Конкурсна наукова робота

на тему «Дослідження особливостей кластерного аналізу даних та розробка програмного забезпечення для проведення кластерного аналізу даних»

Галузь 62 Комп’ютерні науки

Шифр - User Experience

ЗМІСТ

ВСТУП

Об'єкт дослідження наданої роботи – кластери, предмет дослідження – дослідження особливостей та призначення кластерного аналізу даних. Залежно від особливостей конкретного завдання кластеризація може мати різні цілі: визначення структури множини даних (шляхом розбиття на групи схожих об'єктів); виділення об'єктів, не придатних до жодного з кластерів; спрощення роботи з даними, коли розглядаються не цілі класи даних, а лише типові представники класів. При такому широкому колі використання кластеризації виникає необхідність в програмних засобах, що надають гнучкі можливості для аналізу даних, так і роботи алгоритмів, а також для зручного представлення результатів. Актуальність моєї дослідницької роботи полягає в тому, що в роботі проведений кластерний аналіз та розроблено програмне забезпечення здійснення кластеризації даних.

Мета роботи: розробка програмного забезпечення розбиття вибірки даних на групи схожих об'єктів для спрощення подальшої обробки даних і прийняття рішень, застосовуючи до кожного кластеру свій метод аналізу.

Завданнями роботи є:

1. Проведення аналізу особливостей та призначення кластерного аналізу даних.

2. Аналіз існуючих методів кластерного аналізу даних та існуючих програмних продуктів з кластеризації.

3. Обґрунтування засобів програмної реалізації.

4. Розробка ментальної карти дослідження.

5. Розробка UML діаграм проекту програмного забезпечення.

6. Розробка алгоритму роботи програми.

7. Програмна реалізація програмного забезпечення кластеризації даних.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ ТА ПРИЗНАЧЕННЯ КЛАСТЕРНОГО АНАЛІЗУ ДАНИХ

Кластерний аналіз (Data clustering) - задача розбиття [7] заданої вибірки об'єктів (ситуацій) на підмножини, які називаються кластерами, так, щоб кожен кластер складався з схожих об'єктів, а об'єкти різних кластерів істотно відрізнялися. Завдання кластеризації відноситься до статистичної обробки, а також до широкого класу задач навчання без керівника.

Кластерний аналіз (КА) - це багатовимірна статистична процедура, що виконує збір даних, що містять інформацію про вибірку об'єктів, і потім впорядковує об'єкти в порівняно однорідні групи (кластери) (Q-кластеризація, або Q-техніка, власне КА) [3,7].

Кластер – це група елементів, які характеризуються загальною властивістю, головна мета КА - знаходження груп схожих об'єктів у вибірці [3].

Спектр застосувань кластерного аналізу дуже широкий: його використовують в археології, медицині, психології, хімії, біології, державному управлінні, філології, антропології, маркетингу, соціології та інших дисциплінах.

КА [17] виконує такі основні завдання:

- розробка типології або класифікації;
- дослідження корисних концептуальних схем групування об'єктів;
- породження гіпотез на основі дослідження даних;
- перевірка гіпотез або дослідження для визначення, чи дійсно типи (групи), які виділені тим чи іншим способом, присутні в наявних даних.

Незалежно від предмета вивчення застосування КА припускає наступні етапи:

- відбір вибірки для кластеризації;
- визначення безлічі змінних, за якими будуть оцінюватися об'єкти у вибірці;
- обчислення значень тієї чи іншої міри подібності між об'єктами;

– застосування методу КА для створення груп схожих об'єктів;

– перевірка достовірності результатів кластерного рішення.

КА [3] пред'являє наступні вимоги до даних:

– показники не повинні корелювати між собою;

– показники повинні бути безрозмірними;

– розподіл показників повинний бути близьким до нормального;

– показники повинні відповідати вимозі «стійкості», під якою розуміється відсутність впливу на їх значення випадкових факторів;

– вибірка повинна бути однорідна, не містити «викидів».

Кластер має наступні математичні характеристики [9]:

1. Центр кластера - це середнє геометричне місце точок у просторі змінних.

2. Радіус кластера - максимальна відстань точок від центру кластера. Кластери можуть перекриватися. У цьому випадку неможливо за допомогою математичних процедур однозначно віднести об'єкт до одного з двох кластерів.

3. Спірний об'єкт - це об'єкт, який у міру подібності може бути віднесений до кількох кластерів.

4. Розмір кластера може бути визначений або по радіусу кластера, або по середньоквадратичному відхиленню об'єктів для цього кластера. Об'єкт відноситься до кластеру, якщо відстань від об'єкта до центру кластера менше радіуса кластера. Якщо ця умова виконується для двох і більше кластерів, об'єкт є спірним.

Неоднозначність даного завдання може бути усунена експертом або аналітиком.

Прийнято ділити все алгоритми кластеризації на ієрархічні і неієрархічні. Класифікація алгоритмів КА в загальному вигляді є наступною (рис.1.1.):

1. Ієрархічні алгоритми [12]:

– агломеративні алгоритми;

– дівізімние алгоритми.

2. Неієрархічні алгоритми:

– ітеративні;
– плотностні;
– модельні;
– концептуальні;
– мережеві.

Розподіл відбувається по видаваних на виході даними. Ієрархічні алгоритми на виході видають певну ієрархію кластерів, тому користувач може вибрати будь-який рівень цієї ієрархії для того, щоб інтерпретувати результати алгоритму [13, 28].

Неієрархічні - це всі алгоритми, які на виході ієрархію не видають (або вибір інтерпретації відбувається не за рівнем ієрархії).

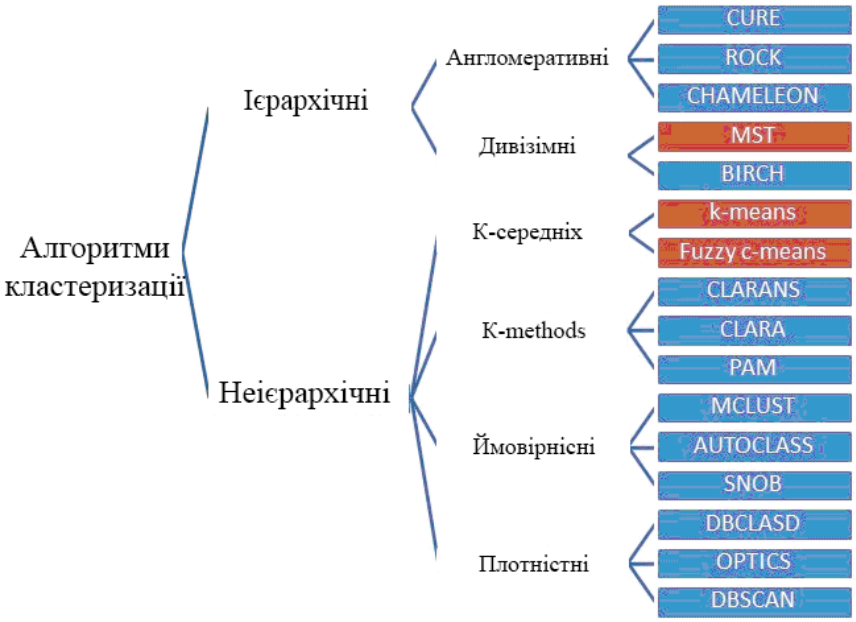

Рисунок 1.1 – Класифікація алгоритмів кластеризації [14]

Ієрархічні алгоритми діляться на агломеративні і дівізімні. Агломеративні алгоритми - це алгоритми, які починають своє виконання з того, що кожен об'єкт заносять в свій власний кластер і в міру виконання об'єднують кластери, до тих пір, поки в кінці не отримує один кластер, що включає в себе всі об'єкти набору.

Дивізімні алгоритми, навпаки, спочатку відносять всі об'єкти в один кластер і потім розділяють цей кластер до тих пір, поки кожен об'єкт не виявиться в своєму власному кластери.

Принцип роботи описаних вище груп методів [15] у вигляді дендрограми наведено на рис. 1.2.

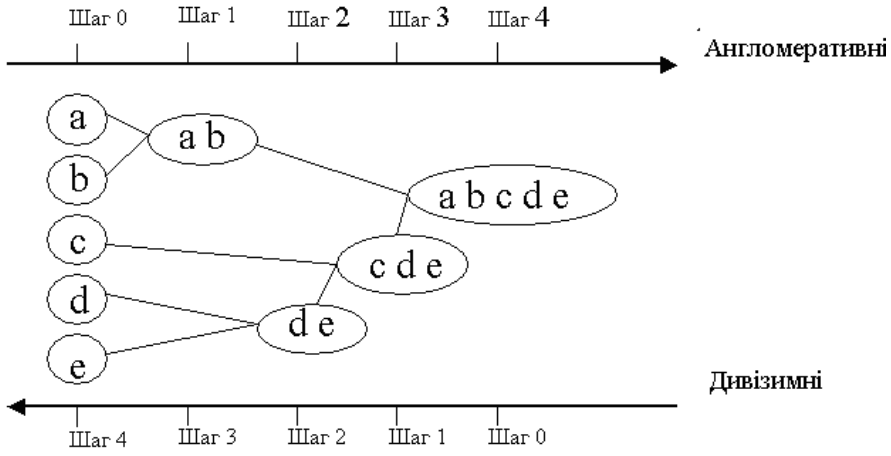

Рисунок 1.2 – Дендрограма агломеративного і дивізімних методів

Ітеративні алгоритми називаються так тому, що ітеративно перерозподіляють об'єкти між кластерами. До них відносять Алгоритм k-means, hard c-means, Farthest First, Алгоритм Fuzzy C-Means. Загальна ідея алгоритмів * -means: Мінімізація відстаней між об'єктами в кластерах. Зупинка відбувається, коли мінімізувати відстані більше вже неможливо. Останній заснований на нечіткій логіці [22].

Інший клас алгоритмів - плотностні. Вони так називаються тому, що визначають кластер як групу об'єктів, розташованих досить купчасто [24]. Наприклад, Алгоритм DBSCAN зазвичай проводиться над даними, впорядкованими в R-дерева (для зручності вибірки навколишніх точок). Але в загальному випадку цього не вимагає [25]. основним його недоліком є нездатність зв'язувати кластери через вузькі місця, де правило щільності не виконується.

Результати порівняльного аналізу деяких розглянутих алгоритмів кластеризації наведено у таблиці 1.1 [28, 29].

Таблиця 1.1 – Результати порівняльного аналізу алгоритмів

Алгоритм кластеризації	**Форма кластерів**	**Вхідні дані**	**Результати**
Ієрархічний	Довільна	Число кластерів або поріг відстані для усічення ієрархії	Бінарне дерево кластерів
K-середніх	Гіперсфера	Число кластерів	Центри кластерів
C-середніх	Гіперсфера	Число кластерів, ступінь нечіткості	Центри кластерів, матриця приналежності
Виділення зв'язкових компонент	Довільна	Поріг відстані R	Деревоподібна структура кластерів
Мінімальне покриваюче дерево	Довільна	Число кластерів або поріг відстані для видалення ребер	Деревоподібна структура кластерів
Пошарова кластеризація	Довільна	Послідовність порогів відстані	Деревоподібна структура кластерів з різними рівнями ієрархії

Розглянуто декілька комп'ютерних програм для статистичної обробки даних: Statistical Package for the Social Sciences (SPSS Statistics), Hierarchical Clustering Explorer (HCE), Statistica.

Ключовою можливістю системи SPSS Statistics [30-31] є двоетапний КА, Ієрархічний КА, КА методом k-середніх. Недоліками системи SPSS є: висока вартість ліцензії, підтримка коректної роботи усіх функцій лише під операційною системою Windows та складність інтерфейсу.

HCE - це програма для аналізу і дослідження даних [32], яка розрахована на одного користувача, зокрема, для проведення ієрархічного кластерного

аналізу даних. Недоліками цієї програми є: низька швидкість роботи, складність розгортання та не інтуїтивний вигляд інтерфейсу користувача.

Statistica - це універсальна інтегрована комерційна система [33], призначена для статистичного аналізу та візуалізації даних, управління базами даних і розробки призначених для користувача додатків, що містить широкий набір процедур аналізу для застосування в наукових дослідженнях, техніці, бізнесі, а також спеціальні методи видобутку даних. Недоліками цієї системи є висока вартість ліцензії для використання та достатньо малі можливості візуалізації результатів проведеного кластерного аналізу даних.

Для створення програмного додатку були використані система математичних досліджень та моделювання Matlab [34-36] і мова програмування Python.

Matlab - це високорівнева мова і інтерактивне середовище для програмування, чисельних розрахунків і візуалізації результатів, в якій можна аналізувати дані, розробляти алгоритми, створювати моделі і додатки. Мова, інструментарій та вбудовані математичні функції Matlab дозволяють досліджувати різні підходи і отримувати рішення швидше, ніж з використанням електронних таблиць або традиційних мов програмування, таких як C / C ++ або Java.

Python – сучасна кросплатформена мова програмування. Характерні особливості даної мови [37]: наявність механізмів динамічної типізації, наявність підтримки модульності, вбудована підтримка Unicode, підтримка парадигми ООП, підтримка автоматичної процедури по «збірці сміття» (відсутні деструктори), гнучкий автоформатіруємий синтаксис, що забезпечує легкість читання коду, підтримка великої кількості модулів і бібліотек, кросплатформеність [38].

