МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп’ютерної інженерії

Король Віталій Ростиславович

Програмний модуль кластеризації пасажиропотоків м. Тернополя / Software module for clustering passenger flows in Ternopil

спеціальність: 123 – Комп’ютерна інженерія
освітньо-професійна програма – Комп’ютерна інженерія

Кваліфікаційна робота

Виконав: студент групи КІ-41
Король Віталій Ростиславович

Науковий керівник
д.т.н., професор Березький О.М.

ТЕРНОПІЛЬ – 2026

РЕЗЮМЕ

Кваліфікаційна робота містить 97 сторінок пояснюючої записки, 6 рисунків, 16 таблиць, 4 додатки. Обсяг графічного матеріалу 2 аркуші формату А3.

Метою кваліфікаційної роботи є розробка програмного модуля кластеризації пасажиропотоків міста Тернополя.

В кваліфікаційній роботі шляхом розробки програмного модуля кластеризації зупинок за характером добового навантаження розв'язано актуальну задачу аналізу пасажиропотоків міського громадського транспорту. Виконано аналіз предметної області та огляд сучасних методів кластерного аналізу, які застосовуються для обробки транспортних і соціально-економічних даних. Розроблено математичну модель кластеризації пасажиропотоків, описано алгоритм k-means, методи вибору кількості кластерів та критерії оцінювання якості кластеризації, розроблено архітектуру програмного модуля з механізмом зворотного зв'язку для ітераційного підбору параметрів кластеризації, здійснено програмну реалізацію розробленого модуля кластеризації та проведено експериментальні дослідження на основі реальних даних.

Для реалізації програмного модуля використано мову програмування Python та бібліотеки pandas, NumPy, scikit-learn, matplotlib.

Подальший розвиток роботи може полягати у розширенні набору ознак, використанні часових рядів замість агрегованих показників, застосуванні альтернативних алгоритмів кластеризації та інтеграції модуля з геоінформаційними системами.

Ключові слова: АЛГОРИТМ, КЛАСТЕРНИЙ АНАЛІЗ, ПАСАЖИРОПОТОКИ, ПРОГРАМНИЙ МОДУЛЬ, ПАСАЖИРСЬКІ ЗУПИНКИ, k-MEANS.

RESUME

The qualification work contains 97 pages of explanatory notes, 6 figures, 16 tables, and 4 appendices. The amount of graphic material is 2 sheets in A3 format.

The aim of the qualification thesis is to develop a software module for clustering passenger flows of the city of Ternopil.

In the qualification work, by developing a software module for clustering stops by the nature of daily load, the current problem of analyzing passenger flows in urban public transport was solved. An analysis of the subject area and a review of modern clustering methods used for processing transport and socio-economic data were carried out. A mathematical model for passenger flow clustering was developed; the k-means algorithm, methods for selecting the number of clusters, and criteria for evaluating clustering quality were described. The architecture of the software module with a feedback mechanism for iterative selection of clustering parameters was designed. The developed clustering module was implemented in software, and experimental studies were conducted using real data.

The software module was implemented using the Python programming language and the libraries pandas, NumPy, scikit-learn, and matplotlib.

Further development of the work may include expanding the set of features, using time series instead of aggregated indicators, applying alternative clustering algorithms, and integrating the module with geographic information systems.

Keywords: ALGORITHM, CLUSTER ANALYSIS, PASSENGER FLOWS, SOFTWARE MODULE, TRANSPORT STOPS, K-MEANS.

ЗМІСТ

ВСТУП

Міський громадський транспорт є одним із ключових елементів інфраструктури сучасного міста, що забезпечує мобільність населення та впливає на соціально-економічний розвиток території. Ефективне функціонування транспортної системи значною мірою залежить від раціонального планування маршрутної мережі, розкладів руху та оптимального використання транспортних засобів, що, у свою чергу, потребує аналізу пасажиропотоків.

