією може приховати початок формування нового кластера. Отже, алгоритм має розділяти одиничні віддалені спостереження та компактні групи подібних віддалених точок, що накопичуються з часом [6], [7], [9], [15].

Поточна невизначеність пов'язана зі зміною структури даних після запуску алгоритму. Одним із її проявів є поступове зміщення центрів кластерів. Наприклад, середні значення ознак об'єктів певної групи можуть змінюватися, тому раніше визначений центр уже не буде коректно представляти її поточне положення. Іншим проявом є зміна геометрії кластера: група може ставати більш витягнутою, стискатися, збільшувати або зменшувати розсіювання в окремих напрямках [8], [9].

У потоці також може з'явитися нова група спостережень, яка не відповідає жодному з наявних кластерів. На момент появи перші її точки можуть виглядати як аномалії, оскільки вони віддалені від поточних центрів. Віддалені точки, які не утворюють такої групи, доцільно вважати за дійсні аномалії.

1.2 Огляд класичних, нечітких методів кластеризації

1.2.1 Алгоритм K-means

Алгоритм K-means є одним із базових методів кластеризації, який використовується для поділу множини об'єктів на наперед визначену кількість кластерів – K і обирає для кожного з них початкові значення центрів [2]. Його метою є формування таких груп, щоб об'єкти всередині кожної з них були якомога ближчими до власного центроїда, а відстані до центрів інших кластерів були більшими. Центроїд v_j являє собою вектор середніх значень усіх ознак об'єктів, віднесених до кластера j .

Робота класичного алгоритму складається з початкового вибору K центроїдів і повторення двох основних кроків. На першому кроці для кожного об'єкта обчислюється відстань до всіх центроїдів. Зазвичай застосовують евклідову відстань або її квадрат. Об'єкт призначається до того кластера, центр якого є найближчим. На другому кроці для кожного сформованого кластера заново обчислюється центроїд як середнє арифметичне векторів ознак усіх віднесених до нього об'єктів. Ці кроки повторюються доти, поки положення центроїдів перестануть змінюватися, або буде досягнена максимальна кількість ітерацій. У результаті мінімізується сума квадратів відстаней об'єктів до центроїдів відповідних кластерів [2]. K-means є прикладом алгоритму чіткої кластеризації, для кожного об'єкта існує лише один кластер, до якого він належить. У термінах матриці належностей це означає, що u_{ij} набуває значення 1 для вибраного кластера та 0 для всіх інших. Така модель є простою, проте не відображає невизначеність для точок, розташованих у зоні перекриття або поблизу межі між групами. Навіть за майже однакових відстаней до двох центроїдів алгоритм обирає лише один кластер.

Іншим обмеженням є необхідність задавати значення кількості кластерів до початку роботи. Стандартна пакетна реалізація також

передбачає багаторазове переобчислення призначень і центрів на всій доступній вибірці. Тому вона не відповідає вимогам адаптивної нечіткої кластеризації потоків даних: не подає часткової належності, не визначає появу нової групи самостійно та потребує доступу до вже накопичених даних на ітераціях [2]. Отже, K-means доцільно розглядати як вихідний класичний метод, обмеження якого обґрунтовують використання нечітких і адаптивних підходів.

1.2.2 Метод Fuzzy C-Means

Метод Fuzzy C-Means, або FCM, є одним із базових алгоритмів нечіткої кластеризації. Він призначений для поділу множини об'єктів на задану кількість кластерів із урахуванням того, що межі між групами можуть бути нечіткими. На відміну від алгоритму K-means, Fuzzy C-Means не виконує жорсткого призначення кожного об'єкта лише до одного кластера. Замість цього для кожного об'єкта визначаються ступені його належності до всіх наявних кластерів [1], [2].

Результатом роботи алгоритму є множина центрів кластерів та матриця нечітких належностей U . Її елемент u_{ij} характеризує ступінь належності об'єкта до кластера. Значення належності це дійсні числа в інтервалі від 0 до 1. Чим ближчим є об'єкт до певного кластера, тим більшого значення набуває його належність до цього кластера. Для кожного об'єкта сума ступенів належності до всіх кластерів дорівнює одиниці. Таким чином, FCM розподіляє належність об'єкта між усіма кластерами, відображаючи його відносну близькість до кожного з них.

Робота методу має ітераційний характер. На початковому етапі задається кількість кластерів C , а також початкові значення центрів і матриці нечітких належностей. Далі для кожного об'єкта обчислюються відстані до центрів усіх кластерів. На основі цих відстаней визначаються

ступені належності об'єкта до кластерів: об'єкт отримує більшу належність до тих кластерів, центри яких розташовані ближче у просторі ознак [2].

Після обчислення належностей відбувається уточнення центрів кластерів. Новий центр кожного кластера визначається як зважене середнє об'єктів, причому вагами виступають їхні ступені належності до відповідного кластера. Точки з високою належністю мають більший вплив на положення центру, тоді як точки з низькою належністю впливають слабше. Після оновлення центрів належності перераховуються повторно, оскільки відстані між об'єктами та новими центрами змінюються. Цикл уточнення належностей і центрів продовжується до моменту, коли зміни параметрів стають меншими за заданий поріг або досягається максимальна кількість ітерацій.

Важливу роль у Fuzzy C-Means відіграє параметр нечіткості m . Він визначає, наскільки розподіл належностей між кластерами буде чітким або розмитим. За значень m , наближених до одиниці, алгоритм наближається до жорсткого розподілу: для більшості об'єктів один ступінь належності стає значно більшим за інші. У такому випадку поведінка FCM стає подібною до K-means. За збільшення значення m належності до різних кластерів вирівнюються, а межі між ними стають менш різкими. Надто велике значення параметра нечіткості може зробити поділ недостатньо виразним, оскільки багато точок матимуть близькі значення належностей до кількох кластерів. Отже, вибір параметру нечіткості впливає на характер отриманого нечіткого розбиття [2].

Перевагою FCM над K-means є можливість точніше описувати об'єкти, розташовані поблизу межі між кластерами і краще оцінювати їх вплив на уточнення центрів кластерів. Наприклад, точка може мати належності 0,55 та 0,45 до двох кластерів. Це означає, що вона дещо ближча до першої групи, однак також істотно пов'язана з другою. У K-means така точка була б безумовно призначена лише до одного кластера, а інформація про її близькість до іншої групи була б втрачена [2]. Тому FCM є доцільним

для даних із частковим перекриттям кластерів та об'єктами проміжного типу.

Водночас стандартний Fuzzy C-Means має обмеження, пов'язані з використанням евклідової відстані. Така метрика коректно описує близькість об'єктів, коли ознаки мають порівнювані масштаби, а кластери є близькими до сферичних і подібними за формою. Якщо групи мають витягнуту форму, різну дисперсію, різну орієнтацію у просторі ознак або між ознаками наявні кореляції, евклідова відстань може неточно відображати структуру даних. Унаслідок цього належності об'єктів і положення центрів можуть визначатися недостатньо коректно.

Стандартна реалізація FCM має пакетний характер. На кожній ітерації алгоритм повинен обчислювати належності всіх об'єктів вибірки та повторно уточнювати параметри всіх кластерів. Це потребує зберігання матриці належностей і доступу до всього набору даних. У потоковому середовищі, де спостереження надходять послідовно, майбутні дані відсутні, а зберігання повної історії може бути неможливим, класична схема FCM не може застосовуватися без модифікації. Для потокової нечіткої кластеризації необхідні інкрементальне оновлення параметрів, обмежене використання пам'яті та механізми реагування на віддалені або нові групи спостережень.

1.2.3 Алгоритм Gustafson-Kessel

Алгоритм Gustafson-Kessel, або GK, є розвитком методу Fuzzy C-Means, призначеним для нечіткої кластеризації даних зі складнішою геометричною структурою. Як і FCM, він формує нечітке розбиття множини об'єктів: кожний об'єкт не жорстко відноситься до одного кластера, а має ступені належності до всіх кластерів. Алгоритм також ітераційно уточнює ці ступені належності та положення центрів. Проте головна відмінність GK полягає не в самій нечіткій природі розбиття, а у способі вимірювання

відстані між об'єктом і кластером [3]. У FCM використовується однакова для всіх кластерів евклідова метрика. У GK для кожного кластера формується власна адаптивна метрика, яка враховує його форму та орієнтацію у просторі ознак.

Нехай набір даних містить n об'єктів, кожний з яких має p числових ознак. Як і в FCM, перед початком роботи задається кількість кластерів C , параметр нечіткості $m > 1$ та початкові значення матриці належностей і центрів кластерів. На кожній ітерації обчислюються центри, ступені належності та додаткові параметри, характерні саме для GK. До них належать нечіткі коваріаційні матриці F_j і матриці адаптивної метрики A для кожного кластера. Процес повторюється доти, доки зміни матриці належностей, центрів не стануть меншими за встановлену межу [3], [14].

Основою методу є нечітка коваріаційна матриця кластера. Для кластера J вона обчислюється як зважена оцінка розсіювання об'єктів навколо його центра:

$$F_j = \frac{\sum_{i=1}^n u_{ij}^m (x_i - v_j)(x_i - v_j)^T}{\sum_{i=1}^n u_{ij}^m}. \quad (1.5)$$

У цій формулі внесок кожного об'єкта зважується за значенням u_{ij}^m . Отже, точки з високим ступенем належності до кластера мають більший вплив на оцінювання його форми, тоді як граничні точки впливають слабше. Завдяки цьому параметри кластера визначаються насамперед його найбільш характерними об'єктами, а не всіма точками однаковою мірою.

Діагональні елементи матриці F_j характеризують розсіювання даних уздовж окремих ознак. Наприклад, велике значення на діагоналі означає, що об'єкти кластера суттєво відрізняються за відповідною ознакою. Позадіагональні елементи відображають коваріацію, тобто спільну зміну пар ознак. Якщо зі збільшенням однієї ознаки в межах кластера зазвичай збільшується інша, їхня коваріація є додатною. Якщо одна ознака зростає, а

інша зменшується, коваріація буде від'ємною. Таким чином, коваріаційна матриця містить інформацію не лише про масштаб розсіювання, а й про залежності між ознаками.

На основі нечіткої коваріаційної матриці формується матриця адаптивної метрики. У загальному вигляді вона визначається як:

$$A_j = [n_j \det(F_j)]^{-1/n} F_j^{-1}, \quad (1.6)$$

де $\det(F_j)$ – визначник коваріаційної матриці;

F_j^{-1} – обернена матриця;

$n_j > 0$ – параметр, який пов'язаний з обмеженням об'єму кластера.

Нормування через визначник не дозволяє метриці змінюватися довільно лише за масштабом [3]. У спрощених реалізаціях часто прирівнюють його до 1.

Квадрат адаптивної відстані від об'єкта x_i до центра v_j обчислюється за формулою

$$d_{ij}^2 = (x_i - v_j)^T A_j (x_i - v_j). \quad (1.7)$$

Така відстань є кластерно-специфічною: для одного й того самого об'єкта її значення може відрізнятися залежно від того, до якого кластера вона обчислюється. Якщо матриця адаптивної метрики є одиничною, формула переходить до звичайної суми квадратів різниць координат, тобто до квадрата евклідової відстані. Проте в ГК матриця A_j визначається окремо для кожного кластера, тому алгоритм може по-різному оцінити відхилення в різних напрямках простору ознак [3].

