

<https://doi.org/10.31891/2219-9365-2025-84-55>

УДК 004.8:004.451.2

КОРЕЦЬКИЙ Олександр

Державний університет інформаційно-комунікаційних технологій

<https://orcid.org/0009-0001-4809-7556>

e-mail: okoretsky@gmail.com

МОДЕЛЬ КЛАСТЕРІЗАЦІЇ КОРИСТУВАЧІВ ФРЕЙМВОРКУ, ПОБУДОВАНОГО НА ПЛАТФОРМІ МІКРОСЕРВІСНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

В роботі подані результати розробки моделі кластеризації користувачів фреймворку, побудованого на платформі мікросервісного програмного забезпечення. Подана модель кластеризації користувачів фреймворку, побудованого на платформі мікросервісного програмного забезпечення на основі методології кластеризації k -середніх, дозволяє згрупувати користувачів фреймворку в зоні скупчення в межах визначених кордонів кластерів по умові співпадіння центру скупчення користувачів та центру кластера. Одержані результати показали, що подана модель дозволяє згрупувати користувачів в зоні розподілу в кластери відповідно поданим розмірам по критерію співпадіння центрів розподілу користувачів та центрів кластерів. Це забезпечує подальшу оцінку навантаження на обчислювальні можливості, яке може створити відповідний кластер групування користувачів.

Висвітлені в роботі результати оцінки застосовності поданої моделі показали, її придатність до згрупувати користувачів фреймворку в зоні скупчення в межах визначених кордонів кластерів, що може стати основою для подальшого визначення доступних Fog-пристроїв, міграції мікросервісів відповідно характеристик кластера на відповідний Fog-пристрій для здійснення розподілених туманних динамічних обчислень.

Ключові слова: фреймворк, мікросервісне програмне забезпечення, кластеризація користувачів, розподілені туманні динамічні обчислення.

KORETSKYI Oleksandr

State University of Information and Communication Technology

USER CLUSTERING MODEL OF A FRAMEWORK BUILT ON A MICROSERVICE SOFTWARE PLATFORM

The paper presents the results of developing and evaluating a user clustering model designed for a framework implemented on a microservice-based software platform. The proposed approach focuses on efficient user grouping as a prerequisite for optimizing resource allocation and load distribution in distributed computing environments. The user clustering model is based on the k -means clustering methodology and enables the formation of stable user groups within predefined clustering boundaries under the condition of coincidence between the user clustering center and the corresponding cluster center.

The model operates by analyzing user distribution within a designated clustering zone and assigning users to clusters of predefined sizes according to the criterion of minimal distance between user distribution centers and cluster centroids. This approach ensures consistent clustering results and allows the system to maintain balanced clusters that reflect real user behavior patterns. As a result, the proposed model supports an accurate estimation of the computational load generated by each user group, which is essential for effective planning and management of computing resources in microservice-oriented systems.

The experimental results obtained during the applicability assessment demonstrate that the developed model is suitable for grouping framework users within the specified cluster boundaries and distribution constraints. The clustering outcomes confirm the model's ability to adapt to varying user densities while preserving the required cluster sizes and structural stability. This, in turn, enables further evaluation of computing capacity demands associated with each cluster.

The presented user clustering model can serve as a foundation for advanced resource management mechanisms, including the identification of available Fog computing devices and the dynamic migration of microservices based on cluster characteristics. By mapping user clusters to appropriate Fog nodes, the proposed approach supports the implementation of distributed fog-based dynamic computing, improves system scalability, and enhances the overall efficiency and responsiveness of microservice platforms operating in heterogeneous and geographically distributed environments.

Keywords: framework, microservice software, user clustering, distributed fog dynamic computing.

Стаття надійшла до редакції / Received 30.09.2025

Прийнята до друку / Accepted 22.08.2025

ПОСТАНОВКА ПРОБЛЕМИ У ЗАГАЛЬНОМУ ВИГЛЯДІ ТА ЇЇ ЗВ'ЯЗОК ІЗ ВАЖЛИВИМИ НАУКОВИМИ ЧИ ПРАКТИЧНИМИ ЗАВДАННЯМИ

Необхідність забезпечення розвитку глобальних економічних відносин, інтенсивний розвиток науки та посилення сфери оборони вимагають постійного удосконалення та розвитку галузі інформаційних технологій в напрямку збільшення швидкості та якості передачі корисних даних. Одним з напрямків досліджень вказаній сфері є розбудова та швидке впровадження перспективних концепцій мереж передачі даних, а саме стандарту 5G/IMT-2020 та, в його розвиток, наступних поколінь таких мереж.