РОЗДІЛ 2. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КЛАСТЕРНОГО АНАЛІЗУ ДАНИХ

2.1 Розробка ментальної карти дослідження

Для структурного відображення основних етапів виконання роботи актуальним є використання методу ментальних карт. Це дозволяє формалізувати та чітко виявити порядок виконання поставлених завдань за рахунок декомпозиції найбільш пріоритетних завдань. Розроблена ментальна карта дослідження наведена на рис.2.1. Після виконання аналізу предметної області та обґрунтування засобів розробки, що було виконано в попередньому розділі здійснюється проектування програмного забезпечення (ПЗ) КА даних завдяки використанню UML-нотації. Після цього необхідно виконати опис логіки виконання процесу кластеризації даних різними алгоритмами. На базі цього стає можливим виконання програмної реалізації класів, методів та полів, які забезпечать функціонування ПЗ КА.

Рисунок 2.1 – Ментальна карта дослідження

2.2 Розробка UML діаграм проекту програмного забезпечення

В роботі розроблено: діаграма варіантів використання програмного забезпечення кластерного аналізу, діаграма класів програмного забезпечення кластерного аналізу даних, діаграма послідовності дій виконання ієрархічної кластеризації (додаток 1, рис.1-3).

Діаграма варіантів використання. При завантаженні головної сторінки програми користувач повинен мати змогу побачити стислу інформацію про розробника програмного продукту, переглянути загальний опис призначення кластерного аналізу даних, закрити головне вікно ПЗ (здійснюється вихід з програми) та перейти до форм ієрархічного та неієрархічного КА даних.

Зокрема, на формі ієрархічного КА даних користувач повинен мати змогу завантажити до системи вектори вхідних даних та застосувати для виконання кластеризації відповідний з підтримуваних алгоритм (найближчого сусіда, дальнього сусіда, середнього зв’язку, центроїду чи покроковий). При виконанні КА даних користувач має можливості перегляду додаткових довідкових даних з кожного з перелічених алгоритмів, імпортувати завантажені дані до робочого простору модуля, здійснити обробку даних та візуалізацію результатів кластеризації за допомоги побудови дендрограми, зберегти отримані результати та змінити параметри відображення дендрограми (шрифти, кольори, масштаб та ін.).

Для подальшої програмної реалізації ПЗ КА необхідно виділити ключові сутності, які будуть слугувати об’єктами під час написання коду та використання програми. Для цього необхідною є розробка діаграми класів ПЗ КА даних. Головний клас (Main) повинен здійснювати створення нових екземплярів класів HierarchicalModule та Non HierarchicalModule, кожен з яких має функціонал з обробки однойменного класу, що реалізує відповідний алгоритм КА даних. Для більшої гнучкості загальні функції графічної візуалізації результатів кластеризації на базі ієрархічних агломеративних алгоритмів (побудови дендрограми) винесено до окремого інтерфейсу, якій імплементується відповідними класами. У випадку з візуалізацією результатів кластеризації даних неієрархічними алгоритмами формат графічного відображення не є однаковим для усіх підтримуваних алгоритмів, тому їх узагальнення у інтерфейс не є доцільним, тому виконано у окремих класах. За візуалізацію результатів кластеризації при побудові карт Коххонена відповідає окремих майстер WizardMakerClass.

Для більш детального опису процесу проведення КА засобами розробленого ПЗ доцільно використання sequence diagram, яка дозволяє наочно відобразити переходи між об’єктами програми. Відкривши головну форму ПЗ користувач має змогу здійснити перехід до форми ієрархічного агломеративного КА даних, після чого відбувається завантаження вхідної виборки даних з файлу чи з робочого простору (workspace) системи Matlab. Це дає змогу імпортувати дані для їх обробки та виконання розрахунків за обраним алгоритмом. Після завершення цих дій виконується відображення отриманих результатів при переході до окремої форми візуалізації. Дана форма має функції зміни параметрів відображення та збереження побудованої дендограми до окремого файлу. Після цього користувач має можливість вийти з активного модуля шляхом закриття форми.

2.3 Розробка алгоритмів виконання кластеризації

На базі розробленого проекту ПЗ КА даних є доцільним конкретизація процесу логіко-математичного розрахунку кластерів по основних алгоритмах, використання яких підтримує програма. В роботі розроблено загальний алгоритм виконання ієрархічного агломеративного КА даних (додаток 2, рис.1), загальний алгоритм виконання КА даних на базі карт Кохонена (додаток 2, рис.2). Для роботи даного алгоритму будується штучна нейронна мережа, одразу після імпорту вхідних даних здійснюється конфігурація налаштувань її формування. Після цього виконується ініціалізація синапатичних ваг випадковим чином. На базі перетворення вхідного вектора значень здійснюється розрахунок відстаней між нейронами, визначення та пошук найкращої одиниці відповідності. Після цього стає можливою зміна векторів ваг нейронів та розрахунок помилки карти. У разі, якщо вхідний вектор є останнім у імпортованому наборі даних, то процес розрахунків припиняється та здійснюється візуалізація підсумкової карти.

Також розроблено загальний алгоритм виконання КА даних на базі методу k-середніх (додаток 2, рис.3) і загальний алгоритм виконання КА даних на базі методу с-середніх (додаток 2, рис.4). Метою даного методу кластеризації є автоматична класифікація безлічі об'єктів, які задаються векторами ознак в їх просторі. Даний алгоритм визначає кластери і відповідно класифікує об'єкти. Кластери представляються нечіткими множинами, кордони між якими також є нечіткими. Алгоритм передбачає, що об'єкти належать всім кластерам з певною функцією приналежності.

Ступінь приналежності визначається відстанню від об'єкта до відповідних кластерних центрів. Даний алгоритм ітераційно обчислює центри кластерів і нові ступені приналежності об'єктів.

2.4 Програмна реалізація та опис процесу виконання кластерного аналізу даних

Після виконання етапів формалізації розробки ПЗ КА даних необхідним є розробка інтерфейсу системи, розташування потрібних користувачу графічних компонентів взаємодії та ініціалізації коректної обробки подій.

Проект розробки інтерфейсу головної форми завдяки засобам фреймворку Guide наведено в додатку 3, рис.1.

Головна форма містить 4 кнопки для переходу до відповідного модуля КА даних, перегляду загальної інформації про кластеризацію та виходу з ПЗ відповідно.

2.4.1 Розробка модуля програмного забезпечення здійснення ієрархічного агломеративного кластерного аналізу даних

Розроблена форма модулю здійснення ієрархічного агломеративного КА даних наведено в додатку 3, рис.2.

Для більш зручної навігації у розробленому ПЗ додано кнопку переходу до головної форми. Для завантаження даних з обраного файлу чи з глобального робочого простору системи Matlab використовується кнопка «Завантаження даних». У разі обирання користувачем варіанту імпорту даних з файлу (кнопка «File») системою ініціюється відкриття вікна вибору необхідного файлу з переліку підтримуваних системою форматів вхідних даних. У разі необхідності користувач може переглянути зміст імпортованої змінної у системі, для цього необхідно двічі натиснути на ім’я цієї змінної у робочому просторі Matlab. Результат перевірки коректності завантаження вхідних даних до системи Matlab наведено в додатку 3, рис.3.

Завдяки передбаченій кнопці «Інформація по…» по кожному алгоритму КА даних користувач має змогу переглянути допоміжну інформацію щодо логіки методу. Розроблена форма перегляду стислої довідкової інформації про дію алгоритму найближчого сусіда наведена в додатку 3, рис. 4.

При виконанні процедури обробки та розрахунку для інформування користувача ПЗ про динаміку виконання даного обчислювального процесу передбачено появу лінії прогресу (Progress Bar).

Для перевірки коректності обробки та імпорту вхідних даних з робочого простору системи до окремого модуля КА передбачена поява відповідних діалогових вікон.

Після успішної обробки даних та проведення необхідних розрахунків стає можливим побудова ієрархічної дендограми як наочного результата приналежності об’єктів до окремих кластерів.

Побудована дендограма складається з шарів вершин, кожен з яких представляє кластер. Лінії, що з'єднують вершини, представляють кластери, які вкладені один в інший. Горизонтальний зріз дендрограмми утворює кластеризацію. Результат створеної дендограми виконання кластеризації даних за алгоритмом найближчого сусіда наведено на рис.2.2. Результат створеної дендограми виконання кластеризації даних за алгоритмом середнього зв’язку наведено на рис.2.3.

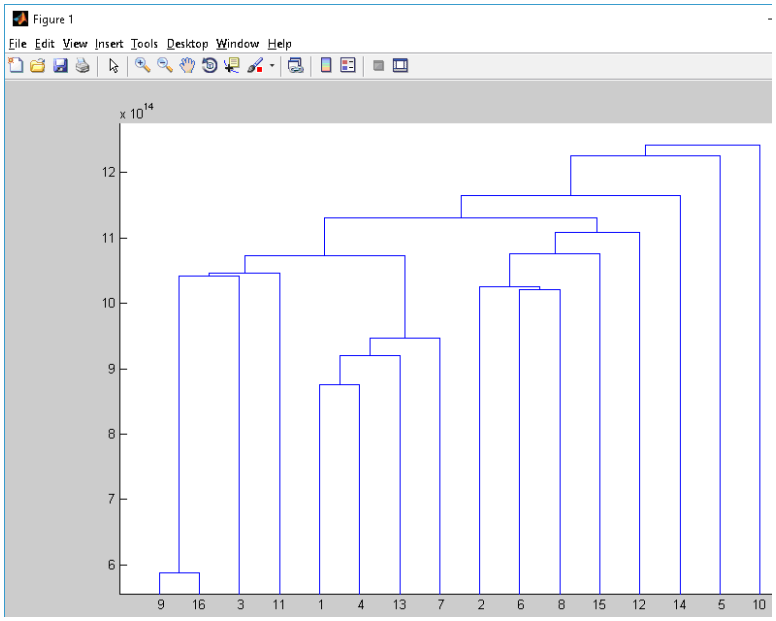

Рисунок 2.2 – Результат створеної дендограми виконання кластеризації даних за алгоритмом найближчого сусіда

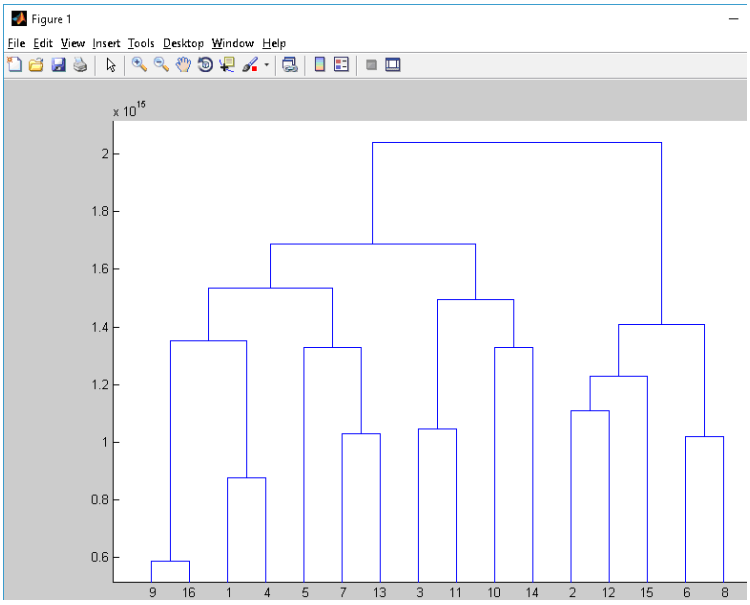

Рисунок 2.3 – Результат створеної дендограми виконання кластеризації даних за алгоритмом середнього зв’язку

Як можна побачити з наведених вище дендограм алгоритм найближчого сусіда виконується швидче ніж алгоритм середнього зв’язку та потребує здійснення меншої кількісті обчислювальних ітерації, однак результати кластеризації є більш гладкими та прийнятними. Зокрема, 3-й набір даних у першому випадку об’єднується з 9 та 16 до одного кластеру на 6-й ітерації. Однак, при виконанні кластеризації за алгоритмом середнього зв’язку це

здійснюється на 14-й ітерації, коли дані набори вже є складовими інших кластерів.

Результат створеної дендограми виконання кластеризації даних за алгоритмом дальнього сусіда наведено на рис.2.4. Результат створеної дендограми виконання кластеризації даних за алгоритмом центроїду наведено на рис.2.5.