Геометрично поверхні однакової адаптивної відстані мають еліпсоїдальну форму. У двовимірному просторі це еліпси, а не кола, як у класичному FCM з евклідовою метрикою. Напрямки головних осей еліпса визначаються власними векторами коваріаційної матриці, а співвідношення

довжин осей – величинами розсіювання даних у відповідних напрямках. Тому GK здатний описувати витягнуті кластери, кластери з різною шириною уздовж окремих ознак та кластери, повернуті відносно координатних осей.

Наприклад, у двовимірному просторі об'єкти одного кластера можуть утворювати витягнутий діагональний еліпс. Для такого кластера переміщення вздовж напрямку великої осі є природним і не повинно суттєво зменшувати ступінь належності точки.

Водночас значне відхилення у напрямку, малої осі, може означати, що точка не відповідає структурі кластера. Евклідова відстань не враховує цієї різниці, оскільки однаково оцінює зміни координат незалежно від напрямку. Адаптивна метрика GK, навпаки, може надавати менший штраф за зміщення вздовж головної осі кластера та більший за відхилення від неї.

Саме врахування залежностей між ознаками є однією з головних переваг GK. У FCM кожна ознака впливає на евклідову відстань окремо. Якщо між двома ознаками існує виражена залежність, стандартна евклідова метрика не використовує цю інформацію. У GK залежність відображається позадіагональними елементами коваріаційної матриці. Це дає змогу врахувати природну орієнтацію групи об'єктів у просторі ознак та коректніше оцінювати їхню близькість до кластера.

Концептуально FCM і GK мають спільну основу. Обидва методи використовують матрицю нечітких належностей, параметр нечіткості та ітераційне чергування уточнення належностей і центрів. В обох випадках кожному об'єкту призначаються значення належності до всіх кластерів, а сума цих значень дорівнює одиниці.

Проте FCM використовує одну глобальну евклідову метрику, що відповідає припущенню про близьку до сферичної форму кластерів у нормалізованому просторі ознак. GK замінює це припущення набором локальних метрик: кожний кластер має власний центр, власну нечітку коваріаційну матрицю та власну геометрію відстаней.

Завдяки цьому GK є більш придатним для аналізу даних із анізотропними кластерами, тобто кластерами з неоднаковим розсіюванням у різних напрямках. Він також може краще працювати, коли кластери мають різну орієнтацію або ознаки всередині кластерів є залежними. Водночас це збільшує складність алгоритму. На кожній ітерації необхідно не лише оновити центри та належності, а й обчислити коваріаційні матриці, їхні визначники та обернені матриці для всіх кластерів.

Додатковою практичною проблемою є можливість виродження коваріаційної матриці. Якщо кластер тимчасово містить недостатньо точок або його ознаки майже лінійно залежні, матриця F_j може бути необерненою або погано обумовленою. У такому випадку обчислення F_j^{-1} стає нестійким. Тому в практичних реалізаціях застосовують перевірку матриць та регуляризацію, наприклад додавання малого додатного значення до діагональних елементів. Крім того, як і у FCM, результат GK залежить від масштабу ознак, тому перед кластеризацією дані доцільно нормалізувати або стандартизувати.

Класичний алгоритм GK має пакетний характер. На кожній ітерації він потребує доступу до всього набору даних для перерахунку матриці нечітких належностей, центрів і коваріаційних матриць. Це означає, що стандартний варіант не призначений для послідовного надходження потенційно необмеженого потоку, обмеженої пам'яті та необхідності одразу приймати рішення для нової точки. Також кількість кластерів у класичному GK задається до початку роботи та не змінюється автоматично [3].

Таким чином, алгоритм Gustafson-Kessel зберігає головну перевагу нечіткої кластеризації – подання невизначеної належності граничних об'єктів і розширює FCM засобами адаптації геометрії кластерів. Його застосування є обґрунтованим, коли природні групи мають еліпсоїдальну форму, різну орієнтацію та залежності між ознаками. Водночас для задачі адаптивної нечіткої кластеризації потоків даних класичний GK слід розглядати як базовий алгоритм, що потребує подальшого розвитку шляхом

інкрементального оновлення параметрів, контролю віддалених точок і створення нового кластера після накопичення щільної групи кандидатів.

1.3 Постановка задачі

Метою роботи є розроблення та аналіз адаптивного методу нечіткої кластеризації потокових даних на основі алгоритму Gustafson-Kessel, що функціонує за умов апіорної та поточної невизначеності.

Об'єктом дослідження є процес нечіткої кластеризації багатовимірних потокових даних, що надходять на оброблення послідовно та можуть змінювати свої властивості з часом.

Предметом дослідження є алгоритми нечіткої кластеризації потокових даних на основі алгоритму Gustafson-Kessel та методи їх адаптації до невідомої або змінної структури кластерів, наявності шуму й аномальних спостережень.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- провести аналіз існуючих методів кластеризації даних;
- розглянути особливості нечіткої кластеризації та методу Fuzzy C-Means;
- дослідити алгоритм Gustafson-Kessel, провести його порівняльний аналіз із методом Fuzzy C-Means та обґрунтувати доцільність його використання як основи розроблюваного методу;
- проаналізувати специфіку оброблення потоків даних в онлайн-режимі;
- визначити основні прояви апіорної та поточної невизначеності у задачах кластеризації потокових даних;
- сформулювати вимоги до адаптивного методу нечіткої кластеризації потокових даних;
- розробити способи адаптації алгоритму Gustafson-Kessel до послідовного надходження даних і зміни структури потоку;

- розробити алгоритмічну модель адаптивної нечіткої кластеризації потокових даних на основі алгоритму Gustafson-Kessel;
- виконати програмну реалізацію розробленої моделі;
- провести експериментальну перевірку роботи методу та аналіз отриманих результатів.

2 РОЗРОБКА МЕТОДА НЕЧІТКОЇ КЛАСТЕРИЗАЦІЇ ДАНИХ ЗА УМОВ АПРІОРНОЇ ТА ПОТОЧНОЇ НЕВИЗНАЧЕНОСТІ

2.1 Вимоги до розроблюваного методу та обґрунтування вибору базового алгоритму

Передусім метод повинен виконувати нечітку кластеризацію багатовимірного потоку, у якому спостереження надходять по одному. Після появи чергового вектора ознак модель має визначати його зв'язок із наявними кластерами без очікування завершення потоку та без використання майбутніх спостережень. Результатом є вектор нечітких належностей, що відображає близькість точки до всіх активних кластерів, а також її поточна мітка, потрібна для візуалізації та оцінювання результатів. Водночас оброблення нової точки не повинно вимагати повторного запуску алгоритму для всієї накопиченої вибірки.

Важливою вимогою є обмеженість обчислювальних витрат і пам'яті. Час оброблення однієї точки має залежати від розмірності простору ознак, кількості активних кластерів і фіксованого обсягу службового буфера, а не від довжини всієї історії потоку. Тому в методі мають зберігатися дані прогнозованого об'єму. За фіксованих параметрів поточної моделі це забезпечує обмежений час реакції та контрольований обсяг пам'яті. Із ростом кількості кластерів витрати на одну точку закономірно збільшуються, однак повне переобчислення для всіх попередніх даних не виконується.

Окремою вимогою є адаптивна нормалізація. У потоковій задачі середні значення та розсіювання ознак не можуть бути визначені до початку роботи за повною вибіркою, тому їх необхідно уточнювати послідовно. Після оновлення параметрів нормалізації центри кластерів, матриці форми й інші параметри моделі мають бути узгоджені з новою системою

координат. Це необхідно для коректного порівняння нової точки з параметрами, оціненими на попередньому відрізку потоку.

Метод також повинен враховувати апріорну та поточну невизначеність. Кількість природних груп на початку роботи невідома, тому алгоритм має знаходити утворення нового кластеру даними та виділяти його, зберігаючи стійкість при обробці аномальних даних. Окрім цього, У межах роботи передбачається додавання нових кластерів; їх злиття, поділ і видалення не розглядаються.

Як основу доцільно використати класичний алгоритм, а не готовий комплексний метод потокової кластеризації. Класичний алгоритм має чітко визначені математичні параметри, послідовність обчислень і обмеження. Тому можна окремо обґрунтувати вплив кожного доповнення. Такий підхід також дозволяє порівнювати розроблену модель із базовою пакетною схемою та визначати, яке саме її обмеження усуває кожна модифікація.

Серед класичних нечітких методів базою обрано Gustafson-Kessel, а не Fuzzy C-Means. FCM використовує одну спільну евклідову метрику, тому найкраще відповідає кластерам, близьким до сферичних у нормалізованому просторі. GK дозволяє розрізняти витягнуті, еліпсоїдальні й по-різному орієнтовані структури, а також враховувати залежності між ознаками. Вибір GK створює запас гнучкості для подальшого аналізу даних зі складнішою геометрією, не обмежуючи метод припущенням про однакову форму всіх груп [2], [3].

Крім того, параметри GK утворюють змістовний компактний опис поточного стану кластера: центр, накопичена нечітка вага, оцінка коваріації та матриця адаптивної метрики. У класичному пакетному GK ці характеристики обчислюються за всією вибіркою, проте саме вони є основою для рекурсивного оновлення без постійного зберігання старих точок. Отже, розроблюваний метод розглядається як потокове розширення GK, яке поєднує нечітке подання граничних об'єктів, адаптацію геометрії кластерів, інкрементальне оновлення, адаптивну нормалізацію,

автоматичне збільшення кількості кластерів і контрольоване оброблення аномальних спостережень.

2.2 Принцип потокового оновлення параметрів моделі

Під час переходу від пакетного алгоритму Gustafson-Kessel до потокового режиму основна зміна полягає у способі зберігання та оновлення інформації про кластери. У класичному пакетному варіанті алгоритм має доступ до всієї вибірки та на кожній ітерації повторно обчислює ступені належності, центри й коваріаційні матриці за всіма об'єктами. У потоковій моделі така схема є непридатною, оскільки кількість уже отриманих спостережень постійно зростає. Тому замість зберігання всіх попередніх точок модель підтримує лише компактний набір параметрів, достатній для опису поточного стану кожного кластера [3], [14], [15].

На цьому етапі поки вважається, що початкову кількість кластерів уже визначено під час ініціалізації моделі та вона тимчасово є сталою. Для кожного кластера C_j зберігаються чотири основні групи параметрів. Першим параметром є центр кластера v_j , який визначає його поточне положення у просторі ознак. Другим параметром є нечітка коваріаційна матриця F_j . Вона містить інформацію про розсіювання точок навколо центра та залежності між окремими ознаками. Третім параметром є матриця адаптивної метрики A_j , що обчислюється на основі коваріаційної матриці та використовується для визначення відстані від нової точки до конкретного кластера. Четвертим параметром є накопичена нечітка вага s_j , яка відображає сумарний внесок уже прийнятих точок у формування параметрів відповідного кластера. І так для кожного кластера.

Сама історія точок $x_1, x_2, \dots, x_{t-1}$ для оновлення моделі не зберігається. Це є принциповою відмінністю потокового підходу: нова інформація враховується через зміну параметрів кластерів, а не через повторне оброблення всіх раніше отриманих об'єктів.

Після надходження нового спостереження x_t алгоритм спочатку обчислює його адаптивну відстань до центрів усіх наявних кластерів. Для кластера j використовується квадрат відстані:

$$d_{tj}^2 = (x_t - v_j)^T A_j (x_t - v_j). \quad (2.1)$$

На відміну від звичайної евклідової відстані, ця величина залежить від параметрів конкретного кластера. Тому одна й та сама точка може бути близькою до витягнутого кластера вздовж його головної осі, але водночас далекою від іншого кластера з іншою формою або орієнтацією.