Наразі впровадження інформаційно-комунікаційних мереж на основі концепції високонадійного зв'язку з низькою затримкою (URLLC) є одним із найскладніших завдань, що стоять перед науково-технічною

спільнотою. Основною вимогою до мереж URLLC є висока надійність передачі даних з мінімальною затримкою [1,2].

Ці сервіси впроваджуються в таких галузях, як електронна медицина (e-health), автономний транспорт, промислове виробництво небезпечних матеріалів у форматі 4.0 та інші.

Одним із найвідоміших типів мережевих сервісів URLLC є Інтернет речей (IoT) та Інтернет дотику.

Основні цілі та вимоги для URLLC [2,3]:

1. Мінімальна затримка (цільові P99/P999 latency).
2. Висока надійність (низький packet loss, швидке відновлення).
3. Детермінованість — передбачувана поведінка під навантаженням.
4. Ефективне використання ресурсів — CPU/GPU/FPGA, мережа, пам'ять

Одним з аспектів досягнення вище поданих цілей та забезпечення вище поданих вимог є ефективне застосування програмного забезпечення, яке в кінцевому і елементом, який безпосередньо вирішує функціональні завдання, які викликають відповідні обсяги та об'єми трафіку в напрямку відповідного сервісу телекомунікаційної мережі.

Наразі у світі розробки програмного забезпечення накопичено достатньо знань щодо ефективного створення коду, методів, принципів та підходів до розгортання ІТ-продуктів (з точки зору коду) [3,4]. Також існують різні архітектурні рішення, включаючи монолітні, багатопарові, подієво-керовані, безсерверні та мікросервісні [3,4]. Протягом останніх 5–6 років архітектурний стиль мікросервісів у розробці та розгортанні програмного забезпечення активно розвивається, особливо після його успішного пілотного застосування в таких відомих ІТ-продуктах, як Amazon та Netflix. Ця архітектура важлива для інфраструктурних рішень, таких як платформи Інтернету речей, веб-платформи, що реалізують більше одного продукту, та інші [4,5].

На Рис.1 зображено приклад послуги, реалізованої на основі мікросервісного підходу, яка забезпечує функції покращення знімків, зроблених на користувацьких пристроях. Така послуга реалізована, наприклад, у рамках застосунку Google Photos.

Сервіс реалізовано в такому порядку. Після того, як користувач робить фотографію камерою мобільного пристрою, фотографія надсилається до найближчого мікросервісу, який готує її до обробки штучною нейронною мережею (наприклад, згорткового типу). Потім, після попередньої обробки, фотографія надсилається до мікросервісу, де реалізується обробка за допомогою штучної нейронної мережі (штучна нейронна мережа [4, 5]).

Цей приклад демонструє базовий сервіс, який може бути розгорнутий у рамках даного фреймворку. Наприклад, цікавішим і актуальнішим напрямом, якщо говорити про обробку відео/фото контенту, є застосунки доповненої реальності (AR – Augmented Reality) і/або віртуальної реальності (VR – Virtual Reality) [4,5,6].