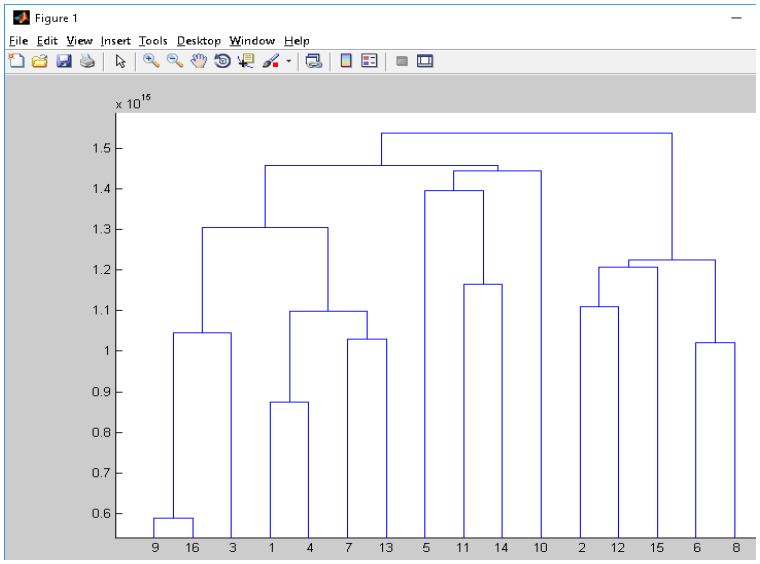

Рисунок 2.4 – Результат створеної дендограми виконання кластеризації даних за алгоритмом дальнього сусіда

Результат створеної дендограми виконання кластеризації даних за покроковим алгоритмом наведено на рис.2.6.

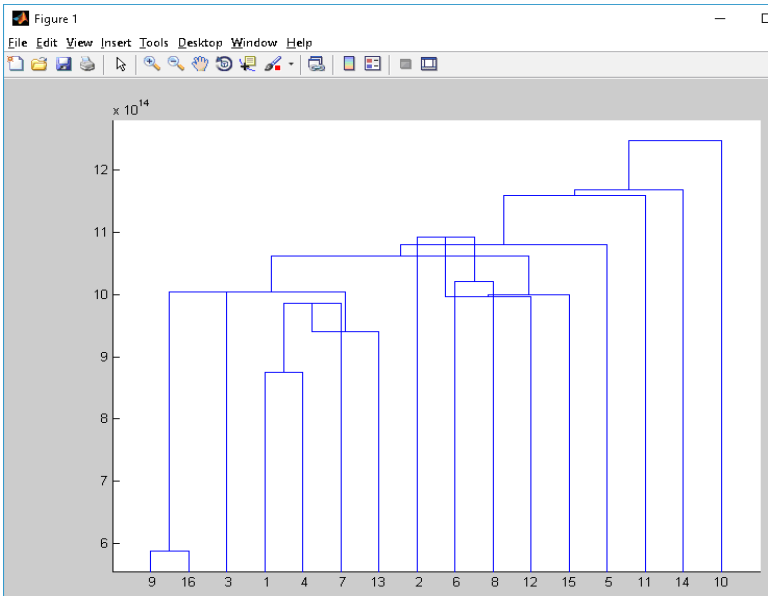

Рисунок 2.5 – Результат створеної дендограми виконання кластеризації даних за алгоритмом центроїду

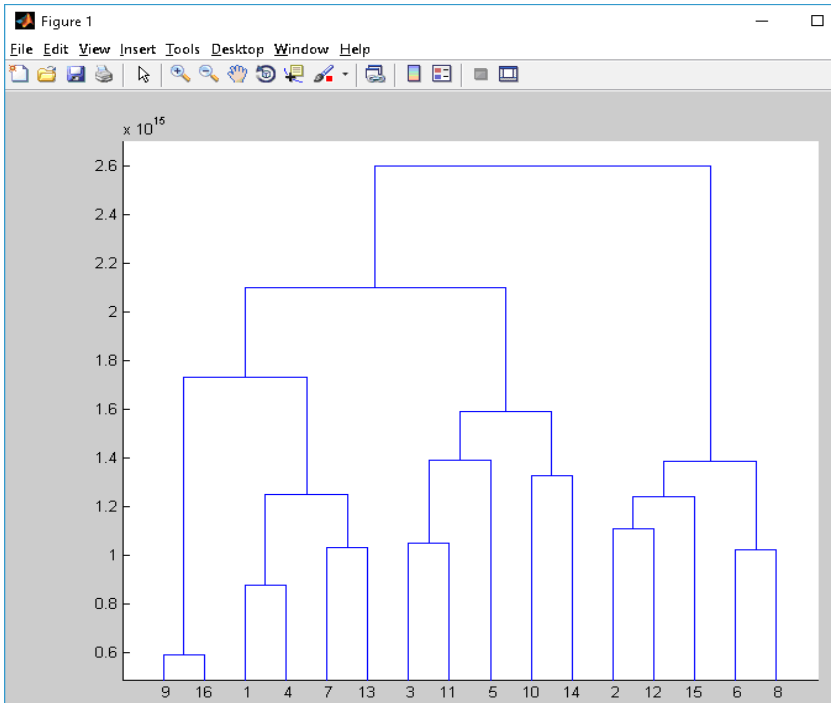

Рисунок 2.6 – Результат створеної дендограми виконання кластеризації даних за покроковим алгоритмом

Аналізуючи результати кластеризації даних за алгоритмами дальнього сусіда, центроїда та покроковим чином слід зазначити, що найбільш прийнятним з них є перший. Це зумовлено більш чітким позначенням виявлених кластерів, з іншого боку цей алгоритм потребує більше ітерацій розрахунку, ніж інші методи. Найгіршим чином кластеризація виконується за алгоритмом центроїду, що може бути зумовлено більшою похибкою при виконанні розрахунку, при цьому його швидкість виконання є досить високою.

Програмно підтримується можливість зберегти результати кластеризації даних до файлів наступних форматів: *.fig (бінарного файлу форми), *.ai (для подальшої графічної обробки засобами продуктів Adobe), *.bmp, *,eps, *,emf (графічний формат збереження даних у MS Windows), *,jpg, *.pcx (для обробки у системі Paintbrush), *.tif та інших.

2.4.2 Розробка модуля програмного забезпечення здійснення неієрархічного кластерного аналізу даних

У разі переходу користувачем з головної форми до вікна забезпечення неієрархічного КА даних необхідно забезпечити інтерфейс користувача такої форми. Для цього розроблена форма модулю здійснення неієрархічної кластеризації, яка наведена в додатку 3, рис.5. В рамках даного модуля передбачено використання трьох алгоритмів: карт Коххонена, k-means та c-means.

Користувач також може переглянути стислу інформацію по кожному з реалізованих у даному модулю алгоритмів КА даних. Після імпорту та обробки вхідних даних можна відобразити вхідну вибірку та результуючий розподіл об’єктів за кластерами з зазначенням їх центрів у окремих вікнах для наочного порівняння.

Для перевірки роботи даного алгоритму завдамо початкову кількість кластерів рівною 2.

Приклад візуалізації вхідних даних для кластеризації за алгоритмом k-means наведено на рис.2.7.

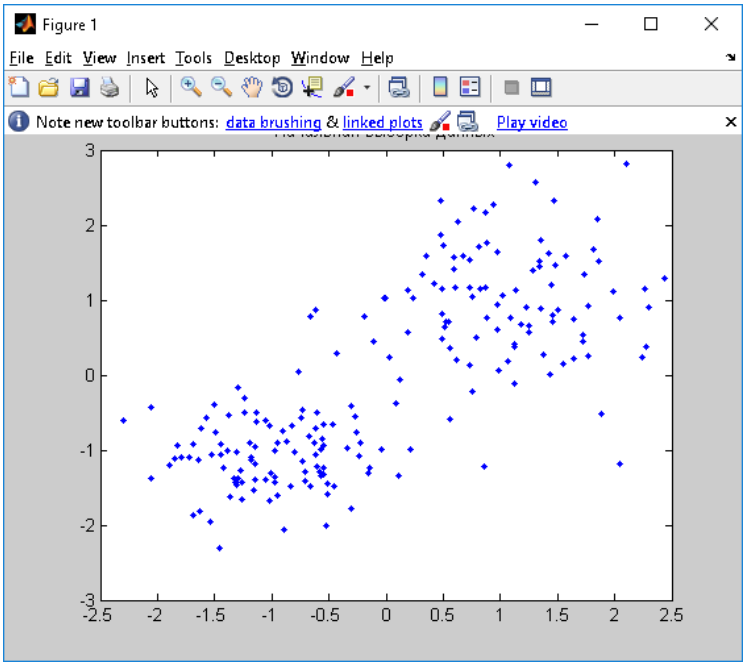

Рисунок 2.7 – Приклад візуалізації вхідних даних для кластеризації за алгоритмом k-means

Результат виконання кластеризації введеного масиву даних за алгоритмом k-means наведено на рис.2.8. Як можна побачити виконання розрахунків дозволило встановити центри відповідних кластерів та виключити найбільш видалені викиди. На базі виконання даного процесу кластеризації можна додатково дослідити кореляцію між обробленими даними за іншими алгоритмами чи розширити їх опис.

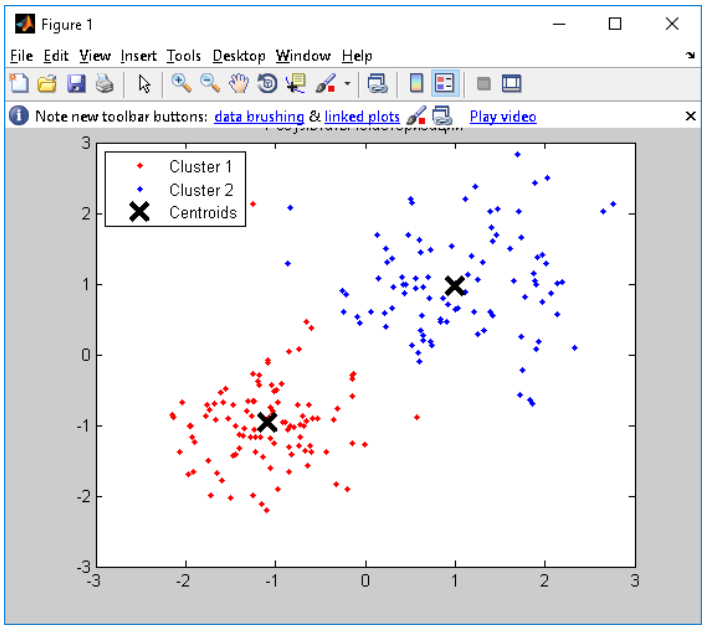

Рисунок 2.8 – Результат виконання кластеризації введеного масиву даних за алгоритмом k-means

Також, для іншої вхідної вибірки даних проведемо динамічну кластеризацію за алгоритмом k-means, у якій кожен крок візуалізується графічно кожні 0,5 секунди, а окремі кластери пофарбовані у різні кольори, що дає більший ступінь наочності здійснення обчислювальних ітерацій. Центри кожного з кластерів відображаються у вигляді заповнених чорних квадратів.

Приклад виконання різнокольорової динамічної кластеризації введеного масиву даних за алгоритмом k-means наведено на рис.2.9.

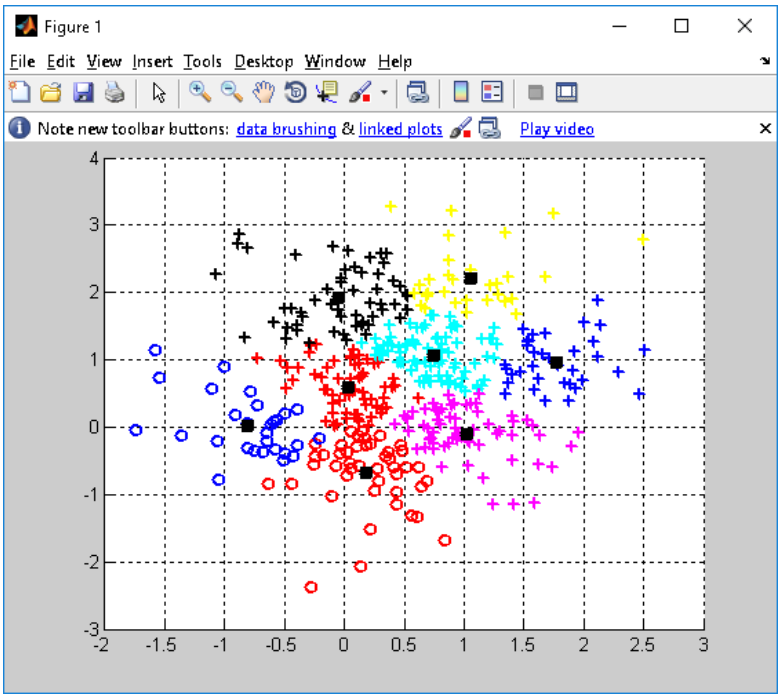

Рисунок 2.9 – Приклад виконання різнокольорової динамічної кластеризації введеного масиву даних за алгоритмом k-means

У разі обирання користувачем алгоритму c-means для здійснення КА даних розроблене ПЗ надає можливості перегляду довідкової інформації, алгоритму роботи даного методу та відображає вікно завдання налаштувань та візуалізації результатів. Це зумовлене особливістю методу, яка полягає у використанні нечіткої матриці приналежності U з елементами u_{ij}, визначальними приналежність і-го елемента вихідної безлічі векторів - j-му кластеру. Кластери описуються своїми центрами c_j - векторами того ж простору, якому належить вихідна безліч векторів. Фрагмент інтерфейсу форми завдання параметрів кластеризації за алгоритмом c-means наведено в додатку 3 на рис.6. Можна обрати один з імпортованих наборів даних (Data Set), обрати кількість вихідних кластерів, до яких слід віднести об’єкти вибірки, вказати порядок експоненти, максимальну кількість ітерацій та мінімальне значення поліпшення. Після цього необхідно натиснути на кнопку «Start».