На основі отриманих відстаней визначаються нечіткі ступені належності нової точки до всіх кластерів. Чим меншою є відстань до кластера, тим більшим буде відповідний ступінь належності. При цьому для точки виконується умова нормування:

$$\sum_{j=1}^K u_{tj} = 1. \quad (2.2)$$

Отримані значення не означають, що точка оновлює лише один кластер. Навпаки, вона впливає на параметри всіх кластерів, проте сила цього впливу визначається її нечіткою належністю. Для цього обчислюється нечітка вага точки:

$$w_{tj} = u_{tj}^m, \quad (2.3)$$

де m , що обов'язково більший за 1, це параметр нечіткості. Чим більшою є належність точки до кластера, тим більшим буде її внесок у зміну параметрів цього кластера. А накопичена вага кластера оновлюється рекурсивно:

$$s_j(t) = s_j(t-1) + w_{tj}. \quad (2.4)$$

Далі визначається коефіцієнт оновлення:

$$\alpha_{tj} = \frac{w_{tj}}{s_j(t)}. \quad (2.5)$$

Цей коефіцієнт показує, наскільки суттєво нова точка повинна змістити центр кластера [2]. Нове положення центра визначається за формулою:

$$v_j(t) = v_j(t-1) + \alpha_{tj}(x_t - v_j(t-1)). \quad (2.6)$$

Отже, центр зміщується у напрямку нової точки. Якщо точка має високу належність до кластера, зміщення буде більшим. Якщо її належність низька, центр практично не зміниться.

Подібним чином оновлюється нечітка коваріаційна матриця. Вона уточнюється на основі попередньої матриці та відхилення нової точки від попереднього центра кластера. У результаті F_j поступово накопичує інформацію про форму кластера без необхідності зберігати всі точки, що раніше до нього надходили. Після оновлення F_j заново обчислюється матриця метрики A_j , яка використовуватиметься під час оброблення наступного спостереження.

Важливо, що відстані та ступені належності для точки обчислюються за одним і тим самим старим станом моделі. Лише після цього параметри всіх кластерів оновлюються одночасно. Такий порядок не допускає ситуації, у якій параметри одного кластера вже змінено поточною точкою, а параметри іншого ще належать до попереднього стану.

Отже, для оброблення потоку модель зберігає не попередні об'єкти, а компактний опис кожного кластера: його центр, форму, метрику та

накопичену вагу. Час оброблення однієї точки залежить від кількості кластерів і кількості ознак, але не від загальної кількості вже отриманих спостережень. Саме це створює основу для подальшого доповнення базової моделі потоковою нормалізацією, механізмами виявлення аномалій та адаптацією кількості кластерів.

2.3 Формування початкової моделі на стартовому фрагменті потоку

Рекурсивне оновлення параметрів кластера можливе лише тоді, коли модель уже має початкові значення центрів, матриць форми та накопичених ваг. На початку потоку таких параметрів ще не існує, тому перші спостереження використовуються для формування початкової моделі. Для цього виділяється обмежений стартовий фрагмент потоку, або warm-up буфер, що містить перші N_0 векторів ознак. Його розмір задається наперед і не залежить від загальної довжини потоку.

Використання стартового фрагмента не означає переходу до пакетної обробки всього набору даних. Алгоритм застосовує лише ці перші N_0 об'єктів, необхідних для створення початкового стану моделі. Майбутні точки потоку під час ініціалізації не використовуються. Це дозволяє сформувати мінімально необхідне початкове уявлення про структуру даних і після завершення ініціалізації перейти до послідовного опрацювання кожного нового об'єкта.

На першому етапі зі стартового фрагмента обираються K (вже відомих, або початкових кластерів) початкових прототипів. У реалізованій моделі вони можуть бути вибрані випадково лише серед точок warm-up буфера або задані як перші K об'єктів потоку. Обрані прототипи утворюють початкові центри кластерів v_j . Для кожного кластера також задається початкова коваріаційна матриця у вигляді одиничної матриці, помноженої на встановлений масштабний коефіцієнт. Така матриця відповідає

нейтральному припущенню про початкову форму кластера та гарантує можливість обчислення адаптивної метрики.

Кожному початковому кластеру також надається ненульова накопичена вага s_j . Це запобігає ситуації, коли перша оброблена точка повністю визначає положення центра або форму кластера. Після створення початкових параметрів усі точки стартового фрагмента один раз послідовно проходять через рекурсивну процедуру GK. Для кожної точки обчислюються відстані до поточних кластерів, її нечіткі належності та внесок у зміну центрів, коваріаційних матриць і накопичених ваг.

Отже, після оброблення стартового фрагмента модель містить початково адаптовані параметри кластерів, сформовані лише за даними, доступними на початку потоку. Надалі кожне нове спостереження обробляється вже в потоковому режимі: воно порівнюється з поточними параметрами моделі та рекурсивно уточнює їх без повторного аналізу всіх попередніх об'єктів.

2.4 Потокова нормалізація та переведення параметрів кластерів у нову систему координат

Для методів кластеризації, що використовують відстані між об'єктами та центрами кластерів, масштаб ознак безпосередньо впливає на результат. Якщо окремі ознаки мають різні одиниці вимірювання або суттєво відрізняються за діапазоном значень, ознака з більшими числовими значеннями може непропорційно впливати на відстань і, відповідно, на ступені належності об'єктів до кластерів. Тому перед виконанням кластеризації дані доцільно приводити до порівнюваного масштабу.

У статичній задачі параметри нормалізації можуть бути визначені за всією вибіркою до початку роботи алгоритму. Наприклад, для кожної ознаки обчислюють середнє значення та стандартне відхилення, після чого кожний об'єкт переводять у стандартизовані координати. У потоковій задачі

повний набір даних наперед невідомий, тому глобальні параметри нормалізації обчислити неможливо. Середнє значення й розсіювання ознак необхідно уточнювати послідовно після надходження нових спостережень.

У розробленій моделі використовується потокова z-нормалізація. Для k -ї ознаки нового об'єкта x_t її стандартизоване значення визначається за формулою:

$$z_{tk} = \frac{x_{tk} - \mu_k(t)}{\sigma_k(t)}, \quad (2.7)$$

де $\mu_k(t)$ це поточне середнє значення ознаки після врахування t прийнятих спостережень;

$\sigma_k(t)$ – поточне стандартне відхилення цієї ознаки.

У векторному вигляді новий об'єкт переходить з початкового простору ознак у поточний стандартизований простір, у якому виконується обчислення відстаней і рекурсивне оновлення параметрів GK.

Для зберігання параметрів нормалізації не потрібно накопичувати всі попередні точки. Для кожної ознаки модель зберігає лише кількість прийнятих спостережень t , поточне середнє і величину $M_2(t)$, що є сумою квадратів відхилень від поточного середнього. Після надходження нового об'єкта ці значення уточнюються рекурсивно. Спочатку обчислюється різниця між новим значенням і попереднім середнім [10]:

$$\delta_t = x_t - \mu(t - 1). \quad (2.8)$$

Далі середнє оновлюється за формулою [10]:

$$\mu(t) = \mu(t - 1) + \frac{\delta_t}{t}. \quad (2.9)$$

Після уточнення середнього обчислюється нове значення накопиченої величини $M_2(t)$, а стандартне відхилення визначається як [10]:

$$\sigma(t) = \sqrt{\frac{M_2(t)}{t}}. \quad (2.10)$$

Така схема дозволяє послідовно оцінювати середні значення й дисперсії ознак за сталої кількості збережених параметрів. Для початкових точок потоку та ознак із майже нульовим розсіюванням стандартне відхилення може бути близьким до нуля. Щоб уникнути ділення на нуль, у таких випадках у реалізації використовується умовний масштаб, рівний одиниці.

Потокова нормалізація створює особливу проблему для кластерної моделі. Після прийняття нового об'єкта значення поточного середнього значення ознак і поточного стандартного їх відхилення змінюються. Отже, змінюється й сама система стандартизованих координат. Центри кластерів і коваріаційні матриці, обчислені до оновлення нормалізації, не можна безпосередньо порівнювати з новою точкою, переведеною вже за новими параметрами. Формально вони описують ті самі об'єкти, але в різних системах координат. Для усунення цієї невідповідності всі параметри GK переводяться зі старої системи координат у нову. Нехай для попереднього стану моделі використовувалися параметри μ_{old} і σ_{old} , а після прийняття точки отримано μ_{new} і σ_{new} . Тоді зв'язок між старими та новими стандартизованими координатами має вигляд:

$$z_{new} = Bz_{old} + b, \quad (2.11)$$

де

$$B = diag\left(\frac{\sigma_{old}}{\sigma_{new}}\right), \quad (2.12)$$

$$b = \frac{\mu_{old} - \mu_{new}}{\sigma_{new}}. \quad (2.13)$$

Матриця B описує зміну масштабу кожної координати, а вектор b описує її зміщення. Оскільки центр кластера є точкою у просторі ознак, він переноситься за правилом:

$$v_j^{new} = Bv_j^{old} + b. \quad (2.14)$$

Коваріаційна матриця описує не положення, а розсіювання об'єктів навколо центра. Тому для неї враховується лише зміна масштабу координат:

$$F_j^{new} = BF_j^{old}B^T. \quad (2.15)$$

Після перетворення коваріаційних матриць для кожного кластера повторно обчислюється матриця адаптивної метрики A_j . Лише після цього нова точка, уже подана в актуальних стандартизованих координатах, використовується для визначення нечітких належностей і рекурсивного оновлення центрів, коваріаційних матриць та накопичених ваг.

Отже, порядок оброблення прийнятої точки є таким: спочатку уточнюються параметри потокової нормалізації; далі всі параметри кластерів переводяться в нову систему координат; після цього нормалізована точка порівнюється з актуальними кластерами та оновлює їх. Така послідовність забезпечує узгодженість центрів, матриць форми та нових спостережень у межах однієї системи координат.

2.5 Адаптація кількості кластерів і оброблення потенційно аномальних спостережень

У класичному алгоритмі Gustafson-Kessel кількість кластерів задається до початку роботи та залишається незмінною протягом усього

процесу кластеризації. Для потокових даних таке припущення є обмежувальним, оскільки на початковому фрагменті потоку можуть бути представлені не всі природні групи спостережень. Тому в розробленій моделі параметр K інтерпретується не як остаточна кількість кластерів, а лише як їх початкове число, необхідне для формування стартової моделі. Надалі кількість кластерів може збільшуватися внаслідок виявлення нової стійкої групи віддалених точок [3].

Після завершення стартової ініціалізації кожна нова точка спочатку перевіряється на сумісність із поточними кластерами. Якщо вона розташована в межах допустимої області найближчого кластера, точка приймається як звичайне спостереження потоку та використовується для оновлення параметрів моделі. Якщо ж точка є надто віддаленою від усіх існуючих кластерів, її не можна відразу оголосити аномалією або новим кластером. Одиначне віддалене спостереження може бути випадковим шумом, помилкою вимірювання або першою точкою раніше невідомої групи. Для розрізнення цих випадків використовується буфер кандидатів.