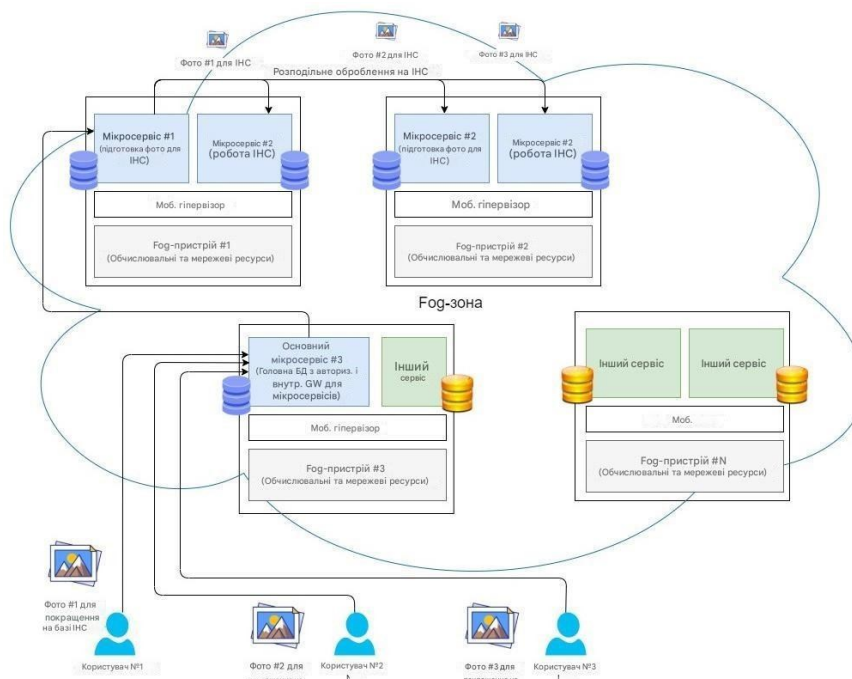


Рис. 1 Функціональна схема прикладу роботи мікросервісної послуги

Для розпізнавання об'єктів на фотографії, а також для забезпечення високої якості потрібні рішення машинного зору на основі штучних нейронних мереж. А з цього витікає, що для широкого впровадження AR/VR-застосунків у життя людей і в різні види діяльності необхідні значні обчислювальні та мережеві

ресурси, які формують прикладну проблему пошуку можливостей доступу до таких ресурсів, в тому числі виходячи з їх наявності та застосованості в даний момент часу в самій мережі.

Протягом останніх трьох-чотирьох років міжнародна наукова спільнота все частіше публікує статті, що обґрунтовують можливість використання пристроїв з обмеженими обчислювальними та мережевими можливостями як повноцінних вузлів у розподіленому обчислювальному кластері. Такий підхід отримав назву «туманні обчислення» — за аналогією з природним явищем. У цьому контексті набір пристроїв (користувацькі термінали, пристрої Інтернету речей, елементи периферійної мережі тощо), виділяючи частину своїх ресурсів, створює в архітектурі мережі набір малих обчислювальних вузлів — своєрідну «крапельку туману» [7,8].

Розподілені туманні динамічні обчислення можуть сприяти реалізації напряму мікросервісної послуги, надаючи величезні обчислювальні можливості (завдяки сукупності навколишніх пристроїв), зменшуючи навантаження на мережу і мінімізуючи затримки у передачі даних.

ПОСТАНОВКА ЗАДАЧІ

Міжнародна рекомендація МСЕ-T Q.3745 «Протокол для застосунків Інтернету Речей із часовими обмеженнями поверх програмно-конфігурованих мереж» описує не лише протокол, але й відповідний фреймворк [9].

Цей фреймворк визначає структуру системи та відповідні системні процеси, у яких уже застосовується протокол для взаємодії між сервером послуг Інтернету Речей та системним застосунком програмно-конфігурованої мережі. Запропонований фреймворк має на меті об'єднання хмарних структур багаторівневих пограничних обчислень і туманних обчислень, а також мережевої інфраструктури, побудованої на основі технологій SDN/NFV, з урахуванням рівня оркестрації. Подана таким чином, об'єднана інфраструктура дозволяє реалізовувати нові підходи до розподілу обчислень, зокрема на межі мережі та навіть на самих кінцевих пристроях, включаючи пристрої Інтернету Речей.

У межах розглянутої архітектури фреймворку виділяється ряд елементів, які виконують специфічні обчислювальні функції. Структура MEC передбачає ієрархію елементів, підпорядкованих нижчим рівням хмарних обчислень. Водночас у структурі туманних обчислень суворої ієрархії не спостерігається. Якщо грамотно побудувати туманні структури з урахуванням технічних характеристик і обчислювальних можливостей кожного з Fog-пристроїв, їхнього динамічного розподілу в просторі, а також технологій взаємодії пристрій-пристрій (D2D – Device-to-Device), можна досягти значного синергетичного ефекту.

Варто зазначити, що в межах цієї структури відбувається динамічний розподіл Fog-пристроїв, що призводить до зміни ресурсоемності кожної із Fog-зон. Над цією динамічною обчислювальною інфраструктурою здійснюється міграція мікросервісів застосунків тієї чи іншої послуги. У межах цього фреймворку реалізується певна логіка взаємодії функціональних елементів при підключенні нового Fog-пристрою. Логіка взаємодії відображена у вигляді діаграм повідомлень на Рис.2

Як було зазначено раніше, у межах фреймворку передбачається вирішення комплексного завдання, пов'язаного з динамічним перерозподілом Fog-пристроїв, які будуть орані для проведення певних визначених обчислень [6,7].