У практиці транспортного планування все частіше застосовуються методи аналізу даних і машинного навчання, зокрема методи кластерного аналізу, які дозволяють виявляти приховані закономірності у великих масивах даних. Кластеризація зупинок громадського транспорту за характером добового пасажиропотоку дає змогу групувати об’єкти з подібними властивостями та отримувати узагальнені типи зупинок, що спрощує процес прийняття управлінських рішень.

Актуальність даної роботи зумовлена необхідністю підвищення ефективності аналізу пасажиропотоків міського громадського транспорту з використанням сучасних інформаційних технологій. Використання програмних модулів кластеризації дозволяє автоматизувати процес обробки даних, зменшити суб’єктивність аналізу та забезпечити наочну інтерпретацію результатів.

Метою кваліфікаційної роботи є розробка програмного модуля кластеризації пасажиропотоків міста Тернополя на основі агрегованих даних щодо середньої кількості пасажирів на зупинках громадського транспорту.

Для досягнення поставленої мети у роботі необхідно розв’язати такі завдання:

– проаналізувати предметну область та сучасні методи кластерного аналізу, що застосовуються для аналізу транспортних даних;

– обґрунтувати вибір алгоритму кластеризації, метрик подібності та програмних засобів реалізації;

– розробити математичну модель кластеризації пасажиропотоків;

– спроєктувати архітектуру програмного модуля кластеризації;

– реалізувати програмний модуль із використанням мови програмування Python;

– провести експериментальні дослідження на основі реальних даних та виконати аналіз отриманих результатів.

Об’єктом дослідження є пасажиропотоки міського громадського транспорту.

Предметом дослідження є методи та алгоритми кластерного аналізу, а також програмні засоби їх реалізації для аналізу пасажиропотоків.

У роботі використано такі методи дослідження: методи кластерного аналізу, статистичної обробки даних, нормалізації ознак, візуалізації даних, а також методи програмної інженерії для проєктування та реалізації програмних систем.

Практичне значення роботи полягає у можливості використання розробленого програмного модуля як інструменту підтримки прийняття рішень у сфері міського транспортного планування, зокрема для аналізу структури пасажиропотоків, оптимізації маршрутної мережі та коригування розкладів руху.

Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків [1].

За результатами досліджень опубліковані тези доповідей на IV Всеукраїнській науково-практичній конференції молодих вчених і студентів «Інтелектуальні комп’ютерні системи та мережі» (20 травня 2026 р., м. Тернопіль, Західноукраїнський національний університет) [2]:

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА МЕТОДІВ КЛАСТЕРИЗАЦІЇ ПАСАЖИРОПОТОКІВ

1.1 Аналіз транспортної системи міста Тернополя як об’єкта дослідження

Громадський транспорт міста Тернополя є складною динамічною системою, яка забезпечує щоденні пасажирські перевезення між житловими, діловими, освітніми та промисловими зонами міста. До складу транспортної системи входять автобусні та тролейбусні маршрути, що функціонують за фіксованими схемами руху та розкладом.

Тернопіль, як обласний центр із населенням близько 225 тисяч осіб [3], має розвинену мережу громадського транспорту, яка налічує понад 30 автобусних маршрутів та 11 тролейбусних ліній [4-5]. Щоденно послугами громадського транспорту користуються від 80 до 120 тисяч пасажирів, що становить значну частку мешканців міста. Загальна протяжність маршрутної мережі перевищує 400 кілометрів, охоплюючи практично всі житлові райони та ключові об'єкти міської інфраструктури.

Транспортна інфраструктура міста формувалася протягом десятиліть і відображає як історичні особливості забудови, так і сучасні тенденції урбанізації [5]. Центральна частина міста характеризується високою щільністю маршрутів та зупинок, тоді як периферійні райони обслуговуються меншою кількістю ліній із збільшеними інтервалами руху. Така нерівномірність розподілу транспортних ресурсів створює передумови для застосування оптимізаційних методів на основі аналізу реальних пасажиропотоків.