Буфер кандидатів є обмеженою за розміром структурою даних, у якій тимчасово зберігаються точки, не сумісні з наявними кластерами. Для кожної такої точки зберігається її початковий вектор ознак, порядковий номер у потоці та порядок надходження. Поки точка перебуває у буфері, вона не впливає на центри кластерів, коваріаційні матриці, накопичені ваги та параметри потокової нормалізації. Це запобігає ситуації, у якій одиничний викид штучно зміщує або деформує один із наявних кластерів.

Буфер має фіксовану максимальну місткість. Якщо до заповненого буфера надходить новий кандидат, найстаріша точка витісняється з нього та отримує остаточний статус аномального спостереження. Після цього вона більше не бере участі в оновленні моделі, створенні кластерів або оцінюванні якості кластеризації. Такий механізм обмежує обсяг пам'яті, необхідний для зберігання віддалених точок, і водночас надає кожному

кандидату певний проміжок часу для підтвердження або спростування його належності до нової структури.

Після кожної зміни стану моделі виконується повторна перевірка всіх точок, що залишаються у буфері. Зміна моделі може відбутися після прийняття звичайної точки або після створення нового кластера. Оскільки внаслідок такого оновлення можуть змінитися центри, матриці метрик, радіуси кластерів і параметри нормалізації, точка, яка раніше була віддаленою, може стати сумісною з одним із наявних кластерів.

Перевірка кандидатів виконується у порядку їх надходження. Якщо виявляється точка, яка в поточному стані моделі проходить перевірку сумісності з наявним кластером, вона вилучається з буфера та обробляється як наступне звичайне спостереження потоку. Для неї обчислюються нечіткі належності, після чого оновлюються параметри GK і потокової нормалізації. Після такого оновлення повторна перевірка буфера починається знову, оскільки стан моделі вже змінився. Такий порядок дозволяє поступово повернути до моделі ті точки, які перестали бути віддаленими внаслідок адаптації кластерів.

Якщо жодна точка буфера не сумісна з поточними кластерами, виконується перевірка можливого створення нового кластера. Для цього точки буфера подаються в актуальних нормалізованих координатах. Усередині буфера шукається щільна група за DBSCAN-подібним принципом. Дві точки вважаються сусідніми, якщо їх евклідова відстань у поточному просторі стандартизованих ознак не перевищує заданого радіуса. Точка вважається ядровою, якщо кількість її сусідів, включно з нею самою, не менша за встановлений поріг q . Нова група формується як компонента зв'язності ядрових і суміжних із ними точок.

Для створення нового кластера недостатньо лише наявності декількох близьких точок. Розмір знайденої щільної групи повинен бути не меншим за мінімальне значення $N_{\min}$. Така умова зменшує ймовірність того, що випадкове локальне накопичення шуму буде помилково інтерпретоване як

новий кластер. Якщо у буфері одночасно присутні декілька груп, які задовольняють умову щільності, першою обробляється група, найстаріша точка якої надійшла раніше.

Після підтвердження групи для неї створюється новий кластер. Його початковий центр визначається як середнє значення точок групи, коваріаційна матриця – як оцінка їхнього розсіювання, а матриця адаптивної метрики обчислюється на основі цієї коваріації. Накопичена вага нового кластера дорівнює кількості точок, що утворили групу. Усі точки групи отримують мітку нового кластера та вилучаються з буфера. Щоб не врахувати їх двічі, вони не проходять повторне звичайне нечітке оновлення GK: їхні центр, коваріація й вага вже були обчислені безпосередньо за цією групою.

Після створення нового кластера його точки послідовно враховуються у статистиках потокової нормалізації, а всі параметри GK переносяться до оновленої системи координат. Далі буфер знову перевіряється, оскільки поява нового кластера може зробити сумісними інші віддалені точки. У межах розробленого методу кількість кластерів може лише збільшуватися. Операції злиття, поділу та видалення кластерів не передбачені.

Отже, буфер кандидатів виконує дві функції: тимчасово ізолює віддалені точки від параметрів наявних кластерів і накопичує інформацію, достатню для розрізнення одиничних аномалій та нових стійких груп. Точка може завершити оброблення трьома способами: бути прийнятою до існуючого кластера після повторної перевірки, стати частиною нового кластера або отримати статус аномалії внаслідок витіснення з переповненого буфера.

Детальніше, процедура прийняття рішення щодо віддаленої точки складається з двох послідовних етапів: перевірки її відповідності поточним кластерам і, за потреби, аналізу структури всіх накопичених кандидатів.

Спочатку нова точка перевіряється без зміни параметрів моделі. Вона переводиться до поточної системи нормалізованих координат, сформованої

за вже прийнятими спостереженнями. Далі для цієї точки обчислюються квадрати адаптивних відстаней до центрів усіх наявних кластерів.

Така відстань враховує не лише положення центра, а й форму кластера. Наприклад, для витягнутого кластера точка може бути досить віддаленою від центра у звичайному евклідовому сенсі, але все ще відповідати його структурі, якщо вона лежить уздовж напрямку найбільшого розсіювання. А найзручніше для нас те, що враховується саме в обчисленні відстані, а отже ми можемо спокійно порівнювати між собою відстані до різних за формою кластерів. Із усіх обчислених відстаней вибирається найменша. Вона визначає найближчий кластер, однак сама по собі не означає прийняття точки. Для кожного кластера підтримується поточний допустимий радіус r_q^2 . Він оцінюється за останніми опорними точками, які раніше були прийняті до цього кластера. Ці точки зберігаються в обмеженому буфері у вихідних координатах, після чого за потреби переводяться в актуальну систему нормалізації. Для них заново обчислюються GK-відстані до центра відповідного кластера. Радіус не визначається як найбільша з таких відстаней, оскільки одна нетипова точка могла б непропорційно розширити допустимі межі кластера. Замість цього використовується заданий квантиль розподілу відстаней. Наприклад, квантиль рівня 0,95 означає, що межа визначається так, щоб 95 % останніх опорних точок мали меншу або рівну відстань. Для малих кластерів також встановлюється мінімально допустиме значення радіуса, щоб межа не стала майже нульовою. Нова точка визнається сумісною з найближчим кластером C_q , якщо виконується умова

$$d_{tq}^2 \leq \gamma r_q^2, \quad (2.16)$$

де γ – коефіцієнт розширення межі. Він дозволяє приймати точки, що лежать трохи далі від типової області кластера, але ще не є достатньо віддаленими для переведення до буфера кандидатів. Якщо умова

виконується, точка проходить звичайне нечітке оновлення: включається до потокової нормалізації, отримує ступені належності до кластерів і впливає на їхні центри, коваріаційні матриці та накопичені ваги.

Якщо умова не виконується, точка отримує тимчасовий статус кандидата. Це не означає, що вона вже є аномалією. Вона лише не відповідає поточній моделі на момент надходження. Надалі модель може змінитися, тому така точка здатна стати сумісною з одним із існуючих кластерів або увійти до складу нової групи.

Пошук нової групи здійснюється лише серед точок буфера. Усі вони переводяться в поточні стандартизовані координати, після чого між кожною парою кандидатів обчислюється евклідова відстань. Дві точки вважаються сусідніми, якщо вона не перевищує заданого радіуса. Для кожної точки визначається кількість таких сусідів, включно з нею самою. Якщо ця кількість не менша за порогове значення, точка вважається ядровою.

Далі алгоритм починає з ядрової точки та послідовно додає до групи її сусідів. Якщо серед сусідів є інші ядрові точки, пошук продовжується вже через них. Таким способом формується компонента щільно пов'язаних точок. До неї можуть входити як ядрові точки, так і граничні точки, що розташовані поруч із ядром групи, але самі не мають достатньої кількості сусідів. Саме тому процедура є DBSCAN-подібною: вона використовує поняття околу, ядрових точок і щільнісної зв'язності, проте застосовується лише до обмеженого буфера кандидатів, а не до всього потоку.

Побудована компонента інтерпретується як новий кластер лише тоді, коли її розмір не менший за мінімально встановлений. Якщо ця умова не виконується, точки залишаються у буфері та очікують наступних змін моделі або нових подібних кандидатів. Якщо умова виконується, для групи створюються центр, коваріаційна матриця, адаптивна метрика та початкова вага нового кластера.

Якщо потік завершується до переповнення буфера, кандидати, що залишилися, не отримують статус аномалій автоматично. Вони зберігають

проміжний статус непідтверджених віддалених спостережень, оскільки для остаточного рішення про їхню природу могло не вистачити наступних точок потоку.

2.6 Узагальнена схема роботи розробленого методу

Розроблений метод є потоковим розширенням алгоритму Gustafson-Kessel, у якому класична пакетна процедура уточнення параметрів замінюється послідовним обробленням окремих спостережень. Робота моделі складається з етапу формування початкового стану та подальшого циклу оброблення точок потоку.

На першому етапі з потоку виділяється обмежений стартовий фрагмент, або warm-up-буфер. Його призначення полягає у формуванні початкової моделі, оскільки до надходження перших даних неможливо визначити центри кластерів, їхню форму та параметри нормалізації. За даними цього фрагмента обчислюються початкові статистики потокової нормалізації, обираються початкові прототипи та створюються K стартових кластерів. Для кожного з них задаються центр, коваріаційна матриця, адаптивна матриця метрики та накопичена нечітка вага. Значення K розглядається лише як початкове число кластерів, а не як остаточний параметр моделі.

Після завершення ініціалізації модель переходить до послідовного оброблення кожної наступної точки x_t . Спочатку вона перевіряється на сумісність із поточними активними кластерами. Для цього точка переводиться в актуальну систему нормалізованих координат без її включення до статистик нормалізації. Далі обчислюються адаптивні відстані Gustafson-Kessel до центрів усіх кластерів, визначається найближчий кластер і порівнюється отримана відстань із допустимим радіусом цього кластера.

Якщо точка є сумісною з одним із наявних кластерів, вона приймається як звичайне спостереження потоку. Її оброблення відбувається у встановленій послідовності. Спочатку точка включається до статистик потокової нормалізації, унаслідок чого можуть змінитися середні значення та масштаби ознак. Після цього центри та коваріаційні матриці всіх активних кластерів переводяться до нової системи стандартизованих координат.

Далі для нормалізованої точки обчислюються нечіткі належності до кластерів, а параметри GK рекурсивно уточнюються: оновлюються центри, коваріаційні матриці, матриці адаптивних метрик і накопичені ваги. Для точки також фіксуються її вектор нечітких належностей та поточна жорстка мітка, що відповідає кластеру з найбільшим ступенем належності.

Якщо нова точка не проходить перевірку сумісності, вона не оголошується аномалією відразу. Натомість вона додається до буфера кандидатів. Поки точка перебуває у цьому буфері, вона не впливає на статистику нормалізації, центри кластерів, коваріаційні матриці або накопичені ваги. Таким чином, потенційний викид не може спотворити вже сформовану структуру активних кластерів.

Буфер кандидатів має обмежену місткість. Якщо до заповненого буфера надходить новий кандидат, найстаріша точка витісняється та остаточно отримує статус аномального спостереження. Вона більше не бере участі в оновленні моделі та не використовується в оцінюванні якості кластеризації. Водночас кандидати, що залишаються у буфері, зберігають можливість бути прийнятими до моделі пізніше.

Після кожної зміни стану моделі виконується процедура повторної перевірки буфера. Насамперед точки-кандидати перевіряються на сумісність із поточними активними кластерами у порядку їх надходження. Якщо певна точка стала сумісною з кластером, вона вилучається з буфера та обробляється як звичайне наступне спостереження потоку. Після її

прийняття параметри моделі змінюються, тому весь буфер знову перевіряється з початку.