Для вирішення низки підзадач у межах зазначеного завдання пропонується використовувати набір ефективних алгоритмів обробки даних.

Загальний алгоритм, закладений у фреймворк, логіка взаємодії функціональних елементів, що подана на Рис.2 вимагає вирішення декількох наукових завдань, пов'язаних з розробкою меделей та методів визначенням центру скупчення користувачів (запитів від користувачів) певного сервісу та, одночасного, визначення обчислювального потенціалу певного середовища туманних обчислень для вибору пристрою, на який буде здійснюватися міграція мікросервісу.



Рис.1 Загальна діаграма взаємодії основних елементів фреймворку

Для вирішення низки часткових задач у межах зазначеного наукового завдання пропонується використовувати набір ефективних алгоритмів обробки даних.

Загальний алгоритм, закладений у фреймворк, передбачає паралельне вирішення двох завдань:

1. Кластеризація користувачів (запитів від користувачів на обробку даних) при умові групування центрів їх концентрації в центрах визначених, для подальшої міграції кластерів;

2. Пошук Fog-пристрою, що має обчислювальний потенціал певного середовища туманних обчислень та на який буде здійснюватися міграція мікросервісу.

В даній роботі буде розглянуте питання, пов'язане з кластеризацією користувачів (запитів від користувачів на обробку даних) при умові групування центрів їх концентрації в центрах визначених кластерів певного функціонального середовища з метою подальшого розподілу запитів в межах одного фреймворку.

Це формує актуальне наукове завдання по розробці моделі кластеризації користувачів фреймворку, побудованого на платформі мікросервісного програмного забезпечення.

АНАЛІЗ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

Питання визначення обсягів запитів від користувачів на обробку даних, побудові моделей їх систематизації та кластеризації присвячено ряд наукових робіт [8,10-13].

В статті [8] подано огляд підходів до fog-обчислень, коли вузли та користувачі мобільні, топологія змінюється. Огляд показує тренди та проблеми динаміки та розподіленості розрахунків, але безпосередні питання розподілу користувачів в даній роботі не розглядаються..

Стаття [10] присвячена Fog-обчисленням в динамічних бездротових mesh-системах з використанням архітектури NDN (Named Data Networking). Враховується мобільність вузлів, стан мережових зв'язків, що змінюється, розподілені ресурси. Подана модель не враховує безпосередні запити користувачів та їх розподіл в зонах розосередження. Відповідно, в даній моделі відсутні складові кластеризації користувачів.

Огляд алгоритмів планування завдань у fog-середовищах — характеристика, проблеми, напрямки майбутніх досліджень подано в роботі [11]. Питання розподілу користувачів та врахування їх щільності в зонах розподілу в даній роботі визначено, як одна з проблем планування завдань у fog-середовищах. Безпосередню механізми кластеризації в даній роботі не подано.

В статті [12] пропонується динамічний розподіл fog-вузлів на групи (працюючі, резервні, «сплячі») з прогнозом навантаження, щоб оптимізувати енергоспоживання та затримки. Подано приклад динамічного управління ресурсами з екологічним акцентом але без акцентування на розрахунки розподілу користувачів в зонах розосередження.

В роботі [13] представлено результати розробки моделі розподілу завдань та ресурсів у Fog-мережі з IoT-пристроями. Акцент в поданій моделі сконцентровано на комплексних системах з підходами та реалізаціями розподіленої динаміки розрахунків.

ФОРМУЛЮВАННЯ ЦІЛЕЙ СТАТТІ

Метою статі є розробка моделі кластеризації користувачів фреймворку, побудованого на платформі мікросервісного програмного забезпечення, яка б дозволила розподілити користувачів в зоні скупчення в межах визначених кордонів кластерів по умові спів падіння центру скупчення і центру кластера

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

Одним із завдань, згідно з алгоритмом фреймворку, зображеним на Рис. 1, є періодичне визначення центру скупчення користувачів. Для пошуку центру пропонується використовувати алгоритм кластеризації K-середніх (англ. k-means). Цей алгоритм характеризується простотою реалізації та швидкістю виконання [14,15].