Транспортна система Тернополя складається з кількох взаємопов'язаних підсистем, кожна з яких має свої функціональні особливості:

Автобусна мережа є найбільш розгалуженою частиною системи громадського транспорту. Вона включає як муніципальні маршрути, що обслуговуються комунальним підприємством "Тернопільське тролейбусне управління", так і приватні маршрутні таксі. Автобусні маршрути забезпечують

сполучення між віддаленими мікрорайонами, промисловими зонами та центром міста. Особливістю автобусної мережі є висока гнучкість у коригуванні маршрутів та можливість швидкого реагування на зміни попиту.

Тролейбусна мережа охоплює центральну частину міста та основні транспортні магістралі. Тролейбуси характеризуються вищою пасажиромісткістю порівняно з автобусами (до 110 пасажирів) та є екологічно чистішим видом транспорту. Інфраструктура тролейбусних ліній включає контактну мережу загальною протяжністю понад 50 кілометрів, що обмежує можливості оперативного змінення маршрутів, але забезпечує стабільність функціонування системи.

Зупинкові пункти є ключовими елементами транспортної інфраструктури, що забезпечують посадку та висадку пасажирів. У місті функціонує понад 200 зупинок громадського транспорту, розташованих із середнім інтервалом 400-600 метрів на магістральних маршрутах та 300-400 метрів у житлових зонах [6-7]. Розміщення зупинок визначається містобудівними нормами, пішохідною доступністю та інтенсивністю пасажиропотоків. Частина зупинок обладнана навісами, інформаційними табло та лавками для очікування, що підвищує комфорт користування транспортом.

Диспетчерська система координує роботу всієї транспортної мережі, контролюючи дотримання розкладів руху, оперативно реагуючи на затримки та інциденти. Сучасна диспетчеризація базується на GPS-моніторингу транспортних засобів, що дозволяє відстежувати їх місцезнаходження в реальному часі та накопичувати дані для подальшого аналізу.

Просторово-часова структура пасажиропотоків

Пасажиропотоки в межах міста мають виражений просторово-часовий характер, який визначається такими факторами:

- щільністю населення в окремих мікрорайонах – житлові масиви Дружба, Сонячний, Кутківці генерують значні ранкові потоки у напрямку центру міста та промислових зон;

– розміщенням навчальних закладів, медичних установ, адміністративних центрів – наявність у Тернополі п'яти університетів (Тернопільський національний технічний університет, Тернопільський національний педагогічний університет, Тернопільський національний медичний університет та інші) створює специфічні піки навантаження о 8:00-9:00 та 13:00-14:00, коли студенти прямують на заняття або повертаються додому;

– часовими піками (ранкові та вечірні години) – найбільша інтенсивність руху спостерігається у проміжках 7:00-9:00 та 17:00-19:00, коли відбувається масове переміщення працюючого населення; у ці періоди наповнюваність транспорту може досягати 120-150% від номінальної місткості [7];

– сезонними та соціальними особливостями – під час навчального року пасажиропотік збільшується на 15-20% порівняно з літнім періодом; святкові дні, міські заходи, ярмарки також суттєво впливають на структуру переміщень.

Просторовий розподіл пасажиропотоків характеризується радіально-кільцевою структурою [8-9]: більшість маршрутів з'єднують периферійні райони з центром міста, де зосереджені органи управління, заклади культури, торговельні центри та транспортний вузол – автовокзал і залізничний вокзал. Такий тип організації руху призводить до перевантаження центральних вулиць у пікові години та недостатнього використання транспортних потужностей на периферії.

Крім радіальних зв'язків, існують також хордові маршрути, що забезпечують пряме сполучення між окремими мікрорайонами без проходження через центр. Однак їх частка у загальній структурі маршрутної мережі є порівняно невеликою, що створює незручності для пасажирів, які змушені здійснювати пересадки та збільшувати час поїздки.