Лише якщо жоден кандидат не став сумісним із наявними кластерами, виконується пошук щільної групи у буфері. Він здійснюється за DBSCAN-подібним принципом у поточних стандартизованих координатах. Для кандидатів визначаються сусідні точки в межах заданого радіуса, виділяються ядрові точки з достатньою кількістю сусідів і формуються щільно зв'язані компоненти. Якщо знайдена група містить не менше мінімально встановленої кількості точок, вона інтерпретується як нова стійка структура даних.

Для підтвердженої групи створюється новий кластер. Його центр визначається за точками групи, коваріаційна матриця описує їхнє розсіювання, а адаптивна метрика формується на основі цієї матриці. Точки групи отримують мітку нового кластера та вилучаються з буфера. Вони не проходять повторне рекурсивне оновлення GK, оскільки вже були використані під час формування початкових параметрів нового кластера. Після цього їхні дані послідовно включаються до статистик потокової нормалізації, а всі параметри моделі переводяться до оновленої системи координат.

На момент завершення потоку результатом роботи методу є множина активних кластерів із поточними центрами, матрицями форми, метриками та накопиченими вагами. Для кожної прийнятої точки зберігаються зафіксована мітка кластера й вектор нечітких належностей.

Окремо зберігаються точки, остаточно визначені як аномальні внаслідок витіснення з переповненого буфера, а також кандидати, для яких до моменту завершення потоку не було достатньо даних для остаточного рішення.

Алгоритм наведено на рисунку 2.1.

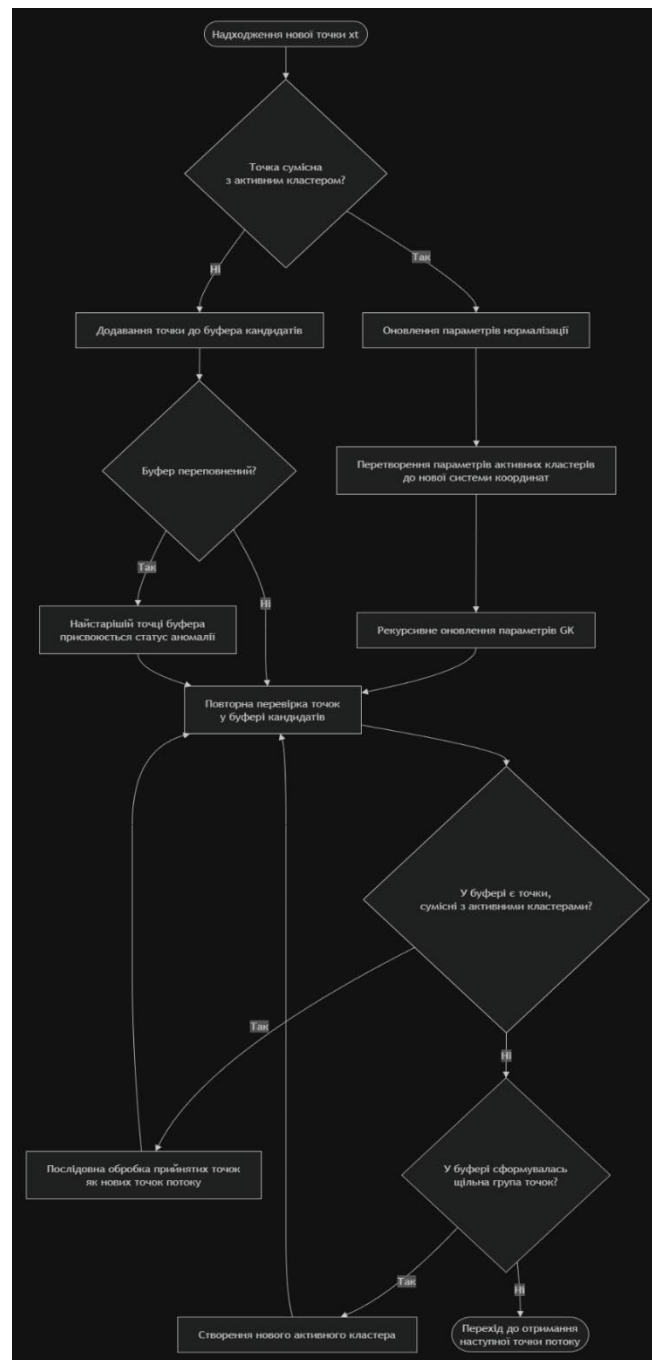


Рисунок 2.1 – Схема алгоритму

Отже, класичний пакетний алгоритм Gustafson-Kessel перетворено на потокову модель, що виконує нечітку кластеризацію без повторного оброблення всієї історії даних, підтримує адаптивну нормалізацію, допускає появу нових кластерів і відокремлює потенційні аномалії від стійких нових груп. Наступний розділ присвячено експериментальній перевірці цих властивостей та аналізу результатів роботи розробленого методу.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

Наступним етапом роботи є експериментальна перевірка якості розробленого методу. Спочатку доцільно порівняти класичні алгоритми Fuzzy C-Means та Gustafson-Kessel. Таке порівняння необхідне для практичного підтвердження вибору алгоритму Gustafson-Kessel як бази розроблюваного методу.

Після порівняння базових алгоритмів буде проведено послідовну перевірку впливу розроблених доповнень на роботу моделі. Спочатку оцінюватиметься рекурсивний варіант GK за фіксованої кількості кластерів. Далі до нього додається потокова обробка, потім онлайн нормалізація, після чого перевіряється вплив механізму буфера кандидатів, створення нових кластерів та фіксації аномальних спостережень. Такий порядок дозволяє окремо визначити, як кожна функція, введена для розв'язання проблем поточності, апріорної та поточної невизначеності, впливає на отримане розбиття даних.

Результати Fuzzy C-Means, Gustafson-Kessel і розробленої потокової моделі залежать від початкового стану алгоритму. Для FCM і класичного GK випадковість пов'язана з формуванням початкової матриці належностей або вибором початкових центрів. У потоковій версії GK випадковість впливає переважно на формування початкової моделі на warm-up-фрагменті потоку, зокрема на вибір початкових прототипів кластерів. Після завершення ініціалізації за фіксованого порядку надходження точок, незмінних параметрів моделі та одного значення random seed подальша обробка потоку є визначеною.

Отже, залежність від випадкової ініціалізації не є окремою проблемою потокової обробки даних. Вона є властивістю налаштування алгоритму, яка також характерна для пакетних методів кластеризації. Водночас її вплив не повинен ігноруватися під час експерименту. Для кожної конфігурації алгоритму буде виконано десять запусків із різними значеннями random

seed. При цьому порядок надходження точок потоку та всі інші параметри залишаються однаковими. За результатами запусків для кожного критерію визначатимуться середнє значення та стандартне відхилення. Це дозволить оцінити не лише середню якість розбиття, а й стабільність алгоритму щодо початкової ініціалізації.

Оцінювання результату нечіткої кластеризації не може базуватися лише на одному показнику. Частина критеріїв дозволяє порівняти отримане розбиття з відомими мітками класів, а інша частина оцінює внутрішню структуру кластерів і характер нечітких належностей. Тому в роботі використовуються чотири взаємодоповнювальні критерії: Adjusted Rand Index, Adjusted Concordance Index, нормована ентропія нечітких належностей та індекс Xie-Beni.

Першим критерієм є Adjusted Rand Index, або ARI. Він використовується для порівняння жорсткого розбиття, отриманого алгоритмом, із відомими мітками класів набору даних. Для його обчислення кожній точці призначається мітка кластера, що відповідає найбільшому значенню її нечіткої належності [11]:

$$\hat{y}_i = \operatorname{argmax}_j u_{ij}. \quad (3.1)$$

ARI оцінює, наскільки однаково алгоритм і еталонне розбиття визначають пари об'єктів як такі, що належать до одного або до різних кластерів. На відміну від звичайного індексу Ренда, цей показник враховує очікуваний рівень випадкового збігу. Значення, близьке до одиниці, відповідає майже повному збігу з еталонними мітками, значення поблизу нуля означає результат, близький до випадкового, а від'ємне значення свідчить про гіршу за випадкову відповідність. ARI обрано тому, що він не залежить від конкретних номерів кластерів і дозволяє безпосередньо оцінити, наскільки правильно алгоритм відтворює відому структуру синтетичного набору даних [11].

Разом із тим ARI використовує лише жорсткі мітки, отримані після вибору найбільшого значення належності. Через це два різні нечіткі розбиття можуть мати однакові жорсткі мітки та однакове значення ARI, хоча розподіл ступенів належності між кластерами буде відрізнятися. Для врахування саме нечіткої структури використовується Adjusted Concordance Index, або ACI.

ACI є розширенням ARI для нечітких розбиттів. На відміну від ARI, цей критерій не потребує перетворення матриці нечітких належностей у набір жорстких міток. Він порівнює ступені спільної належності об'єктів до кластерів у результаті алгоритму та в еталонному розбитті. Для наборів даних із відомими чіткими класами еталонне розбиття подається у вигляді матриці належностей, у якій об'єкт має значення 1 для свого правильного класу та 0 для всіх інших. Таким чином, ACI дозволяє оцінити не лише факт вибору найближчого кластера, а й те, наскільки розподіл нечітких належностей відповідає заданій структурі даних. Використання ACI є важливим саме для цієї роботи, оскільки розроблюваний метод не відмовляється від нечіткої природи GK, а зберігає вектор належностей для кожної прийнятої точки [13].

Третім показником є нормована ентропія нечітких належностей. Вона обчислюється за формулою [13]:

$$H_U = -\frac{1}{n \ln K} \sum_{i=1}^n \sum_{j=1}^K u_{ij} \ln u_{ij} . \quad (3.2)$$

Цей критерій характеризує ступінь невизначеності розподілу належностей між кластерами. Якщо для більшості об'єктів одна належність значно перевищує інші, значення ентропії буде близьким до нуля. Якщо ж точки мають близькі ступені належності до кількох кластерів, ентропія зростає та наближається до одиниці. Нормування на $\ln K$ дозволяє врахувати поточну кількість активних кластерів.

Нормована ентропія не використовується як самостійний показник правильності кластеризації. Мале значення може означати як добре виражену структуру даних, так і надто жорстке призначення точок до кластерів. Її призначення полягає у контролі того, як додавання потокової нормалізації, буфера кандидатів і механізму зміни кількості кластерів впливає на нечіткість отриманого розбиття. Для коректності цей показник обчислюється лише для точок, які були прийняті до активних кластерів, оскільки аномальні спостереження та непідтверджені кандидати не мають остаточного вектора належностей.

Четвертим критерієм є індекс Xіе-Benі. Він належить до внутрішніх критеріїв оцінювання нечіткої кластеризації та не потребує відомих міток класів. У загальному вигляді він враховує компактність кластерів і відстань між їхніми центрами [12]:

$$XB = \frac{\sum_{i=1}^n \sum_{j=1}^K u_{ij}^m \|z_i - v_j\|^2}{n \cdot \min_{p \neq q} \|v_p - v_q\|^2}. \quad (3.3)$$

У чисельнику враховується, наскільки точки розосереджені відносно центрів кластерів, з урахуванням їхніх нечітких належностей. У знаменнику визначається найменша відстань між центрами різних кластерів. Тому мале значення Xіе-Benі відповідає ситуації, коли кластери є компактними та водночас достатньо віддаленими один від одного [12].