Основна модель алгоритму кластеризації k-means в полягає в наступному [15,16]

Нехай задано множину об'єктів (векторів ознак):

$$X = \{x_1, x_2, \dots, x_n\}, x_i \in \mathbb{R}^d.$$

Нехай кількість кластерів дорівнює k, а центри кластерів (прототипи) позначимо як

$$\mu_1, \mu_2, \dots, \mu_k \in \mathbb{R}^d.$$

Введемо змінні приналежності z_{ij} :

$$z_{ij} = \{ 1, \text{якщо об'єкт } x_i \text{ належить кластеру } j; 0, \text{інакше} \}.$$

з обмеженням

$$\sum_{j=1}^k z_{ij} = 1, \quad i = 1, \dots, n.$$

Цільова функція (основна модель)

Алгоритм k-means шукає таке розбиття множини X на кластери та такі центри μ_j , які мінімізують суму квадратів відстаней об'єктів до центрів своїх кластерів:

$$J(Z, \mu_1, \dots, \mu_k) = \sum_{i=1}^n \sum_{j=1}^k z_{ij} \|x_i - \mu_j\|^2 \rightarrow \min.$$

Ітераційний алгоритм k-means

1) Крок призначення (assignment): для фіксованих центрів μ_j кожний об'єкт x_i відносимо до найближчого центру за евклідовою відстанню:

$$z_{ij} = 1, \text{ якщо } j = \arg \min_{\ell \in \{1, \dots, k\}} \|x_i - \mu_\ell\|^2;$$

$$z_{ij} = 0, \text{ інакше.}$$

2) Крок оновлення центрів (update):

для фіксованих z_{ij} кожен центр μ_j дорівнює середньому значенню точок свого кластера:

$$\mu_j = (1 / n_j) \sum_{i: z_{ij} = 1} x_i,$$

$$n_j = \sum_{i=1}^n z_{ij}.$$

Ці два кроки повторюються до збіжності (поки майже не

Вихідним результатом є кількість кластерів (визначається до початку процесу кластеризації, і це значення має суттєвий вплив на результат). Через специфіку застосування алгоритму в цьому завданні слід зазначити, що на кількість кластерів впливають метрики, зібрані оператором мобільного зв'язку (наприклад, кількість дзвінків, середня кількість дзвінків на базову станцію за день тощо).

Радіуси кластерів можуть відрізнятися [36], навіть якщо кластери користувачів послуг були створені в одній зоні туману за допомогою одного й того ж процесу з фіксованою кількістю кластерів.

В межах однієї зони туману можуть бути області з різною щільністю кластерів користувачів послуг. Слід зазначити, що розміри цих кластерів також різняться.

Враховуючи алгоритм формальних елементів (FORELL) із заздалегідь визначеним розміром кластера для кластеризації, можливо, що створений кластер не охопить всю площу (або площа покриття значно перевищує площу кластерів користувачів послуг). Тому для знаходження центру кластерів користувачів було використано алгоритм кластеризації k-means. Однією з початкових умов для цієї тези є статична природа пристроїв Інтернету речей, смарт-пристроїв, смартфонів та інших пристроїв, що створюють зони туману.

Радіусом кластера вважається відстань між центром і найбільш віддаленою точкою в кластері. У цій роботі координати x , y , z вимірюються в метрах.

Три площини позиціонування були обрані для врахування висоти розташування користувачького пристрою з метою коректного формування відповідних кластерів. У реальних умовах координати користувачьких пристроїв і Fog-пристроїв пропонується визначати за допомогою GPS. GPS-приймач визначає координати (широту та довготу) за розташуванням щодо трьох супутників і відстані між ними та приймачем (визначається через час передачі радіосигналу від супутників до приймача).

Роботу обраного алгоритму подамо в п'ятих кроках:

1. Задаються вхідні дані: координати x , y , z точок, кількість кластерів і їхні початкові центри. Призначення користувачів (користувачьких пристроїв) до найближчих центрів кластерів здійснюється за допомогою евклідової метрики відстані між точками у просторі:

$$V = \sqrt{(x - x_k)^2 + (y - y_k)^2 + (z - z_k)^2}$$

x, y, z – координати точки

2. Визначення центрів мас

$$\sqrt{(x - x_k)^2 + (y - y_k)^2 + (z - z_k)^2}$$

де l — кількість Fog-пристроїв у кластері.

$$C_{mx} = \frac{\sum_{i=0}^l x_i}{l}, C_{my} = \frac{\sum_{i=0}^l y_i}{l}, C_{mz} = \frac{\sum_{i=0}^l z_i}{l}$$

1. Порівняння центрів мас і передбачуваних центрів кластерів.