Часова динаміка функціонування системи

Транспортна система міста функціонує за складним часовим графіком, який відображає циклічність міського життя. Можна виділити декілька характерних часових періодів:

Ранковий пік (7:00-9:00) характеризується максимальним навантаженням на маршрути, що прямують до промислових зон, адміністративних центрів та навчальних закладів. У цей період спостерігається найбільша кількість одночасно працюючих транспортних засобів – до 80-90% від загального парку. Середня наповнюваність транспорту у ранкові години становить 70-80 пасажирів на один автобус чи тролейбус при номінальній місткості 60-70 осіб.

Денний міжпіковий період (9:00-16:00) відзначається помірним пасажиропотоком, який формується переважно за рахунок пенсіонерів (які мають право на безкоштовний проїзд), студентів між парами, відвідувачів медичних установ та торговельних закладів. Інтервали руху транспорту у цей час збільшуються на 30-50% порівняно з піковими періодами, середня наповнюваність знижується до 40-50%.

Вечірній пік (17:00-19:00) демонструє зворотний напрямок основних пасажиропотоків – від місць роботи до житлових районів. За інтенсивністю вечірній пік зазвичай дещо менший за ранковий (на 10-15%), оскільки частина населення затримується на роботі або здійснює додаткові поїздки (відвідування магазинів, спортивних секцій тощо).

Вечірній та нічний час (після 20:00) характеризується різким зменшенням пасажиропотоку та відповідним скороченням кількості рейсів. Після 22:00 більшість маршрутів припиняє роботу, залишаються лише окремі нічні лінії з інтервалом 40-60 хвилин.

Вихідні дні мають свою специфіку: відсутність робочих та навчальних поїздок компенсується зростанням рекреаційних переміщень – до парків, озера, торговельно-розважальних центрів. Загальний обсяг перевезень у суботу становить приблизно 60-70% від робочого дня, у неділю – 50-60%.

З точки зору інформаційних технологій, транспортна система міста може бути представлена як сукупність потоків даних, що описують рух транспортних засобів і пасажирів у просторі та часі. Саме це робить можливим застосування методів аналізу даних і машинного навчання для виявлення закономірностей у пасажиропотоках.

Сучасна транспортна система Тернополя поступово переходить до використання цифрових технологій [10-11]. Впроваджено систему GPS-моніторингу транспортних засобів, що дозволяє відстежувати їх переміщення, фіксувати затримки, контролювати дотримання маршрутів та швидкісного режиму. Дані GPS-трекерів передаються на диспетчерський пункт із частотою 30-60 секунд, формуючи детальну траєкторію руху кожної одиниці транспорту.

Електронна система оплаті проїзду на основі безконтактних карток також генерує цінні дані про пасажиропотоки. Кожна валідація картки фіксує час, маршрут і зупинку посадки пасажира, що дозволяє будувати матриці кореспонденцій та визначати найбільш завантажені напрямки. Хоча система електронного квитування ще не охоплює всіх пасажирів (частка готівкової оплати залишається значною), обсяг накопичених даних уже дозволяє проводити статистично значущий аналіз.

Інформаційні табло на зупинках та мобільні додатки для пасажирів забезпечують доступ до розкладів руху та прогнозованого часу прибуття транспорту. Це не лише підвищує комфорт користування системою, але й створює зворотний зв'язок: дані про запити користувачів у додатках можуть використовуватися для оцінки попиту на окремих напрямках.

Диспетчерська система збирає та зберігає великі обсяги інформації: траєкторії руху, швидкість, кількість виконаних рейсів, затримки, відхилення від графіка, інциденти на маршрутах. Ці дані представляють собою цінний ресурс для ретроспективного аналізу та прогнозування. Застосування алгоритмів машинного навчання до таких масивів дозволяє виявляти неочевидні закономірності, будувати прогнозні моделі завантаження та оптимізувати розклади руху.