Індекс Xіе-Benі обрано як доповнення до ARI та ACI. ARI й ACI показують відповідність результату відомих міткам класів, а Xіе-Benі оцінює внутрішню структуру отриманого розбиття. Для порівняння різних алгоритмів значення Xіе-Benі обчислюється в одному й тому самому нормалізованому просторі ознак. Водночас цей індекс не буде використовуватися як єдиний доказ переваги GK над FCM для витягнутих кластерів, оскільки його значення залежить від правильності сформованих класів.

Таким чином, ARI та ACI дозволяють оцінити відповідність результату відомій структурі даних у жорсткому та нечіткому вигляді відповідно. Нормована ентропія характеризує ступінь невизначеності нечітких належностей, а індекс Xie-Beni оцінює внутрішню компактність і розділеність кластерів. Спільне використання цих критеріїв дозволяє оцінювати не лише кінцеву правильність розбиття, а й характер змін, що виникають після додавання до базового алгоритму GK механізмів потокової обробки, адаптивної нормалізації, виявлення нових кластерів та відокремлення аномальних спостережень.

Додатково для оцінювання однорідності сформованих кластерів використовується показник чистоти кластерів, або *purity*. Він визначає, яка частка об'єктів у кожному отриманому кластері належить до найбільш представленого в ньому еталонного класу. Для його обчислення кожній точці, як і під час розрахунку ARI, призначається жорстка мітка кластера, що відповідає найбільшому значенню нечіткої належності. Далі для кожного отриманого кластера визначається клас, об'єкти якого представлені в ньому найбільше. Її значення може штучно зростати зі збільшенням кількості кластерів, оскільки поділ одного реального класу на декілька окремих кластерів не обов'язково зменшує *purity*. Це особливо важливо для розробленого методу, у якому кількість активних кластерів може збільшуватися під час оброблення потоку. Тому показник чистоти інтерпретується лише разом з ARI, ACI, нормованою ентропією нечітких належностей та індексом Xie-Beni. Але становиться більш самостійним коли відома правильна кількість кластерів.

Для первинної перевірки доцільності вибору алгоритму Gustafson-Kessel як основи розроблюваного методу спочатку було виконано порівняння класичних алгоритмів Fuzzy C-Means та Gustafson-Kessel. Перевірка проводилася на синтетичному наборі даних Synth 2, що містить витягнуті кластери, а також на класичному наборі даних Iris. Для кожного алгоритму було виконано десять запусків зі значеннями *random seed* від 1

до 10. У таблицях в додатку А наведено середні значення критеріїв і вибіркові стандартні відхилення. Синтетичний набір Synth 2 використано для перевірки поведінки алгоритмів у ситуації, коли природні групи мають витягнуту форму та різну орієнтацію, що спричиняє їх накладення. Саме для таких даних відмінність між спільною евклідовою метрикою FCM та окремими адаптивними метриками GK має проявлятися найбільш помітно.

На рисунку 3.1 видно, що Fuzzy C-Means не повністю відтворює структуру витягнутих груп. Алгоритм використовує одну евклідову метрику для всіх кластерів, тому не враховує, що відхилення точки вздовж головної осі витягнутого кластера є природним, а відхилення в поперечному напрямку може означати належність до іншої групи. Унаслідок цього частина точок, що належать до різних витягнутих структур, отримує однакову мітку кластера, а межа між сформованими кластерами не відповідає заданій геометрії даних.

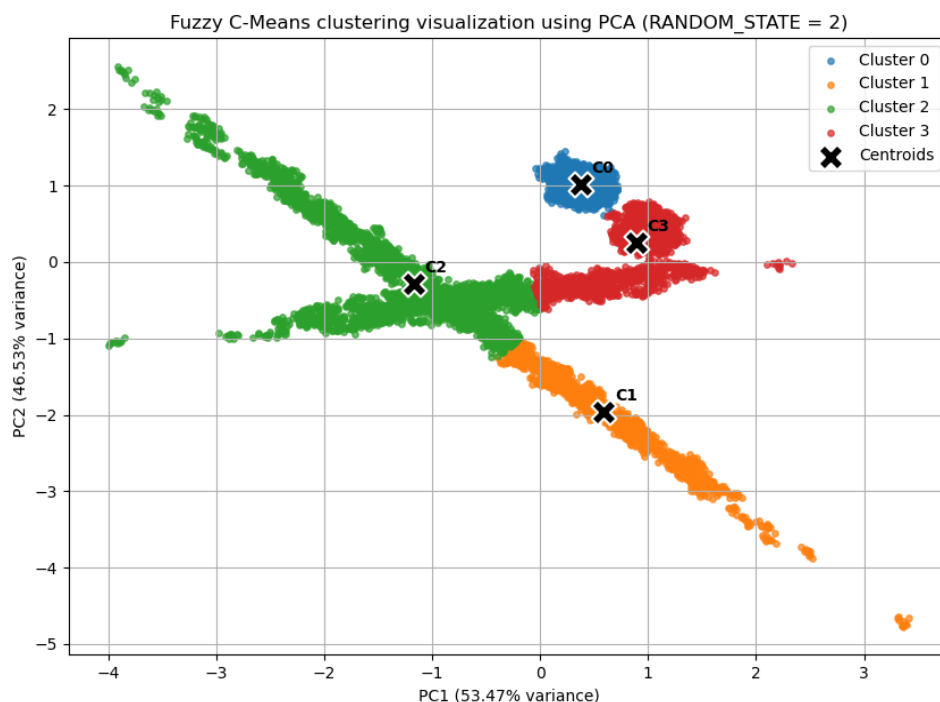


Рисунок 3.1 – Результат кластеризації синтетичного набору Synth 2 алгоритмом Fuzzy C-Means

На відміну від FCM, алгоритм Gustafson-Kessel формує для кожного кластера власну коваріаційну матрицю та адаптивну метрику. Тому під час визначення належності точки враховується не лише відстань до центра, а й поточна форма та орієнтація кластера. На рисунку 3.2 видно, що в цьому запуску ГК точніше відтворює витягнуті структури синтетичного набору та коректніше розділяє точки різних класів.

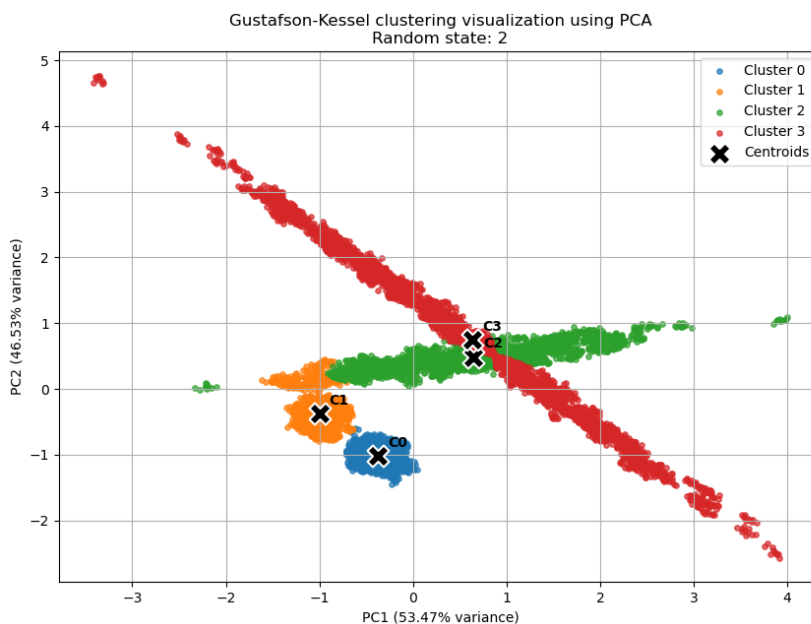


Рисунок 3.2 – Результат кластеризації синтетичного набору Synth 2 алгоритмом Gustafson-Kessel

Водночас наведені результати не означають, що ГК гарантовано формує правильне розбиття за будь-якої ініціалізації. Алгоритм залежить від вибору початкових прототипів, оскільки вони визначають початкове положення центрів і початкову форму кластерів. За невдалого вибору стартових точок адаптивні метрики можуть сформуватися на основі нетипових частин набору, унаслідок чого алгоритм сходиться до менш вдалого розбиття.

Рисунок 3.3 показує, що навіть за наявності адаптивної метрики ГК може некоректно розподілити частину точок витягнутих кластерів.

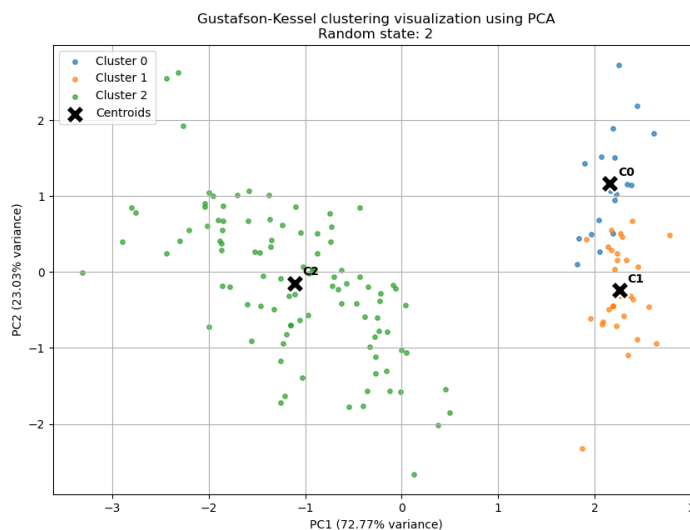


Рисунок 3.3 – Приклад менш вдалого результату кластеризації набору iris алгоритмом Gustafson-Kessel

Для наочного порівняння впливу ініціалізації межі значень ARI за десять запусків наведено в додатку А. На наборі Iris алгоритм GK перевищує FCM за всіма зовнішніми критеріями (ARI, ACI, purity) та має значно меншу ентропію, що означає більш чіткі належності. Водночас FCM є стабільнішим (менші стандартні відхилення), тоді як GK демонструє кращу середню якість, але більшу варіативність результатів. Детальні результати для кожного з десяти запусків (значення критеріїв для окремих random seed) наведено в додатку А, що дозволяє проаналізувати розподіл показників та вплив ініціалізації більш детально.

Після порівняння Fuzzy C-Means та класичного Gustafson-Kessel доцільно окремо оцінити, як перехід від пакетного режиму до послідовного потокового оновлення впливає на якість кластеризації. Для цього класичний GK та його рекурсивний потоковий варіант було застосовано до синтетичного набору Synth 1. У потоковому варіанті кількість кластерів залишається фіксованою, а кожна точка обробляється лише в момент її надходження.

На рисунках 3.4–3.5 видно, що класичний алгоритм має можливість виправити початкові неточності після аналізу всієї вибірки. Навіть якщо

початкові центри були розташовані неідеально, подальші ітерації уточнюють структуру кластерів за всіма точками. У потоковому варіанті така корекція для вже оброблених об'єктів не виконується. Якщо на початковому етапі центр або матриця метрики ще не відповідають фактичній геометрії кластера, частина ранніх точок може отримати неправильні або недостатньо виражені належності.