2. Якщо центри мас і передбачувані центри кластерів збігаються, то центри кластерів вважаються остаточно визначеними, і всі призначені до них користувачі позначаються як елементи цього кластера.

Якщо центри не збігаються, то кроки 2-4 повторюються з новими передбачуваними центрами кластерів, рівними центрам мас, визначеним на поточній ітерації.

Для оцінки застосовуваності запропонованої моделі було проведено моделювання за допомогою програмно – апаратного комплексу та програмного застосунку. За їх допомогою було згенеровано скупчення користувальських пристроїв, розташовані у 5 ти кластерах.

Можливі центри були розподілені випадковим чином на основі принципу найімовірніших місць розташування в просторі натовпу.

Загалом, координати пристроїв були згенеровані з досліджуваних областей відповідно визначених кластерів.

Кластер 1 згенерував 12 пристроїв користувача, кластер 2 згенерував 48, кластер 3 згенерував 16, кластер 4 згенерував 32, а кластер 5 згенерував 53 пристроїв користувача.

Для ясності та оцінки застосовності запропонованої моделі також були згенеровані вузли Fog. Отримані кластери показано на рисунках 2 та 3, де на рисунку 3 вказані відповідні кластери та їх кількість.

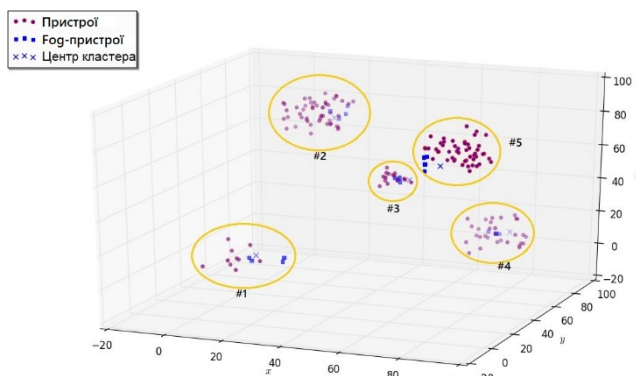


Рис.2 Згенеровані дані про пристрої
(вид №1)

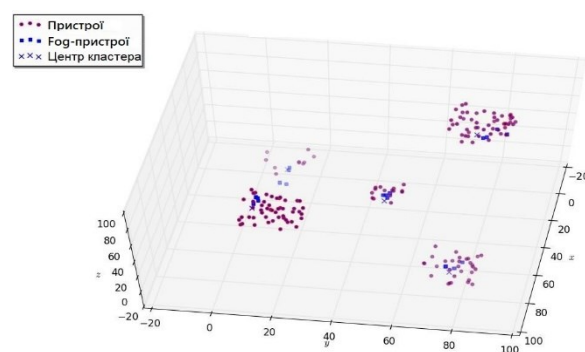


Рис.3 Згенеровані дані про пристрої
(вид №2)

Опис отриманих даних: 3 точки зору практичної (реальної) моделі, кластери 1 та 4 представляють скупчення користувачів у кафе швидкого харчування на перших поверхах будівлі, а також користувачів на наступних поверхах офісної будівлі. Кластери 2, 3 та 5 також відображають скупчення користувачів на верхніх поверхах офісної будівлі, де, наприклад, розташовані офіси відкритого планування, що на практиці означає вищу щільність людей на квадратний метр, або торгову площу торгового центру/комплексу.

На етапі моделювання алгоритмом K-means, відповідно до теорії, визначалися початкові центри користувачів (тобто допустимі центри – стохастичні точки можливого скупчення користувачів). На наступному етапі моделювання алгоритм K-means визначає центри мас (центри скупчення користувачів) та радіус їх розподілу відносно їхнього розташування.

Результат роботи алгоритму, математична суть якого була визначена раніше, представлено на рисунках 4 та 5.