Приклад візуалізації результатів виконання кластеризації за алгоритмом c-means наведено на рис.2.10.

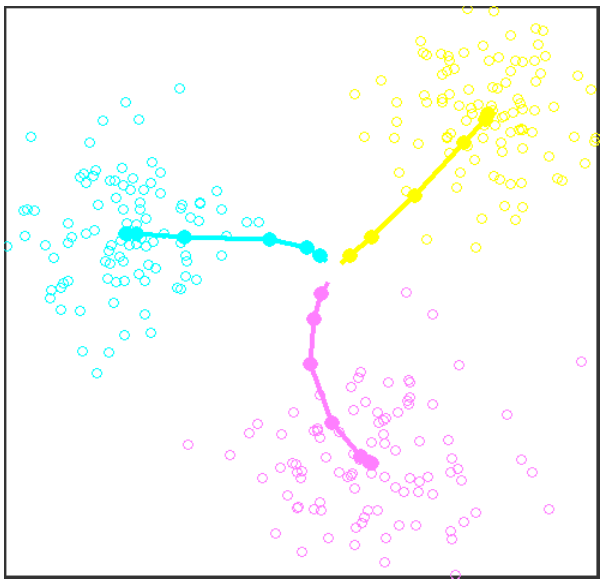

Рисунок 2.10 – Приклад візуалізації результатів виконання кластеризації за алгоритмом c-means

Як можна побачити множина об’єктів була розділена на 3 окремих кластери, кожен з яких виділений окремим кольором та позначено шлях до відносних центрів кожного з виявлених кластерів. Після виконання розрахунків можна обрати окремий відносний центр та побудувати для даного кластеру тривимірну поверхню нечіткої функції приналежності, у якості вхідних аргументів якої використовуються координати об’єктів у просторі станів.

Приклад побудови тривимірної поверхні нечіткої функції приналежності обраного кластеру за алгоритмом c-means наведено на рис.2.11.

У разі, якщо користувач використовує алгоритм карт Коххонена, він маж можливість здійснити імпорт даних завдяки використанню модуля Neutral Network Clustering Tool. Для цього необхідно обрати у полі Inputs потрібний набір даних.

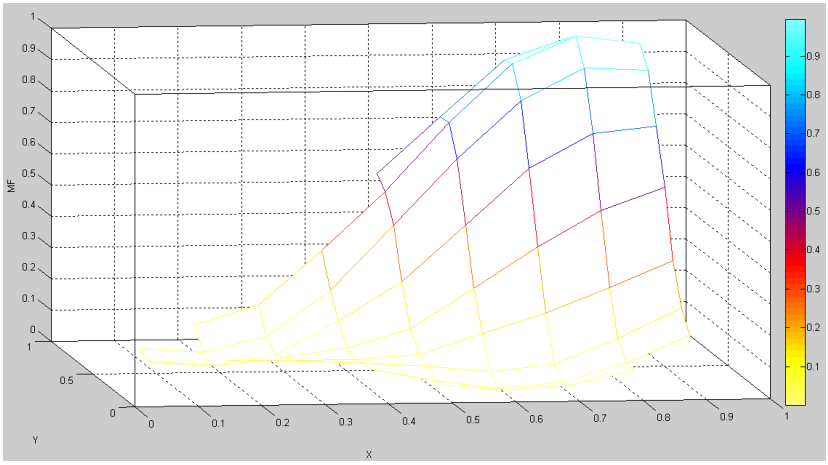

Рисунок 2.11 – Приклад побудови тривимірної поверхні нечіткої функції приналежності обраного кластеру за алгоритмом c-means

Вікно конфігурації розмірності карти Коххонена та відображення структури нейромережі наведено на рис.2.12. Для побудови карти можна ввести її розмірність, наприклад, оберемо 10 на 10 елементів. У нижній частині вікна відобразиться структура мережі, у якій зазначено кількість вхідних векторів (у нашому тестовому випадку їх 4), шар безпосередньо самої карти та кількість вихідних векторів даних (у нашому тестовому випадку їх 100).

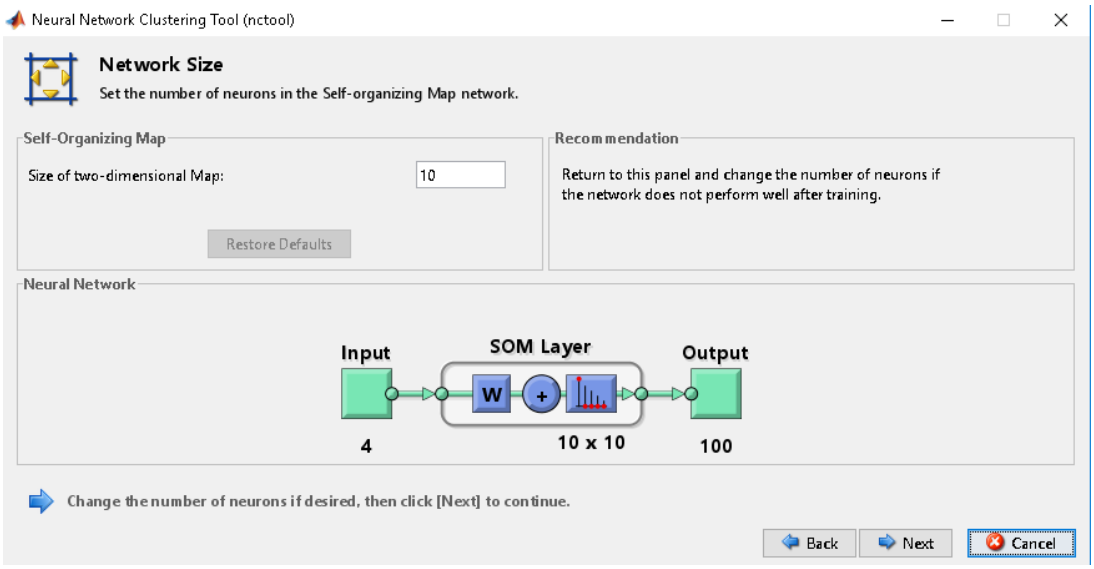

Рисунок 2.12 – Вікно конфігурації розмірності карти Кохонена та відображення структури нейромережі

Після виконання усіх попередніх налаштувань здійснюється процесс навчання нейромережі, у тестовому випадку для цього знадобилося 200 ітерацій. Форма перегляду результатів кластеризації за алгоритмом карт Коххонена наведена на рис.2.13. Карти складаються з конкуруючого шару, який

може класифікувати безліч векторів з будь-яким числом вимірювань на кількість класів, що дорівнює кількості нейронів у шарі. Навчання мережі складається з трьох основних процесів: конкуренція, кооперація і адаптація. Знаходимо найбільш підходящий (який переміг нейрон) на кожному кроці, використовуючи критерій мінімуму Евклидова відстані. Для даного алгоритму характерно зменшення топологічної околиці в процесі навчання.

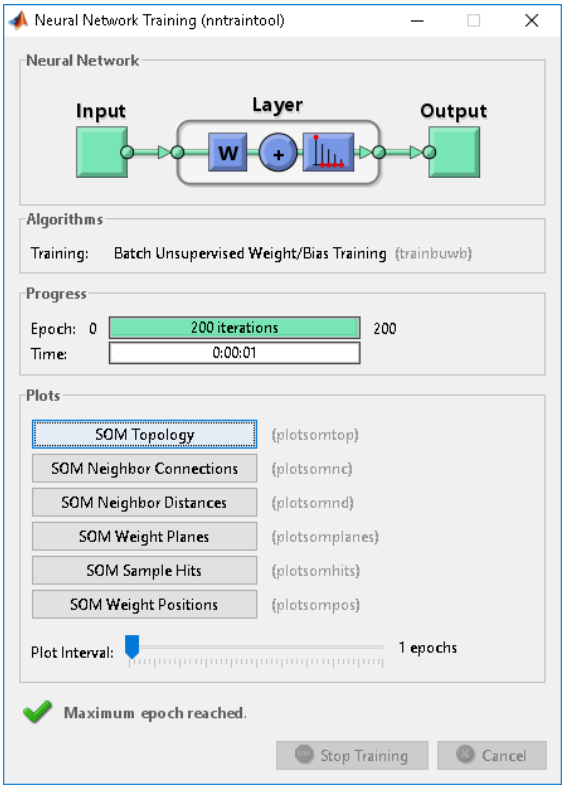

Рисунок 2.13 – Форма перегляду результатів кластеризації за алгоритмом карт Коххонена

Топологічна околиця повинна бути симетричною відносно точки максимуму, що є латеральним відстанню між переможцем і сусідніми нейронами.

Шляхом натискання на активні кнопки: SOM Topology, SOM Neighbor Connections, SOM Neighbor Distances, SOM Weight Planes, SOM Sample Hits, SOM Weight Positions користувач може переглянути відповідні візуальні представлення отриманих результатів. Зокрема, можна переглянути структуру карти та форму її складових. У даному випадку реалізовано підтримку шестикутника у якості базового елементу карти.

Побудована структура топології створеної карти Коххонена для вхідної вибірки даних наведена на рис.2.14. Як можна побачити дана ката має розмір 10 на 10 клітин. Результат розподілу зв'язків між вузлами створеної карти Коххонена наведено на рис.2.15. Результат візуалізації розподілу ваг дистанцій між сусідніми вузлами карти наведено на рис.2.16.

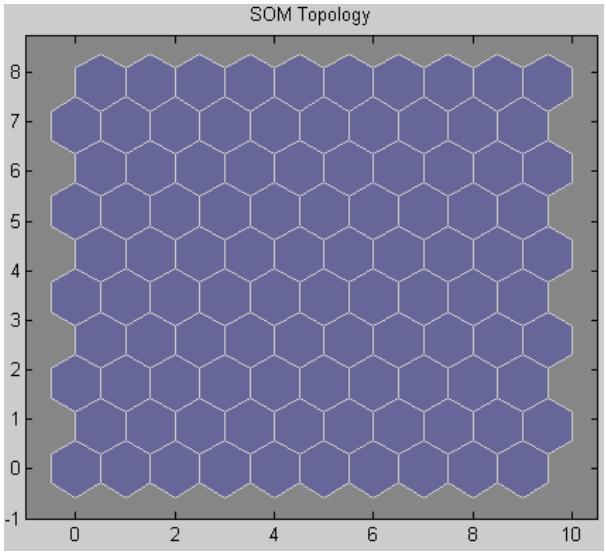

Рисунок 2.14 – Структура топології створеної карти Коххонена

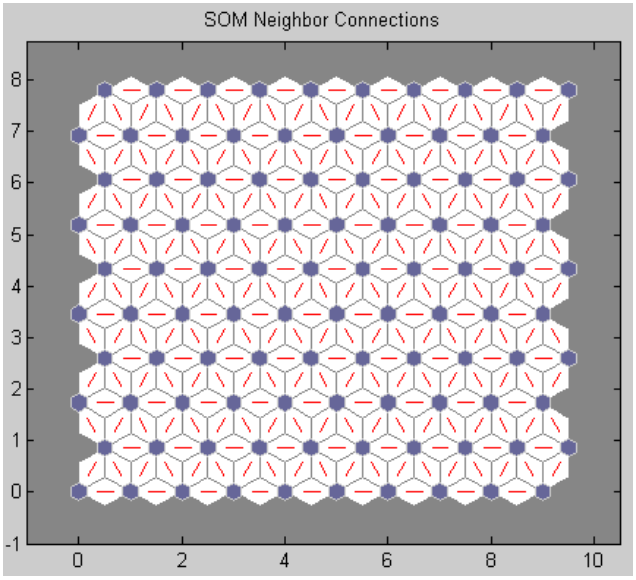

Рисунок 2.15 – Розподіл зв'язків між сусідніми нейронами створеної карти Коххонена

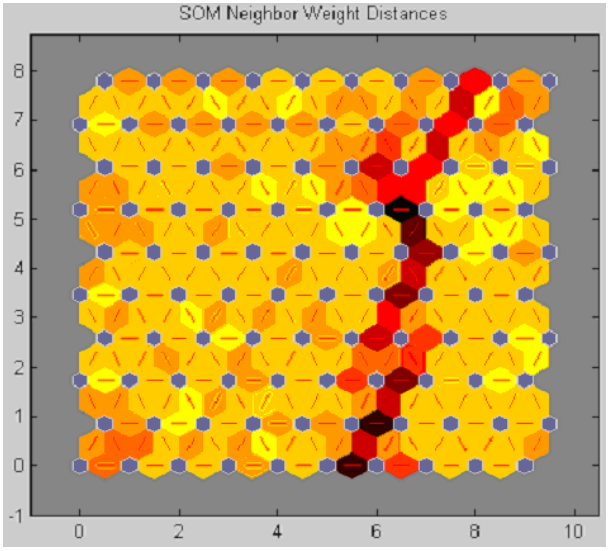

Рисунок 2.16 – Результат візуалізації розподілу ваг дистанцій між сусідніми вузлами карти

Даний розподіл дозволяє встановити порядок взаємодії окремих нейронів у створеній карті для визначення специфіки руху. Сусідні комірки виділено від чорного кольору до жовтого для візуалізації ступеня близькості вагового вектора штучного нейрона мережі до сусідів на карті.