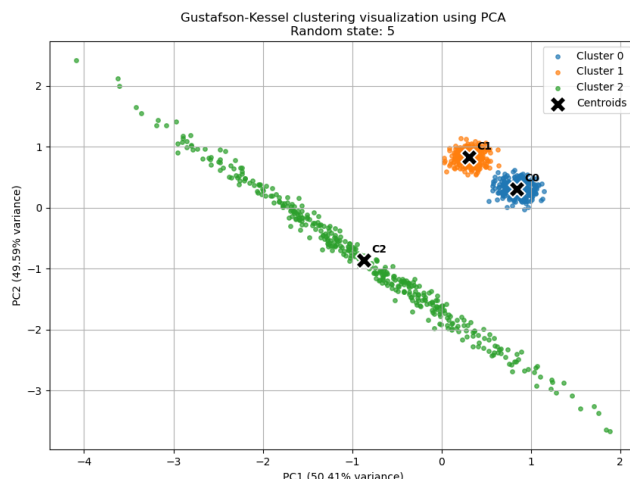


Рисунок 3.4 – Результат кластеризації набору Synth 1 класичним алгоритмом Gustafson-Kessel

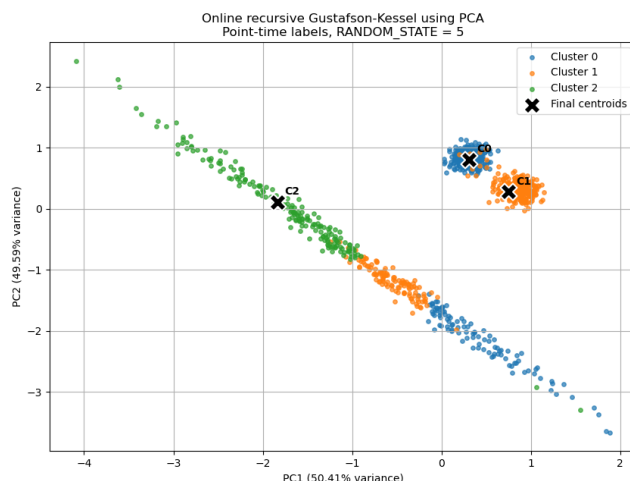


Рисунок 3.5 – Результат кластеризації набору Synth 1 рекурсивним потоковим алгоритмом Gustafson-Kessel

Це особливо помітно на потоках обмеженої довжини. Параметрам кластера необхідно накопичити достатню кількість характерних точок, щоб коректно оцінити його центр, форму та орієнтацію. У класичному GK ця інформація використовується одночасно для всіх об'єктів набору. У рекурсивному варіанті вона надходить поступово, тому модель може не встигнути повністю компенсувати помилки, зроблені на ранньому етапі оброблення потоку. За всіма зовнішніми критеріями класичний GK має суттєво вищу середню якість. Середня чистота кластерів для пакетного алгоритму становить 0,940000, тоді як для потокового варіанта вона дорівнює 0,725800. Різниця за ARI є ще помітнішою: класичний GK має середнє значення 0,891684, а потоковий – 0,421854. Аналогічний результат отримано для ACI: 0,891536 для класичного алгоритму проти 0,362701 для рекурсивного потокового варіанта.

Зростання нормованої ентропії з 0,002382 до 0,341757 показує, що потоковий алгоритм значно частіше формує невизначені належності. Це узгоджується з візуальними результатами: у момент надходження ранньої точки параметри моделі ще не містять достатньої інформації про реальну форму кластерів. Тому точка може мати близькі значення належності до кількох кластерів або бути віднесеною до неправильного кластера. Таким чином, рекурсивний потоковий GK не є повною заміною пакетного алгоритму на коротких потоках: зниження якості зумовлене відсутністю доступу до майбутніх точок і повторного перегляду даних. Наступним етапом експерименту є перевірка впливу адаптивної потокової нормалізації на роботу рекурсивного алгоритму Gustafson-Kessel. Порівнюються два варіанти: без нормалізації (рисунок 3.6) та з адаптивною потоковою нормалізацією (рисунок 3.7).

Метою такого порівняння є визначення фактичного впливу нормалізації на якість кластеризації. Для обох варіантів використано однаковий порядок надходження точок, однакову кількість кластерів, однакові параметри алгоритму та десять значень random seed.

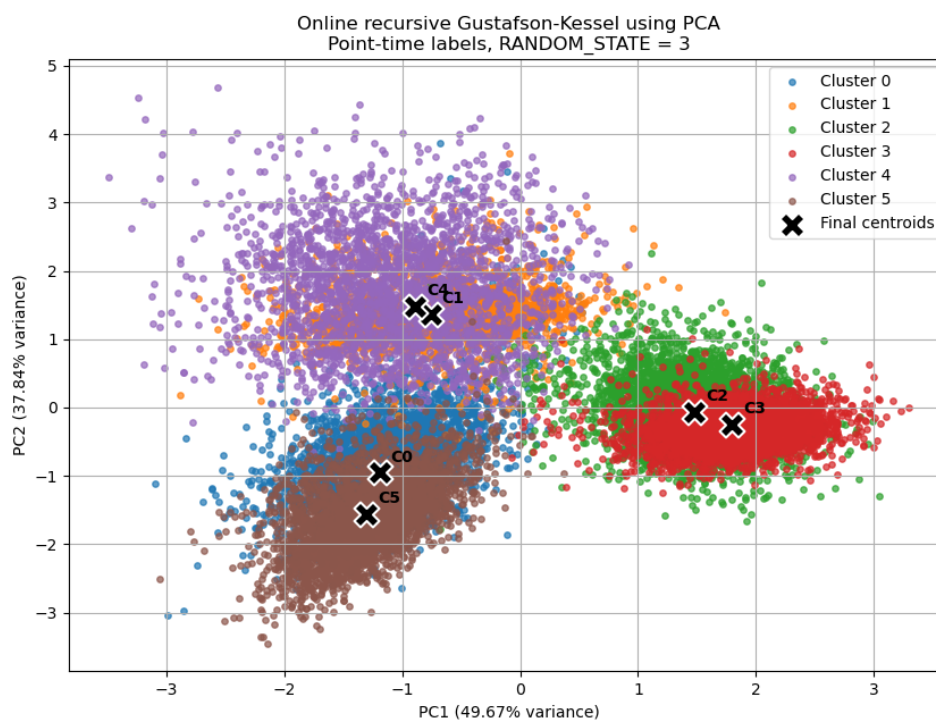


Рисунок 3.6 – Результат роботи рекурсивного потокового ГК на наборі
Synth 4 без адаптивної нормалізації

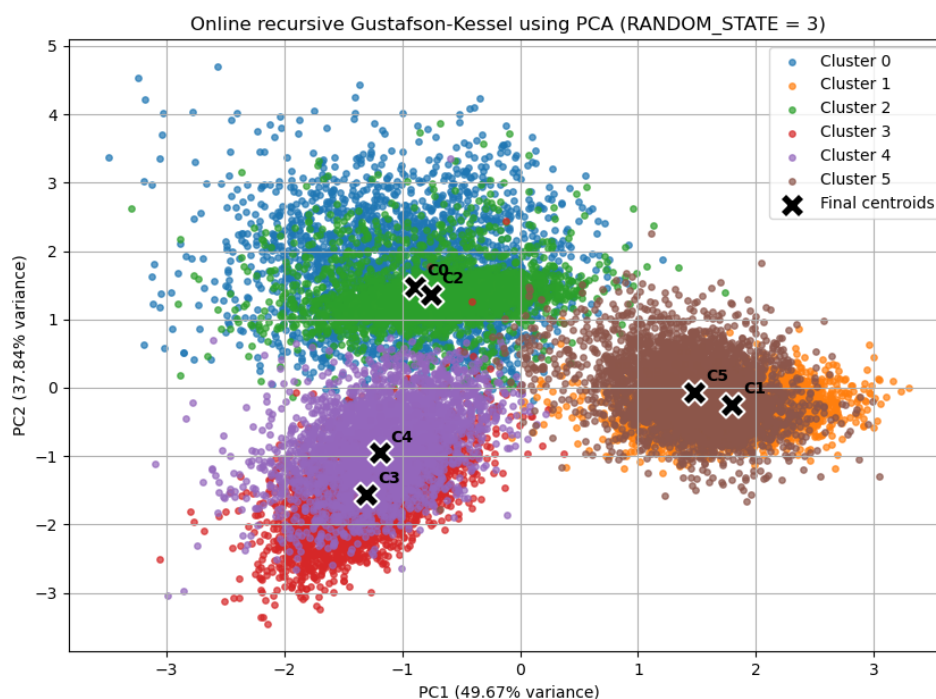


Рисунок 3.7 – Результат роботи рекурсивного потокового ГК на наборі
Synth 4 з адаптивною потоковою нормалізацією

На відміну від пакетної нормалізації, потокова нормалізація не має доступу до повної вибірки. Її параметри уточнюються поступово, тому на ранніх етапах потоку поточні середні значення та масштаби ознак можуть істотно відрізнятися від параметрів, сформованих після оброблення всіх даних. Унаслідок цього модель неодноразово переводиться до нових систем координат, а центри та матриці форми кластерів формуються за ще неповною інформацією про структуру потоку.

За всіма зовнішніми критеріями варіант з адаптивною нормалізацією показав нижчу середню якість. Значення purity зменшилося з 0,725800 до 0,586900. Середнє значення ARI знизилося з 0,421854 до 0,178793, а ACI – з 0,362701 до 0,167815. Отже, у межах цього експерименту адаптивна нормалізація не покращила відповідність отриманих кластерів еталонним міткам класів. Водночас стандартні відхилення purity, ARI та ACI для варіанта з адаптивною нормалізацією є меншими. Наприклад, стандартне відхилення ARI зменшилося з 0,150371 до 0,040032. Це означає, що варіант із потоковою нормалізацією працює більш одноманітно щодо випадкової ініціалізації, проте стабільно формує розбиття нижчої якості. Отриманий результат не означає, що адаптивна потокова нормалізація є недоцільною в усіх задачах. Її призначення полягає у підтриманні порівнюваного масштабу ознак у потоці, де середні значення або розсіювання можуть змінюватися з часом. Проте на обмеженому потоці параметри нормалізації ще не встигають стабілізуватися. Таким чином, у розглянутій конфігурації адаптивна потокова нормалізація призвела до зниження якості кластеризації. Це є важливим проміжним результатом: додавання механізму, необхідного для оброблення змінних поточкових даних, не гарантує покращення на кожному обмеженому та стаціонарному наборі.

Завершальним етапом експериментальної перевірки є порівняння рекурсивного алгоритму Gustafson-Kessel з адаптивною потоковою нормалізацією (рисунки 3.8) та повної розробленої моделі (рисунки 3.9). Експеримент проведено на розширеному до 3000 спостережень наборі

даних Iris. Такий обсяг потоку дає змогу оцінити не лише початковий етап формування моделі, а й її подальшу стабілізацію після накопичення значної кількості спостережень. Для обох варіантів використано однаковий порядок надходження даних, однакові базові параметри та десять запусків зі значеннями random seed від 1 до 10.