Проблематика та виклики

Незважаючи на наявність розгалуженої маршрутної мережі, транспортна система Тернополя стикається з низкою викликів:

Нерівномірність завантаження проявляється як у просторовому, так і в часовому вимірах. Певні маршрути переповнені у пікові години, тоді як інші

рухаються напівпорожніми. Відсутність гнучкого реагування на зміни попиту призводить до неефективного використання транспортних ресурсів.

Застаріла маршрутна мережа була сформована ще в радянський період і не повною мірою відповідає сучасній структурі міста. Нові житлові райони часто залишаються недостатньо забезпеченими транспортним сполученням, що змушує мешканців користуватися особистим автомобілем і посилює транспортні затори.

Недостатня інтеграція видів транспорту – відсутність єдиної системи пересадочних вузлів та синхронізованих розкладів ускладнює мультимодальні поїздки. Пасажири змушені витрачати зайвий час на очікування при пересадках.

Обмежений аналіз даних – наявні інформаційні системи здебільшого використовуються для оперативного контролю, тоді як потенціал накопичених даних для стратегічного планування та оптимізації залишається недостатньо реалізованим.

Фінансові обмеження комунального транспорту стримують оновлення рухомого складу та інфраструктури, що впливає на якість послуг та привабливість громадського транспорту для пасажирів [12].

1.2 Аналіз джерел даних про пасажиропотоки

Для аналізу та кластеризації пасажиропотоків можуть використовуватися різні типи даних. Вибір відповідного джерела даних або їх комбінації визначається конкретними цілями дослідження, доступністю інформації та технічними можливостями її збору й обробки.

Сучасні транспортні системи генерують величезні обсяги інформації завдяки впровадженню цифрових технологій моніторингу та обліку. Ці дані відрізняються за природою, детальністю, частотою оновлення та рівнем покриття транспортної мережі [13]. Комплексний підхід до використання різних

джерел дозволяє отримати більш повну картину функціонування транспортної системи та поведінки пасажирів.

Якість вихідних даних безпосередньо впливає на достовірність результатів аналізу. Неповні, зашумлені або неточні дані можуть призвести до хибних висновків та неефективних управлінських рішень [14]. Тому критичний аналіз джерел даних є обов'язковим етапом будь-якого дослідження транспортних систем.

Розглянемо основні джерела даних.

GPS-дані є одним із найбільш поширених і цінних джерел інформації про рух транспортних засобів. Сучасні системи моніторингу транспорту використовують GPS-трекери, встановлені на кожній одиниці рухомого складу, які з певною частотою (зазвичай 30-60 секунд) передають дані на диспетчерський сервер [11, 15].

Структура GPS-даних включає наступні поля:

- Унікальний ідентифікатор транспортного засобу
- Часова мітка (timestamp) з точністю до секунди
- Географічні координати (широта та довгота) з точністю до декількох метрів
- Швидкість руху в км/год
- Напрямок руху (азимут)
- Додаткова інформація (стан двигуна, рівень палива тощо)

GPS-дані дозволяють будувати реальні траєкторії руху транспортних засобів, визначати середню швидкість на різних ділянках маршруту, виявляти місця заторів та затримок [16]. Шляхом аналізу GPS-треків можна оцінити регулярність руху, виявити систематичні відхилення від розкладу та оптимізувати маршрути.

Для оцінки пасажиропотоків GPS-дані можуть бути опосередковано використані шляхом аналізу часу стоянки на зупинках: більш тривалі зупинки, як правило, свідчать про більшу кількість пасажирів, що садять або виходять.

Однак така оцінка є приблизною і потребує калібрування за допомогою інших джерел даних.

Дані електронних квитків та валідаторів наступне джерело даних.

Системи електронних квитків та валідаторів фіксують кожну транзакцію оплати проїзду, що робить їх надзвичайно цінним джерелом інформації про пасажиропотоки. У Тернополі, як і в багатьох інших містах України, поступово впроваджується система безконтактних транспортних карток.