Група вертикальних сегментів з'являється у верхньому правому краї, обмежена деякими більш темними сегментами. Ця група вказує на те, що мережа об'єднала дані в дві групи. Нижній правий регіон містить невелику групу тісно кластеризованих точок даних. Відповідні маси розташовані ближче один до одного в цьому регіоні, що позначається світлими кольорами на сусідніх комірках. Сегменти в правому верхньому регіоні темніші, ніж у верхньому лівому куті. Ця різниця кольорів вказує на те, що точки даних у цьому регіоні знаходяться далі один від одного. Ця відстань підтверджується показником вагових позицій. Результат відображення ступеню явності окремих вузлів побудованої карти наведено на рис.2.17.

Дана форма візуалізації дозволяє визначити кількість точок даних у просторі асоціюється з кожним окремим нейроном мережі Найкраще, якщо дані розподіляються досить рівномірно між нейронами. У цьому прикладі дані концентруються трохи більше в правих нейронах, але в цілому розподіл є досить рівним.

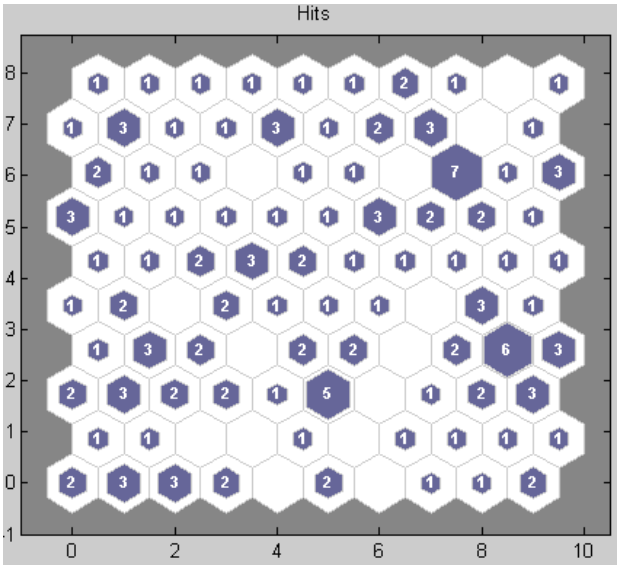

Рисунок 2.17 – Результат відображення ступеню явності окремих вузлів побудованої карти

Розподіл вагових позиції побудованої карти кластеризації наведено на рис.2.18. Дана візуалізація є необхідною для виявлення розташування точок даних та відповідних векторів ваги. Як було показано на рис.2.13 , лише після 200 ітерацій пакетного алгоритму створена карта є достатньо розподіленою за вхідним простіром.

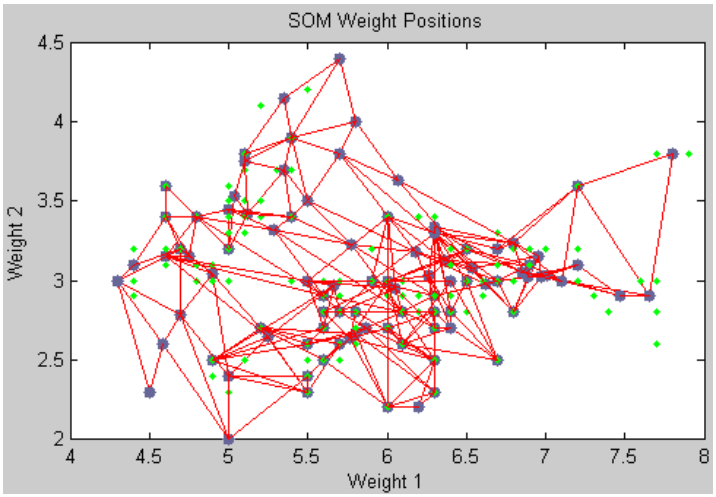

Рисунок 2.18 – Розподіл вагових позиції побудованої карти кластеризації

У результаті користувач має можливість зберегти отримані результати до окремого об’єкту робочого простору системи, зберегти та експортувати вхідні

та вихідні вектори даних, згенерувати окрему програмну структуру даних. Для цього передбачено текстові поля вводу даних та кнопку «Save Results». Генерація програмного коду нейромережі можливо завдяки натисканню на кнопку «Generate M-File» Форма збереження отриманих результатів проведення кластеризації даних на базі карт Коххонена наведена в додатку 3 на рис.7.

Для забезпечення можливості портування результатів розробленої карти та її подальшого дослідження у стимуляційному режимі була розроблена модель нейронної мережі кластеризації даних на базі карт Коххонена у Simulink, загальний вигляд якої наведено на рис.2.19.

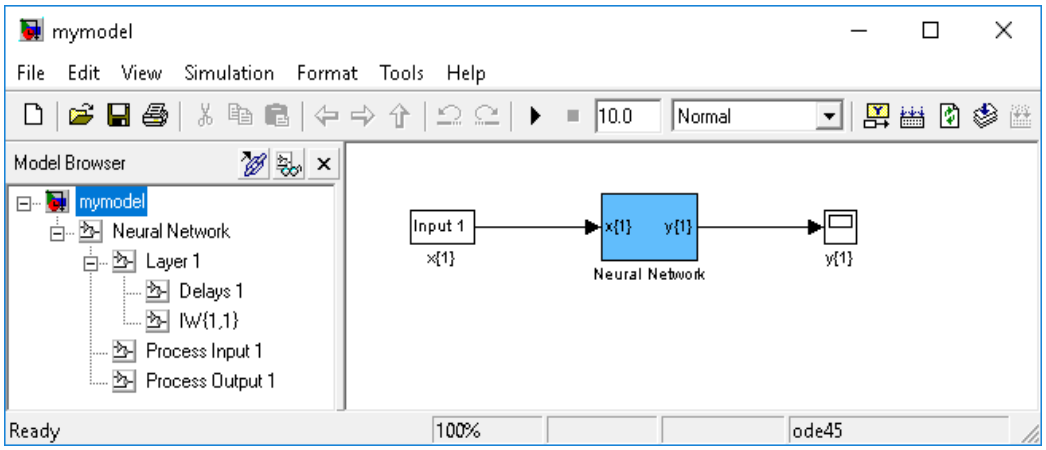

Рисунок 2.19 – Розроблена модель нейронної мережі кластеризації даних на базі карт Коххонена у Simulink

Для проведення тестування моделі двічі на левом значку (з надписом Input 1), що приведе до відкриття діалогового вікна параметрів блоку. У цьому випадку блок Input 1 є стандартним блоком завдання константи. Вхідний вектор даних (x1) уявлений у моделі завдяки блоку Input, нейромережа надана у блоку Neural Network, вихідний сигнал можна побачити у елементі Scope (осцилограф перегляду відповідного вихідного імпульсу). Конфігурація параметрів вхідного блоку створеної моделі нейронної мережі наведена в додатку 3 на рис.8.

Значення константи виражено у вигляді масиву з 5 тестових значень, які інтерпретовано в одномірному режимі. Час моделювання встановлений за замовчуванням (inf-безліч). Двічі клацаючи на блоці Neural Network, а потім на

блоці Layer 1, можна отримати детальну графічну інформацію про структуру мережі. Склад елементів контейнера Neural Network розробленої моделі кластеризації наведено на рис.2.20.

Модель має один шар з кількістю нейроніх, яка дорівнює розмірності топології карти. Склад вхідних (Process Input) та вихідних (Process Output) елементів обробки контейнера Neural Network наведено на рис.2.21. Структура внутрішнього шару моделі створеної нейромережі наведено на рис.2.22.

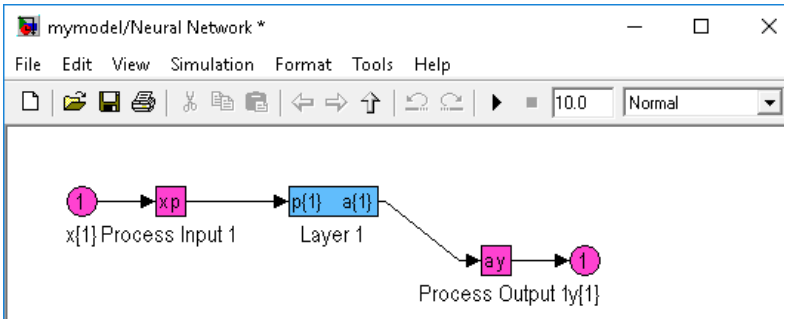

Рисунок 2.20 – Склад елементів контейнера Neural Network розробленої моделі кластеризації

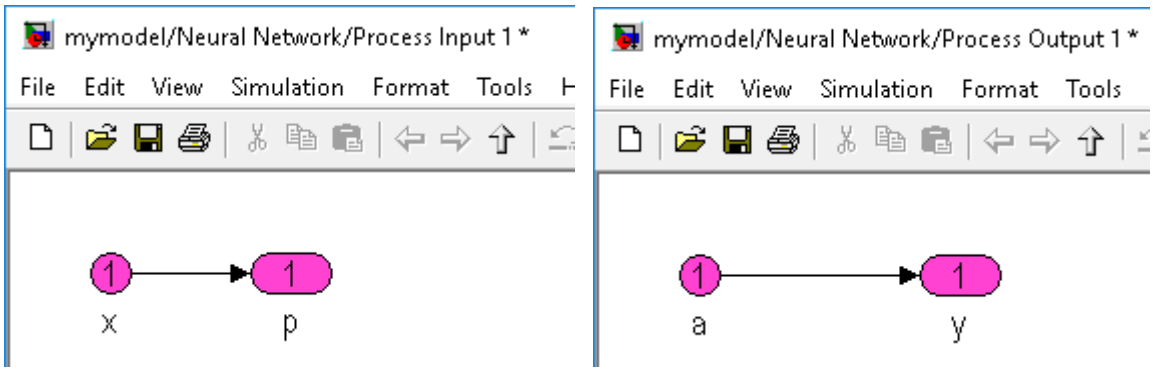

Рисунок 2.21 – Склад вхідних та вихідних елементів обробки контейнера Neural Network

Слід, також, зазначити, що при завданні конкретних цифрових значень в Simulink вектори необхідно представити як стовпці.

Основна функція для формування нейросетевих моделей в Simulink є функція gensim. Конфігурація створеного параметрів блоку суматора netsum наведена на рис.2.23.

Можна побачити, що параметр Gain встановлено рівним 1, мультиплікація виконується за вказаною формулою а час моделювання встановлений нескінченним.

Фрагмент моделі обробки вагових коефіцієнтів нейронної мережі наведено на рис.2.24. Вхідний вектор поступає до відповідних елементів формування ваг, які у свою чергу надають вихідні значення перетворювачам (negdist), з яких отриманий сигнал надходить до підсумкового сумматора (Mux). З створеної мережею можна проводити різні експерименти, можливі в середовищі Simulink, за допомогою команди gensim здійснюється інтеграція створених нейромереж в блок-діаграми цього пакета з використанням наявних при цьому інструментів моделювання різних систем.