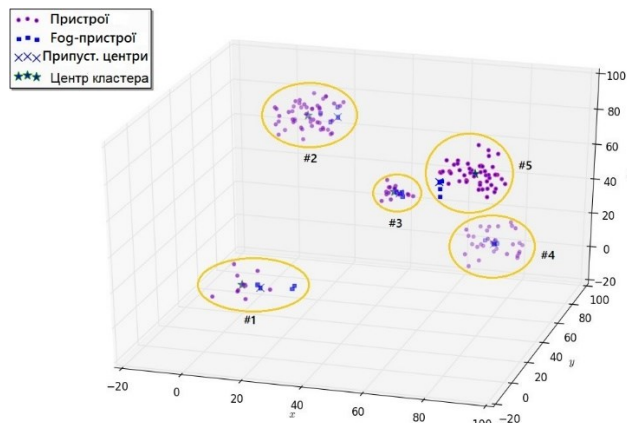


Рис. 4 Результат роботи алгоритму К-середніх (вид даних №1)

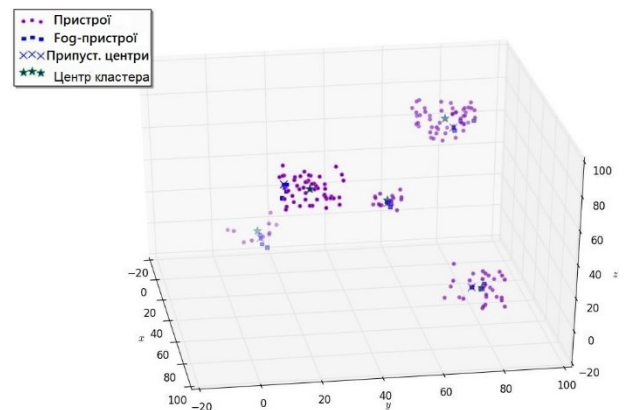


Рис.5 Результат роботи алгоритму К-середніх (вид даних №2)

Центри кластерів, які були розраховані за допомогою алгоритму К-середніх, позначені на Рис 4 та Рис.5 у вигляді зелених зірочок. Користувачські пристрої позначені фіолетовими точками, а припустимі центри кластерів, які були визначені стохастично під час запуску алгоритму, – синіми хрестиками.

У результаті моделювання були визначені центри кластерів (Рис. 4, 5), а також їх радіуси. Центри позначено у вигляді зелених зірочок.

Дані про розраховані радіуси кластерів наведено на гістограмі (Рис. 6).

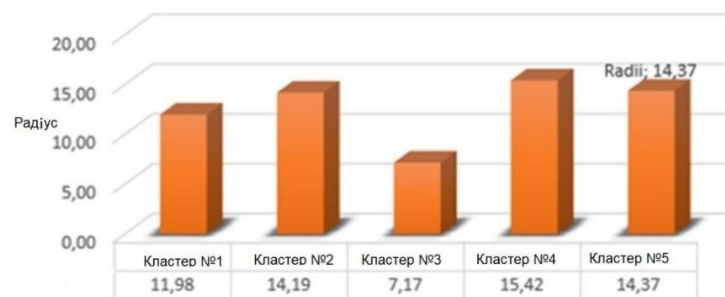


Рис.6 Радіуси кластерів

Згідно з розрахунками, найменшим кластером є кластер 3, який містить 16 пристроїв користувачів (тобто самих користувачів), а найбільшим – кластер 4, який містить 31 пристроїв. Варто зазначити, що з точки зору щільності пристроїв на квадратний метр, кластер 5 є найщільнішим і розташований досить близько до кластера 4, який має найбільший радіус розподілу користувачів.

Результати оцінки застосовності поданої моделі показали, її придатність до згрупувати користувачів фреймворку в зоні скупчення в межах визначених кордонів кластерів, що може стати основою для подальшого визначення доступних Fog-пристроїв, міграції мікросервісів відповідно характеристик кластера на відповідний Fog-пристрій для здійснення розподілених туманних динамічних обчислень.

ВИСНОВКИ З ДАНОГО ДОСЛІДЖЕННЯ І ПЕРСПЕКТИВИ ПОДАЛЬШИХ РОЗВІДОК У ДАНОМУ НАПРЯМІ

1. В роботі подані результати розробки моделі кластеризації користувачів фреймворку, побудованого на платформі мікросервісного програмного забезпечення.

2. Подана модель кластеризації користувачів фреймворку, побудованого на платформі мікросервісного програмного забезпечення на основі методології кластеризації k-середніх, дозволяє згрупувати користувачів фреймворку в зоні скупчення в межах визначених кордонів кластерів по умові співпадіння центру скупчення користувачів та центру кластера.