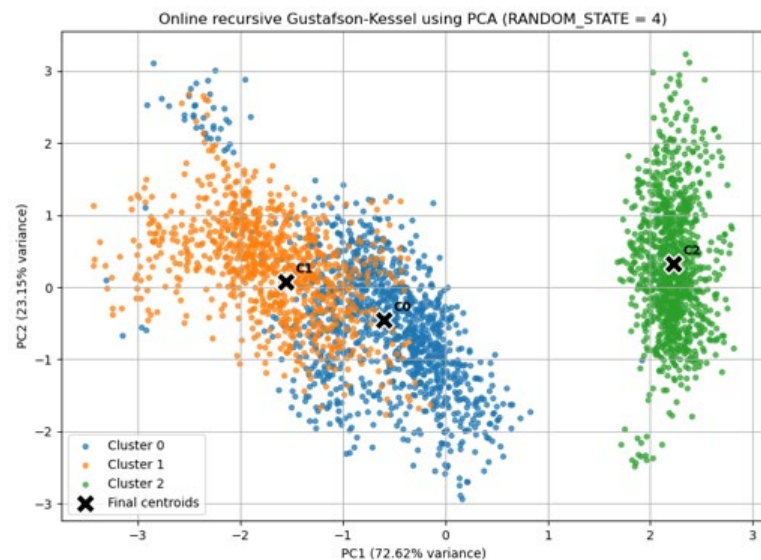


Рисунок 3.8 – Результат роботи рекурсивного ГК з адаптивною потоковою нормалізацією на розширеному наборі Iris

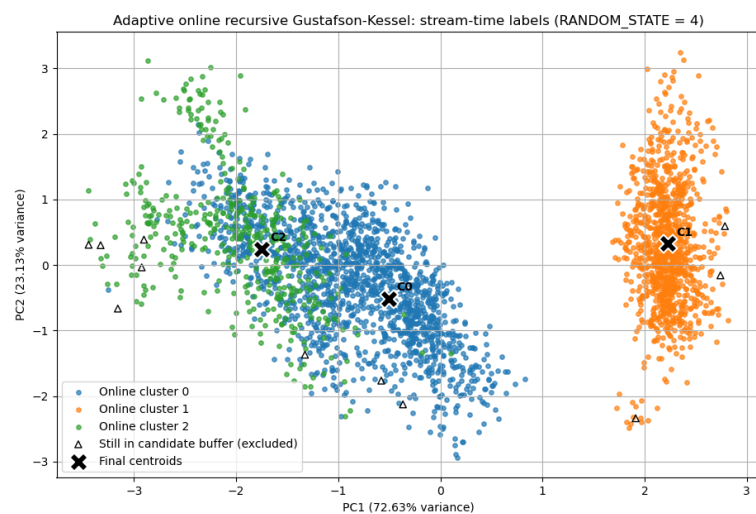


Рисунок 3.9 – Результат роботи повної потокової моделі на розширеному наборі Iris

За показниками purity та ARI варіант лише з адаптивною нормалізацією має вищі середні результати. Значення purity становить 0,913667 проти 0,794115 для повної моделі. Значення ARI також зменшується з 0,794406 до 0,575978. Отже, введення буфера кандидатів і механізму зміни кількості кластерів не покращило точність жорсткого розбиття на цьому наборі даних. Таке зниження не є несподіваним. У базовому варіанті кожна точка обов'язково відноситься до одного з уже наявних кластерів. Повна модель працює обережніше: точки, які не відповідають поточній структурі, не отримують остаточної мітки одразу, а залишаються у буфері до появи достатніх підстав для їх прийняття або формування нового кластера. Крім того, створення додаткового кластера може призвести до поділу одного еталонного класу між кількома сформованими кластерами. Така ситуація зменшує ARI, оскільки цей критерій штрафує за розділення об'єктів одного правильного класу.

Водночас за ACI повна модель має дещо вище середнє значення: 0,575343 проти 0,543109. Це означає, що нечіткі належності прийнятих точок у середньому краще відповідають еталонній структурі. Найпомітніша різниця спостерігається для нормованої ентропії. Її значення зменшується з 0,486720 до 0,015418. Отже, точки, які були прийняті до активних кластерів повною моделлю, отримують значно більш виражену належність до одного кластера.

Зменшення ентропії не слід розглядати окремо як безумовне покращення. Проте в цьому випадку воно супроводжується дещо вищим ACI та меншим значенням Xie-Beni point-time: 0,466309 проти 0,759864. Це свідчить, що активні кластери повної моделі в середньому є компактнішими та краще відокремленими один від одного. Водночас ARI і purity показують, що така компактність досягається ціною менш точного відтворення жорсткої еталонної розмітки всього набору.

Окремо слід оцінити стабільність результатів щодо початкової ініціалізації. Для повної моделі стандартне відхилення ARI

становить 0,042912, тоді як для варіанта лише з адаптивною нормалізацією воно дорівнює 0,139301. Аналогічне зменшення варіативності спостерігається для purity, ACI, ентропії та Xie-Beni.

Отже, використання буфера кандидатів та перевірки сумісності зменшує вплив випадкового вибору початкових прототипів. Повна модель у середньому поступається за ARI, проте її результати є значно стабільнішими. Середня кількість прийнятих точок становить 2978,1 із 3000, тобто 99,27 % потоку було включено до активних кластерів. У буфері після завершення потоку в середньому залишалося 21,9 точки. Це не є автоматичною помилкою моделі: для таких точок до моменту завершення потоку не надійшло достатньо даних, щоб підтвердити їх належність до існуючого або нового кластера. Згідно з розробленим алгоритмом, вони зберігають статус непідтверджених кандидатів. Таким чином, повна потокова модель не перевищила варіант лише з адаптивною нормалізацією за точністю жорсткого відтворення класів Iris. Проте вона забезпечила значно меншу невизначеність нечітких належностей, нижче значення Xie-Beni point-time та істотно вищу стабільність результатів між різними ініціалізаціями. Крім того, модель продемонструвала здатність самостійно збільшувати кількість кластерів і тимчасово ізолювати точки, для яких недостатньо підстав для негайного призначення. Отже, її перевага полягає не в безумовному зростанні ARI на стаціонарному наборі Iris, а у реалізації механізмів, необхідних для роботи з потоками даних за невідомої або змінної структури.

ВИСНОВКИ

У кваліфікаційній роботі розв'язано задачу розроблення адаптивного методу нечіткої кластеризації потокових даних на основі алгоритму Gustafson-Kessel за умов апіорної та поточної невизначеності. У ході дослідження проаналізовано особливості потокового опрацювання даних, відмінності чіткої та нечіткої кластеризації, а також обмеження класичних алгоритмів Fuzzy C-Means і Gustafson-Kessel.

Обґрунтовано вибір алгоритму Gustafson-Kessel як бази розроблюваного методу. На відміну від Fuzzy C-Means, він використовує окремі адаптивні метрики для кожного кластера, що дозволяє враховувати витягнуту форму груп, їх орієнтацію та залежності між ознаками. Розроблено рекурсивний потоковий варіант GK, у якому центри, нечіткі ваги, коваріаційні матриці та метрики оновлюються послідовно без зберігання всієї історії спостережень.

Для узгодження параметрів моделі зі зміною статистик потоку реалізовано адаптивну нормалізацію ознак із переведенням центрів і коваріаційних матриць до нової системи координат. Для врахування невідомої кількості кластерів і віддалених спостережень введено буфер кандидатів. Точки, що не відповідають активним кластерам, тимчасово не впливають на модель, а надалі можуть бути прийняті до наявного кластера, використані для створення нового кластера або зберегти статус непідтверджених кандидатів.

Експериментальна перевірка показала, що GK точніше за FCM відтворює структуру синтетичних даних із витягнутими кластерами. Водночас перехід до потокової обробки на коротких потоках призводить до зниження якості порівняно з пакетним алгоритмом через відсутність доступу до майбутніх даних і повторного перегляду попередніх рішень. Адаптивна нормалізація не забезпечила універсального покращення на стаціонарному наборі, проте зменшила залежність результатів від

початкової ініціалізації. Повна модель продемонструвала стабільнішу роботу, можливість зміни кількості активних кластерів та контрольоване ізолювання віддалених точок.

Подальші дослідження доцільно спрямувати на перевірку методу на потоках із появою нових класів, зміною масштабу ознак і контрольовано доданими аномаліями.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Dunn J. C. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of cybernetics*. 1973. Vol. 3, no. 3. P. 32–57. URL: <https://doi.org/10.1080/01969727308546046> (date of access: 16.06.2026).
2. Bezdek J. C. Objective function clustering. *Pattern recognition with fuzzy objective function algorithms*. Boston, MA, 1981. 256 p. URL: https://doi.org/10.1007/978-1-4757-0450-1_3 (date of access: 16.06.2026).
3. Gustafson D., Kessel W. Fuzzy clustering with a fuzzy covariance matrix. *1978 IEEE conference on decision and control including the 17th symposium on adaptive processes*, San Diego, CA, USA, 10–12 January 1979. 1978. URL: <https://doi.org/10.1109/cdc.1978.268028> (date of access: 18.06.2026).
4. Krishnapuram R., Keller J. M. A possibilistic approach to clustering. *IEEE transactions on fuzzy systems*. 1993. Vol. 1, no. 2. P. 98–110. URL: <https://doi.org/10.1109/91.227387> (date of access: 18.06.2026).
5. Aggarwal C. C. Data streams: models and algorithms. Springer, 2007.
6. A framework for clustering evolving data streams / C. C. Aggarwal et al. *Proceedings 2003 VLDB conference*. 2003. P. 81–92. URL: <https://doi.org/10.1016/b978-012722442-8/50016-1> (date of access: 22.06.2026).
7. Density-Based clustering over an evolving data stream with noise / F. Cao et al. *Proceedings of the 2006 SIAM international conference on data mining*. Philadelphia, PA, 2006. URL: <https://doi.org/10.1137/1.9781611972764.29> (date of access: 23.06.2026).
8. A survey on concept drift adaptation / J. Gama et al. *ACM computing surveys*. 2014. Vol. 46, no. 4. P. 1–37. URL: <https://doi.org/10.1145/2523813> (date of access: 25.06.2026).

9. Evolving fuzzy systems – methodologies, advanced concepts and applications / ed. by S. (. Service). Berlin, Heidelberg : Springer-Verlag Berlin Heidelberg, 2011. 454 p.

10. Welford B. P. Note on a method for calculating corrected sums of squares and products. *Technometrics*. 1962. Vol. 4, no. 3. P. 419–420. URL: <https://doi.org/10.1080/00401706.1962.10490022> (date of access: 25.06.2026).

11. Hubert L., Arabie P. Comparing partitions. *Journal of classification*. 1985. Vol. 2, no. 1. P. 193–218. URL: <https://doi.org/10.1007/bf01908075> (date of access: 29.06.2026).

12. Xie X. L., Beni G. A validity measure for fuzzy clustering. *IEEE transactions on pattern analysis and machine intelligence*. 1991. Vol. 13, no. 8. P. 841–847. URL: <https://doi.org/10.1109/34.85677> (date of access: 29.06.2026).

13. Adjusted concordance index: an extensionl of the adjusted rand index to fuzzy partitions / A. D'Ambrosio et al. *Journal of classification*. 2020. URL: <https://doi.org/10.1007/s00357-020-09367-0> (date of access: 01.06.2026).

14. Robust recurrent credibilistic modification of the Gustafson-Kessel algorithm / Y. Bodyanskiy et al. *Lecture notes in computational intelligence and decision making*. Cham, 2021. P. 613–623. URL: https://doi.org/10.1007/978-3-030-82014-5_42 (date of access: 19.06.2026).

15. Credibilistic robust online fuzzy clustering in data stream mining tasks / A. Y. Shafronenko et al. *Radio electronics, computer science, control*. 2023. No. 3. P. 97. URL: <https://doi.org/10.15588/1607-3274-2023-3-10> (date of access: 22.06.2026).