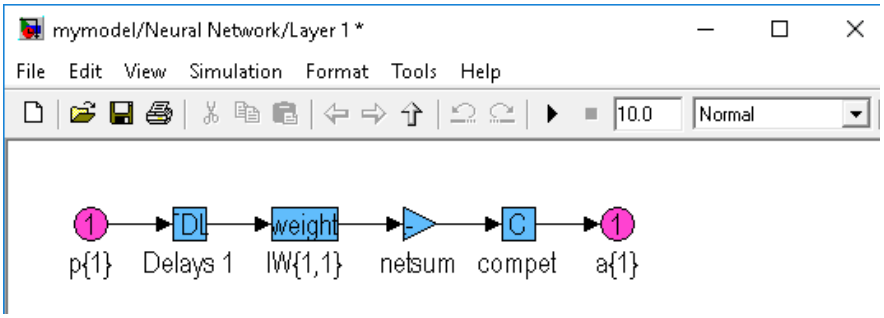

Рисунок 2.22 – Структура внутрішнього шару моделі створеної нейромережі

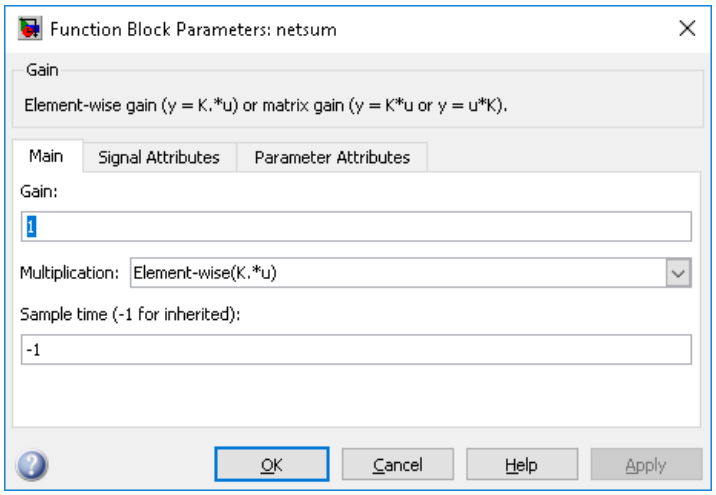

Рисунок 2.23 – Конфігурація параметрів блоку суматора netsum

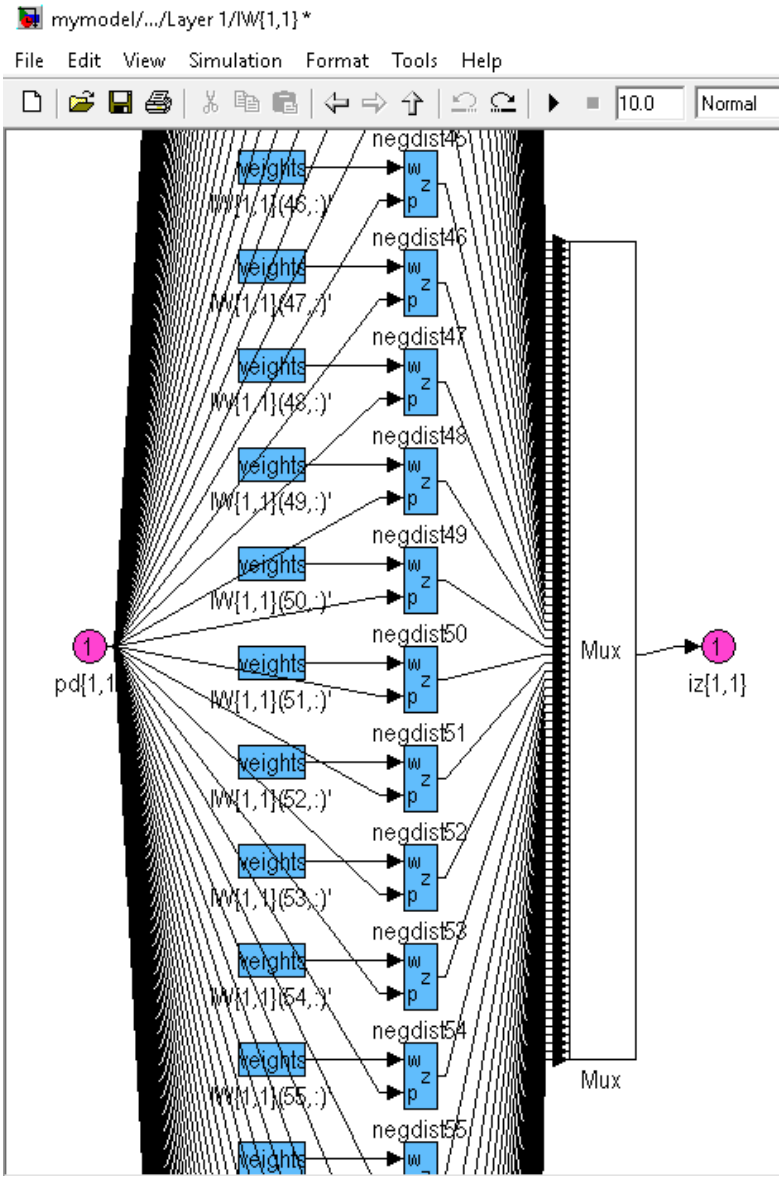

Рисунок 2.24 – Фрагмент моделі обробки вагових коефіцієнтів нейронної мережі

Форма конфігурації параметрів та налаштувань розробленої моделі нейромережі наведена в додатку 3 на рис.9.

Типом вирішувача є Variable-step, у якості функції використано ode45 (вираз Доманда-Принса), алгоритм (налаштування нульового перетинання) встановлений не адаптивний.

ВИСНОВКИ

В рамках виконання науково-дослідницької роботи виконано усі поставлені завдання, мету роботи досягнуто в повному обсязі.

Проаналізовано поняття та основні особливості кластерного аналізу даних, розглянуто основні алгоритми кластеризації, проведено аналіз існуючих програмних продуктів кластерного аналізу, описані переваги використаних засобів розробки.

Розроблена ментальна карта процесу виконання дослідницької роботи, створено UML діаграми проекту програмного забезпечення, формалізовано алгоритми виконання процесу кластеризації, здійснено програмну реалізацію та опис процесу виконання кластерного аналізу даних.

Розроблено модулі програмного забезпечення здійснення ієрархічного агломеративного та неієрархічного кластерного аналізу даних.

Розроблене програмне забезпечення може бути використано для виявлення кластерної структури даних з метою спрощення їх подальшої обробки та аналізу при прийнятті управляючих рішень. Подальшим розвитком програмного забезпечення може бути впровадження нових алгоритмів кластерного аналізу об’єктів та метрик оцінки достовірності роботи кожного методу кластеризації.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Айвазян С.А. Прикладная статистика: классификация и снижение размерности / С.А. Айвазян, В.М. Бухштабер, И.С. Енюков. – М.: Финансы и статистика, 1999. – 641 с.

2. Демидова Л.А. Методы кластеризации в задачах оценки технического состояния зданий и сооружений в условиях неопределенности / Л.А. Демидова, Е.И. Коняева. – М.: Горячая Линия - Телеком, 2012. – 156 с.

3. Большаков Н.М. Кластеризация в современном образовании: методология и практика / Н.М. Большаков, В.В. Жиделева. – СПб: СПбГЛТУ, 2016. – 200 с.

4. Солондаев В.К. Вспользование функции кластеризации / В.К. Солондаев. – Ярославль: ЯрГУ им. П. Г. Демидова, 2013. – 111 с.

5. Волошко А.В. Кластеризация информационных сигналов / А.В. Волошко, Т.Н. Лутчин // VIII Всеукраинской научно – технической конференции. – Кременчуг, КДУ. – 2010. – С. 421-423.

6. Потапов А.С. Технологии искусственного интеллекта / А.С. Потапов. – СПб.: СПбГУ ИТМО, 2010. – 218 с.

7. Дюран Б. Кластерный анализ / Б. Дюран, П. Оделл. – М.: Статистика, 1997. – 328 с.

8. Барсегян А.А. Методы и модели анализа данных: OLAP и Data Mining / А.А. Барсегян, М.С. Куприянов, В.В. Степаненко. – СПб.: БХВ-Петербург, 2004. – 336 с.

9. Боровиков В.П. Популярное введение в современный анализ данных в системе STATISTICA / В.П. Боровиков. – М.: Горячая линия-Телеком, 2013. – 288 с.

10.Дюк В.А. Data Mining / В.А. Дюк, А.П. Самойленко. – СПб: Питер, 2001. – 368 с.

11.Загоруйко Н.Г. Прикладные методы анализа данных и знаний / Н.Г. Загоруйко. – Новосибирск: ИМ СО РАН, 2009. – 270 с.

12.Замятин А.В. Интеллектуальный анализ данных / А.В. Замятин. – Томск: Издательский Дом Томского государственного университета, 2016. – 320 с.

13.Миркин Б.Г. Введение в анализ даннях / Б.Г. Миркин. – М.:Юрайт, 2015. – 174 с.

14.Рафалович В. Data mining, или Интеллектуальный анализ данных для занятих / В. Рафалович. – М.: СмартБук, 2014. – 110 с.

15.Ситник В.Ф., Краснюк М.Т. Інтелектуальний аналіз даних / В.Ф. Ситник, М.Т. Краснюк. – Київ: КНЕУ, 2007. – 376 с.

16.Журавлев Ю.И. Распознавание. Математические методы. Программная система. Практические применения / Ю.И. Журавлев, В.В. Рязанов, О.В. Сенько. – М.: Фазис, 2006. – 383 с.

17.Мандель И.Д. Кластерный анализ / И.Д. Мандель. – М.: Финансы и статистика, 1998. – 262 с.

18.Граничин О.Н. Исследование и рандомизация алгоритмов устойчивой кластеризации на основе индексов / О.Н. Граничин, Д.С. Шалымов // Нейрокомпьютеры: разработка, применение. – 2009. – № 3. – С. 58–64.

19.Calinski R.B. A dendrite method for cluster analysis / R.B. Calinski, J. Harabasz // Communications in Statistics. – vol. 3. – 1974. – PP.1–27.

20.Krzanowski W.J. A criterion for determining the number of clusters in a data set using sum of squares clustering / W.J. Krzanowski, Y.T. Lai // Biometrics. – №44. – 1985. – PP.23–34.

21.Hastie T. The Elements of Statistical Learning: Data Mining, Inference, and Prediction / T. Hastie, R. Tibshirani, J. Friedman. – Springer-Verlag, 2009. – 746 p.

22. Murty J. Flynn Data clustering: a review / J. Murty Flynn // ACM Comput. Surv. – №31(3). – 1999. – PP.54-69.

23.Олдендерфер М.С. Кластерный анализ. Факторный, дискриминантный и кластерный анализ / М.С. Олдендерфер, Р.К. Блэшфилд. – М.: «Финансы и статистика», 2009. – 215 с.

24.Шуметов В. Г. Кластерный анализ: подход с применением ЭВМ / В.Г. Шуметов, Л.В. Шуметова. – ОрелГТУ, Орел, 2010. – 118 с.

25.Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из даннях / П. Флах. – М.: ДМК Пресс, 2015. – 400 с.

26.Чубукова И.А. Data Mining / И.А. Чубукова. – НОУ «Интуит», 2016. – 471 с.

27.Ярушкина Н.Г. Интеллектуальный анализ временных рядов / Н.Г. Ярушкина, Т.В. Афанасьева. – Ульяновск: УлГТУ, 2010. – 320 с.

28.Методы кластерного анализа. Иерархические методы. – [Електронний ресурс]. – Режим доступу: http://www.intuit.ru/studies/courses/6/6/lecture/182?page=1. – Дата доступу: 09.09.2017.

29.Обзор алгоритмов кластеризации данных. – [Електронний ресурс]. – Режим доступу: https://habrahabr.ru/post/101338/. – Дата доступу: 18.09.2017.

30.Кластеризация. Типы алгоритмов.Методы и средства анализа данных. – [Електронний ресурс]. – Режим доступу: http://bourabai.ru/tpoi/analysis6.htm. – Дата доступу: 19.09.2017.

31.Выбор процедуры кластеризации. Модуль Statistics Base. – [Електронний ресурс]. – Режим доступу: https://www.ibm.com/support/knowledgecenter/ru/SSLVMB_21.0.0/com.ibm.spss.statistics.help/cluster_choosing.htm. – Дата доступу: 20.09.2017.

32.Hierarchical Clustering Explorer for Interactive Exploration of Multidimensional Data. – [Електронний ресурс]. – Режим доступу: hhttp://www.cs.umd.edu/hcil/hce/index.html. – Дата доступу: 22.09.2017.

33.Обзор STATISTICA. – [Електронний ресурс]. – Режим доступу: http://statsoft.ru/products/overview/. – Дата доступу: 25.09.2017.

34.Дьяконов В. MATLAB. Специальный справочник / В.Дьяконов. – СПб.: Питер, 2002. – 528 с.

35.Герман-Галкин С.Г. Компьютерное моделирование систем в MATLAB / С.Г. Герман-Галкин. – СПб.: КОРОНА принт, 2001. – 320 с.

36.Курбатова Е.А. MATLAB. Самоучитель / Е.А. Курбатова. – СПб.: Вильямс, 2005. – 256 с.

37.Россум Г. Язык программирования Python. СПб.: Символ-Плюс Медиа, 2012. – 454 с.

38.Сузи Р.А. Язык программирования Python. М.: Интуит, 2016. – 351 с.

Додаток 1.

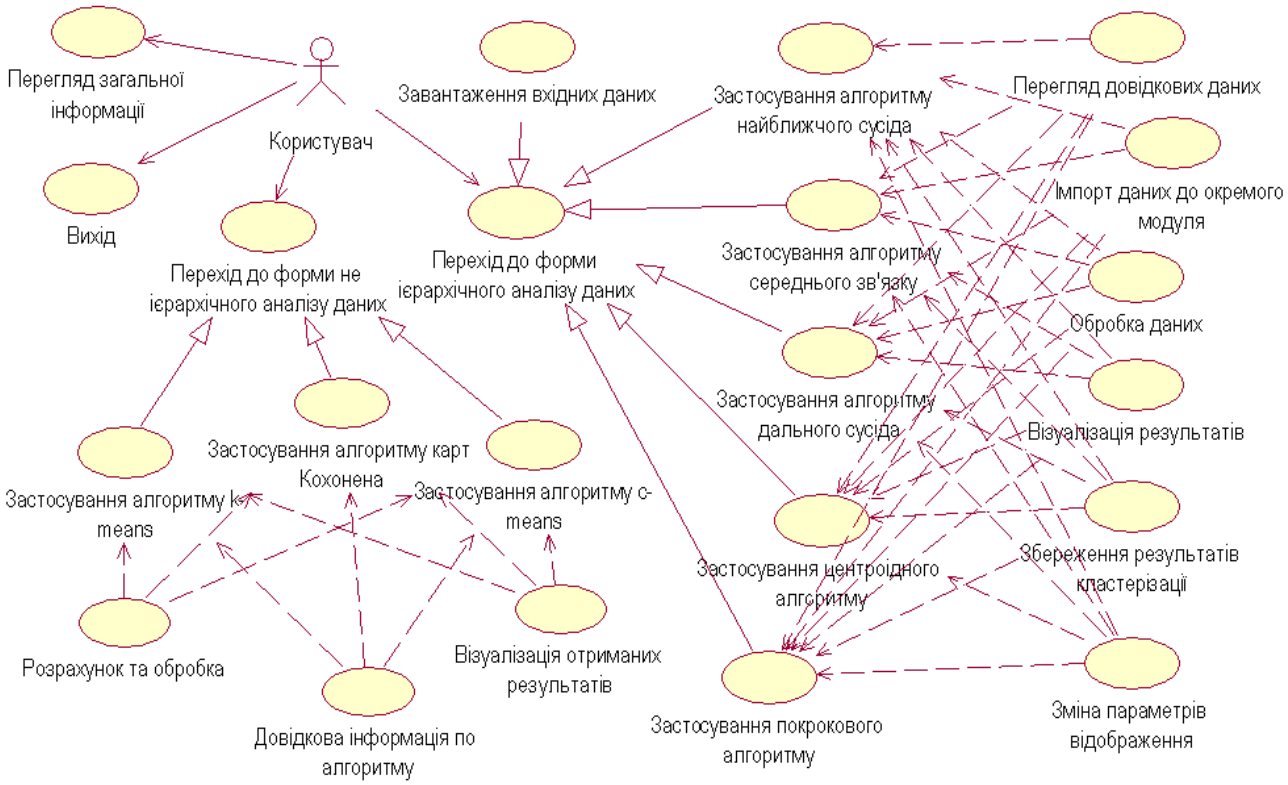

Рисунок 1 – Діаграма варіантів використання програмного забезпечення кластерного аналізу даних