Одержані результати показали, що подана модель дозволяє згрупувати користувачів в зоні розподілу в кластери відповідно поданим розмірам по критерію співпадіння центрів розподілу користувачів та центрів кластерів. Та оцінити навантаження на обчислювальні можливості, яке може створити відповідний кластер.

3. Висвітлені в роботі результати оцінки застосовності поданої моделі показали, її придатність до згрупувати користувачів фреймворку в зоні скупчення в межах визначених кордонів кластерів, що може стати основою для подальшого визначення доступних Fog-пристроїв, міграції мікросервісів відповідно характеристик кластера на відповідний Fog-пристрій для здійснення розподілених туманних динамічних обчислень.

References

1. Shusta, V. S., Susla, A. I., & Biganych, V. Yu. (2024). Transformation of network technologies: from 4G to 5G. Scientific papers of V. I. Vernadsky TNU. Series: Technical sciences, 3, 94–99.
2. ITU-T. (2018). ITU-T GSTP-TNSG: Transport network support of IMT-2020/5G (Technical Report SG15-TD338/PLEN).
3. ITU-T. (2018). Draft recommendation G.ctn5g: Characteristics of transport networks to support IMT-2020/5G (Draft Recommendation SG15-TD295/PLEN).
4. Koretskyi, O. (2024). Architecture of multifunctional application software microservices. Modern Information Security, 3(59), 85–93. <https://doi.org/10.31673/2409-7292.2024.030009>
5. Richardson, C. (2018). Microservices patterns: With examples in Java. Manning.
6. Gabbrielli, M., Giallorenzo, S., Guidi, C., Mauro, J., & Montesi, F. (2016). Self-reconfiguring microservices. In Theory and practice of formal methods (Lecture Notes in Computer Science, Vol. 9660, pp. 194–210). Springer. https://doi.org/10.1007/978-3-319-30734-3_14
7. Khomenchuk, V. (2023). Analysis of the impact of cloud computing and edge computing on network performance. In Technological Horizons: Research and Application of Information Technologies for Technological Progress in Ukraine and the World (pp. 361–362). Kyiv, Ukraine.
8. Ostrowski, K., Malecki, K., Dziurzański, P., & Singh, A. K. (2023). Mobility-aware fog computing in dynamic networks with mobile nodes: A survey. Journal of Network and Computer Applications, 219, 103724. <https://doi.org/10.1016/j.jnca.2023.103724>
9. ITU-T. (2020). ITU-T Recommendation Q.3745: Protocol for time constraint Internet of things-based applications over software-defined networking. International Telecommunication Union.
10. Srikanteswara, S., Samuylov, A., Arrobo, G., Zhang, Y., Feng, H., Himayat, N., Spoczynski, M., & Koucheryavy, Y. (2024). Provisioning of fog computing over named-data networking in dynamic wireless mesh systems. Sensors, 24(4), 1120. <https://doi.org/10.3390/s24041120>
11. Matrouk, K., & Alatoun, K. (2021). Scheduling algorithms in fog computing: A survey. International Journal of Networking and Distributed Computing, 9, 59–74. <https://doi.org/10.2991/ijndc.k.210111.001>
12. Ali Kumar, D. S. N. K., Newaz, S. H. S., Rahman, F., & Au, T. W. (2022). Green demand aware fog computing: A prediction based dynamic resource provisioning approach. Electronics, 11(4), 608. <https://doi.org/10.3390/electronics11040608>
13. Faraji, F., Javadpour, A., Sangaiah, A., & Zavieh, H. (2023). A solution for resource allocation through complex systems in fog computing for the Internet of Things. Computing, 106, 1–25. <https://doi.org/10.1007/s00607-023-01199-1>
14. Lande, D. V., Subach, I. Yu., & Boyarynova, Yu. Ye. (2018). Fundamentals of the theory and practice of intelligent data analysis in the field of cybersecurity: Textbook. Igor Sikorsky Kyiv Polytechnic Institute.
15. Oleshchenko, L. M. (2022). Machine learning: Textbook. Zaporizhzhia: ZNU.
16. MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In L. M. Le Cam & J. Neyman (Eds.), Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Volume 1: Statistics (pp. 281–297). University of California Press.
17. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction (2nd ed.). Springer.