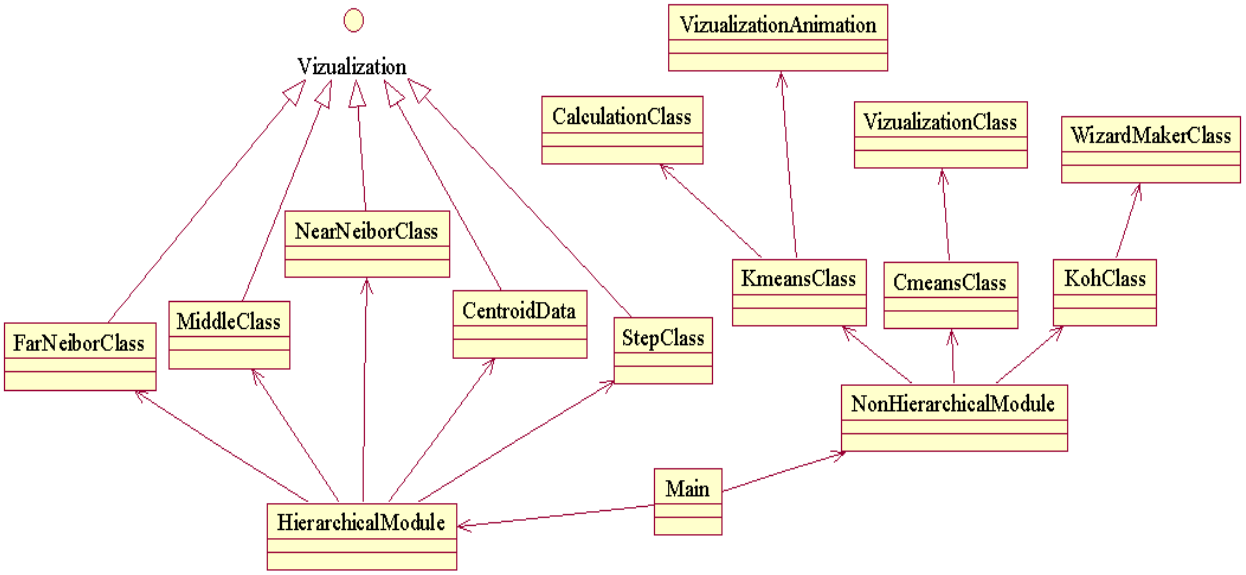

Рисунок 2 – Діаграма класів програмного забезпечення кластерного аналізу даних

Рисунок 3 – Діаграма послідовності дій виконання ієрархічної кластеризації

Додаток 2

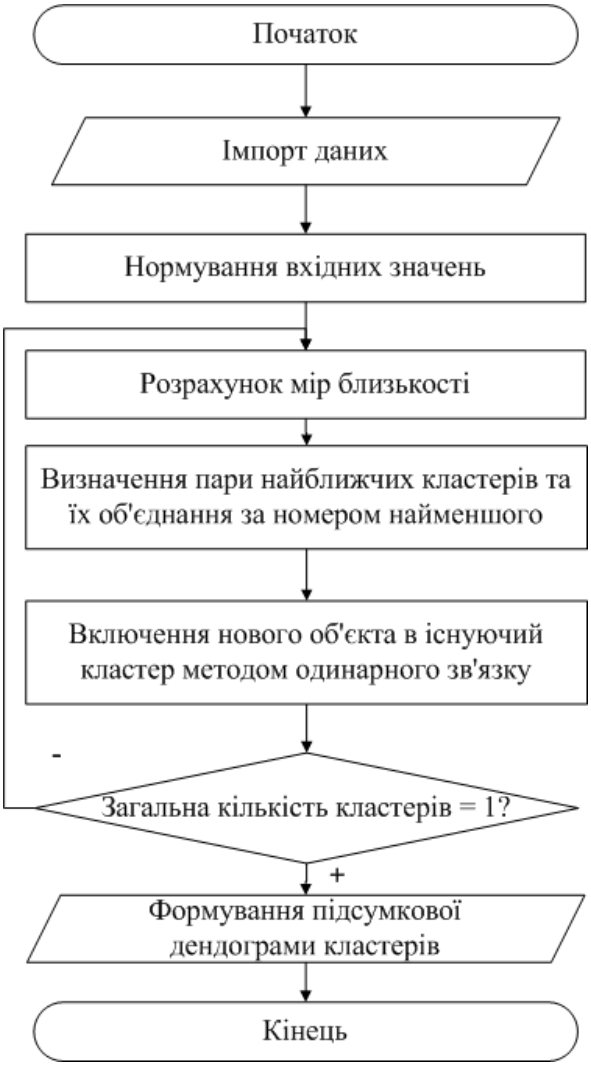

Рисунок 1 – Загальний алгоритм виконання ієрархічного агломеративного КА даних

Рисунок 2 – Загальний алгоритм виконання КА даних на базі карт Кохонена

Рисунок 3 – Загальний алгоритм виконання кластерного аналізу даних на базі методу k-середніх

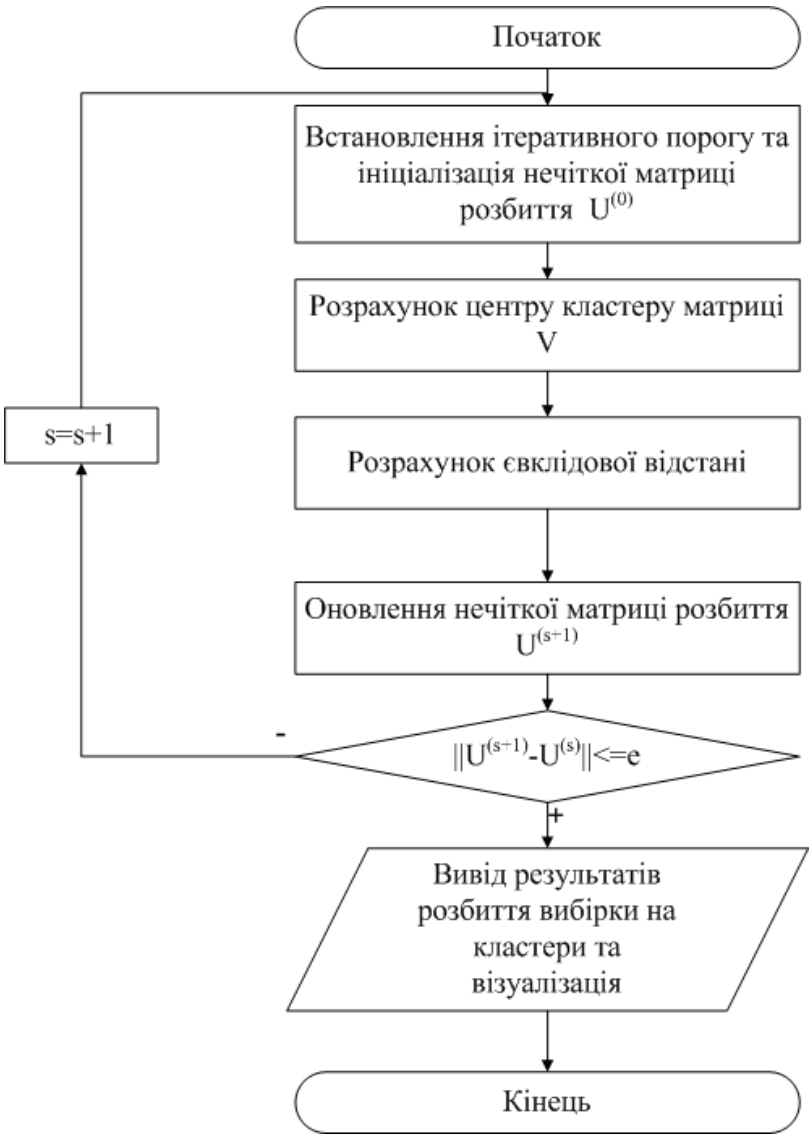

Рисунок 4 – Загальний алгоритм виконання кластерного аналізу даних на базі методу с-середніх

Додаток 3

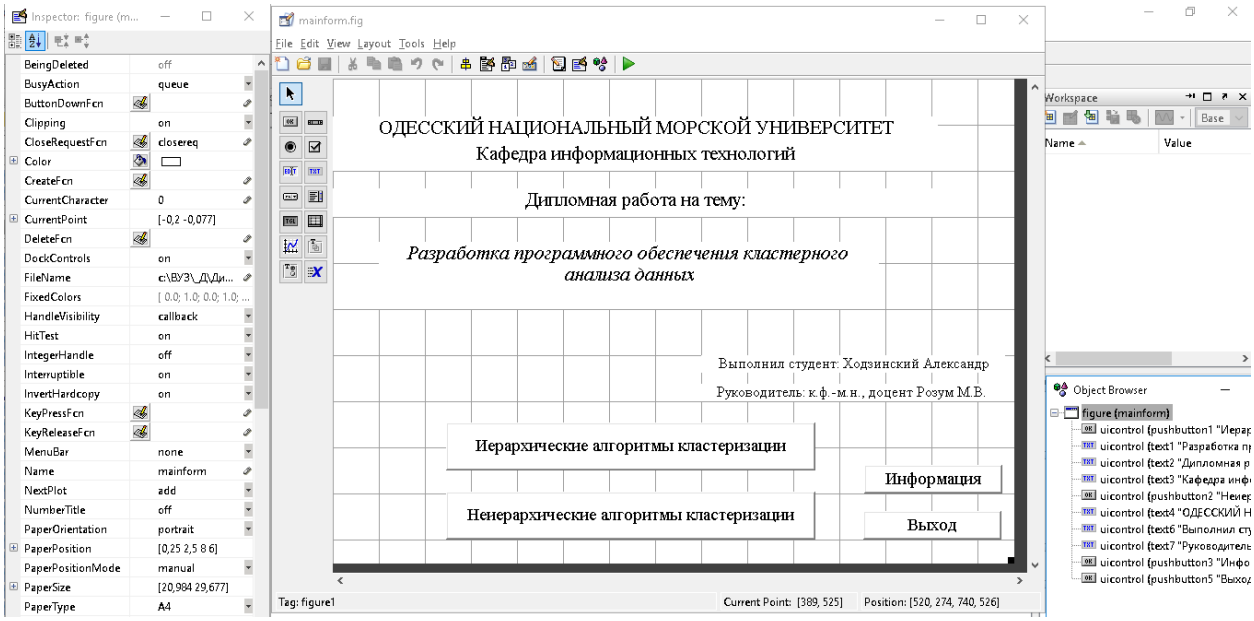

Рисунок 1 – Проект розробки інтерфейсу головної форми завдяки засобам фреймворку Guide

Рисунок 2. – Форма модулю здійснення ієрархічного агломеративного аналізу даних

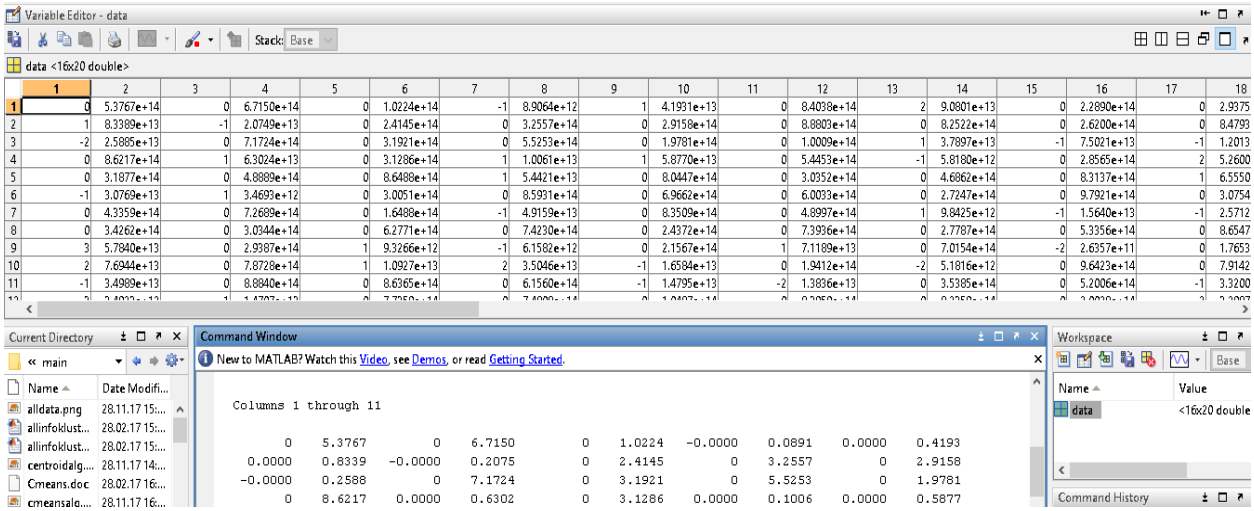

Рисунок 3 – Результат завантаження вхідних даних до системи Matlab

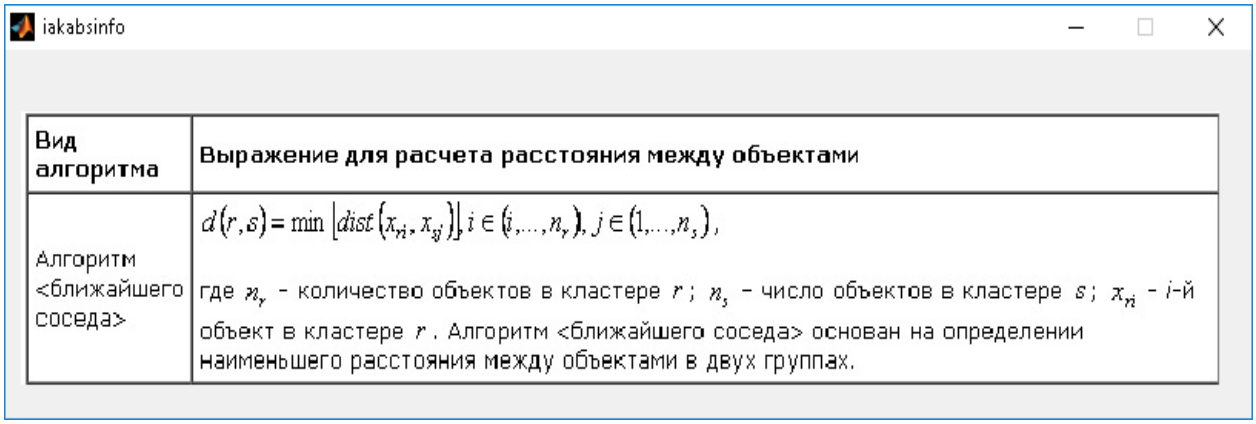

Рисунок 4 – Форма перегляду стислої довідкової інформації про дію алгоритму найближчого сусіда

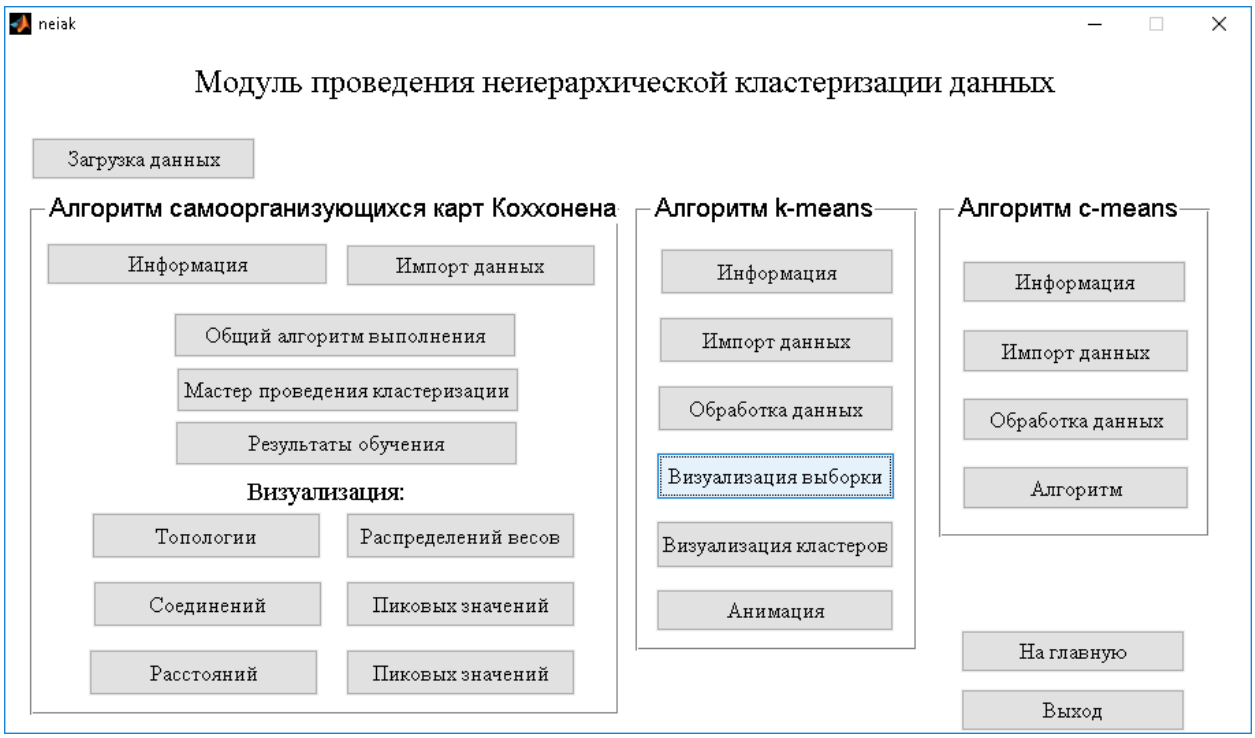

Рисунок 5 – Форма модулю здійснення не ієрархічного аналізу даних

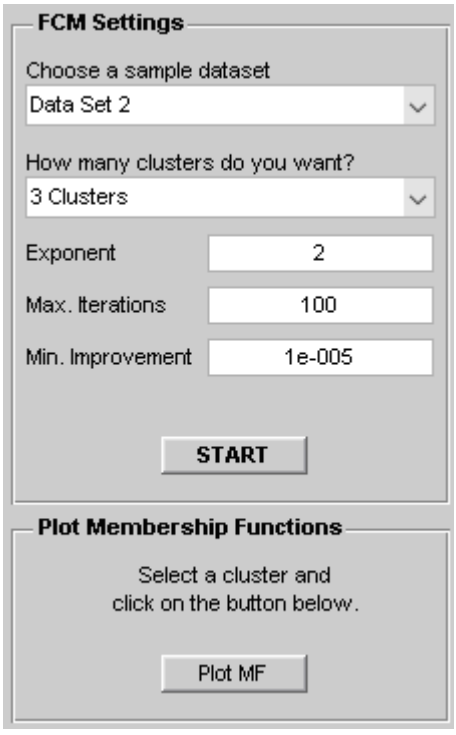

Рисунок 6 – Фрагмент форми завдання параметрів алгоритму c-means

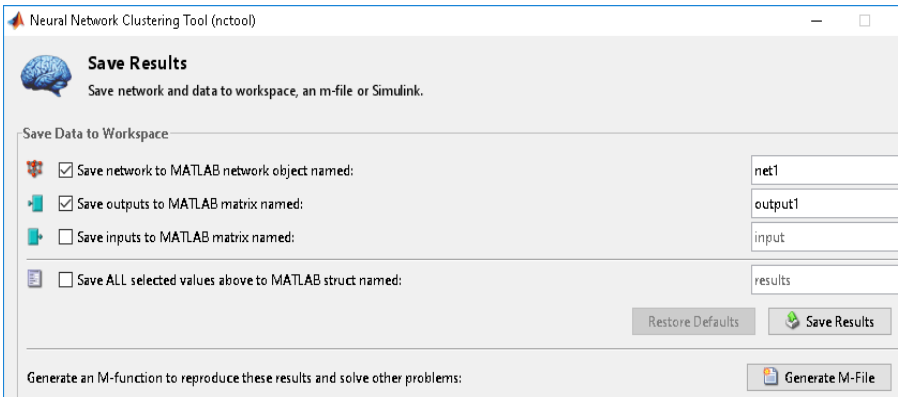

Рисунок 7 – Форма збереження отриманих результатів проведення кластеризації даних на базі карт Коххонена

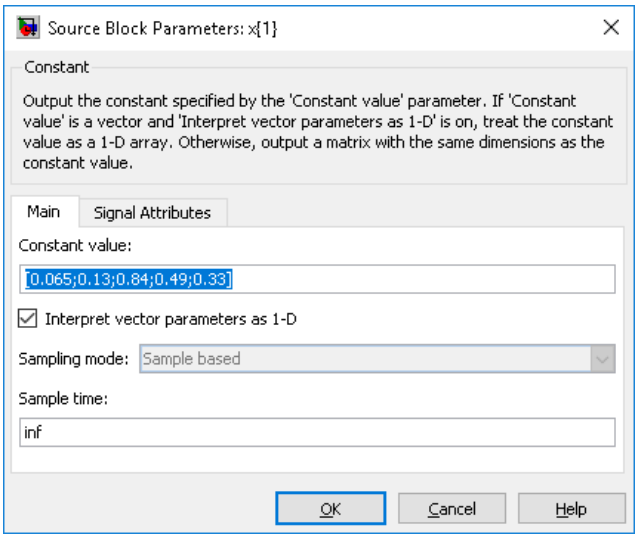

Рисунок 8 – Конфігурація параметрів вхідного блоку створеної моделі нейронної мережі

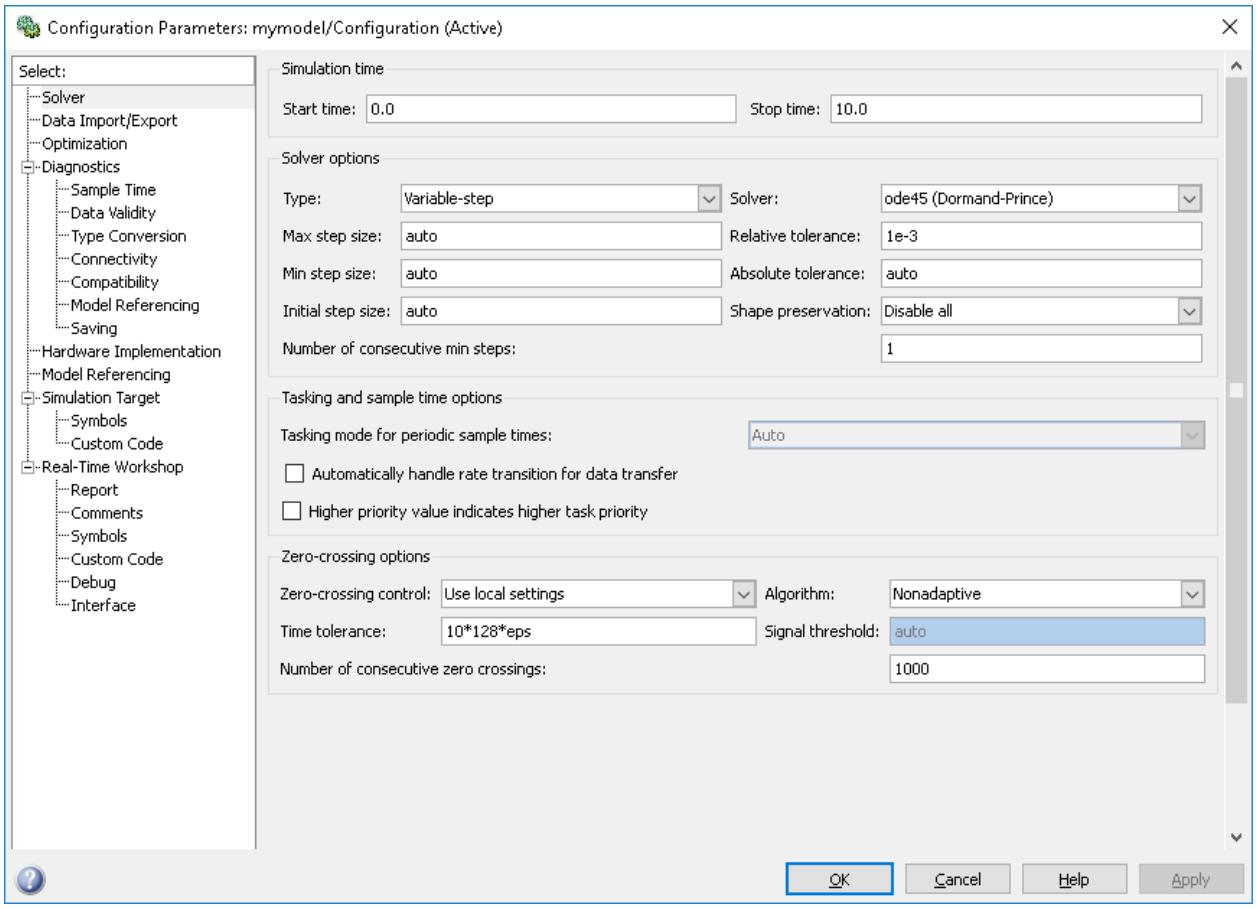

Рисунок 9 – Форма конфігурації параметрів та налаштувань розробленої моделі нейромережі