Структура даних електронного квитування:

- Унікальний ідентифікатор картки пасажира (анонімізований)
- Часова мітка валідації (дата та час)
- Ідентифікатор маршруту
- Номер транспортного засобу
- Ідентифікатор зупинки або координати валідації
- Тип квитка (разовий, пільговий, абонемент)
- Вартість поїздки

Переваги даних електронного квиткування:

- Пряма інформація про кількість пасажирів
- Дані про час посадки пасажира з точністю до секунди
- Можливість відстежування індивідуальних переміщень (за анонімним ID)
- Інформація про типи пасажирів (пільгові категорії, студенти тощо)
- Автоматичний збір без додаткових витрат
- Висока достовірність даних

Обмеження даних електронного квитування:

- Неповне покриття: частина пасажирів досі платить готівкою
- Відсутність інформації про місце висадки (у більшості систем)
- Можливість безквиткового проїзду, що знижує репрезентативність даних
- Пільговики, які не валідують картку, не потрапляють у статистику
- Проблема ідентифікації пересадок

– Дані доступні лише після валідації, немає прогнозу попиту

Традиційним джерелом інформації про роботу громадського транспорту є статистичні звіти, що ведуться транспортними підприємствами та органами управління транспортом. Ці звіти формуються на основі різних джерел: показань лічильників пасажирів, звітів водіїв, кондукторів, даних каси та контрольно-ревізійної служби.

Типи статистичних даних:

– Загальна кількість перевезених пасажирів за період (день, місяць, рік)

– Пасажирообіг (пасажиро-кілометри)

– Кількість виконаних рейсів

– Середня наповнюваність транспорту

– Дані обстежень пасажиропотоків на окремих маршрутах

– Показники роботи транспорту (коефіцієнт випуску на лінію, пробіг тощо)

Статистичні звіти корисні для загального моніторингу стану транспортної системи, виявлення трендів та сезонних коливань на рівні всього міста або окремих маршрутів. Однак для детального аналізу пасажиропотоків, побудови прогнозних моделей та кластеризації маршрутів вони мають обмежену цінність через недостатню гранулярність.

Наступне джерело даних – відкриті міські дані (Open Data).

У рамках міжнародної ініціативи відкритих даних багато міст публікують інформацію про роботу громадського транспорту у відкритому доступі [4]. Це можуть бути як статичні дані (GTFS – General Transit Feed Specification), так і дані в реальному часі (GTFS Realtime).

Формат GTFS (статичні дані) містить:

– Інформацію про маршрути та їх назви

– Координати зупинок

– Розклади руху транспорту

– Схеми маршрутів (послідовність зупинок)

– Інформацію про тарифи

- Дні роботи маршрутів

Формат GTFS Realtime надає:

- Поточне місцезнаходження транспортних засобів
- Прогнозований час прибуття на зупинки
- Інформацію про затримки та відміни рейсів
- Повідомлення для пасажирів

Переваги відкритих даних:

- Вільний доступ для дослідників та розробників
- Стандартизований формат, зручний для обробки
- Можливість порівняння даних різних міст
- Сприяння розробці сторонніх додатків для пасажирів
- Прозорість роботи транспортної системи

Відкриті дані особливо корисні для розробки додатків-планувальників маршрутів, інформаційних систем для пасажирів та дослідницьких проєктів, що потребують базової інформації про транспортну мережу. Вони можуть слугувати основою для моделювання та симуляції транспортних процесів.

Окрім основних джерел, існують і додаткові, що можуть доповнювати інформацію про пасажиропотоки:

- автоматичні лічильники пасажирів;
- мобільні додатки для пасажирів;
- відеоспостереження та комп'ютерний зір;
- опитування та обстеження пасажирів.

1.3 Огляд методів аналізу та кластеризації даних

1.3.1 Основи кластеризації та її застосування у транспортних задачах

Кластеризація є одним із базових методів інтелектуального аналізу даних (Data Mining) і призначена для автоматичного групування об'єктів за мірою їх

подібності без попереднього навчання [17-22]. Кластеризація відноситься до методів навчання без вчителя і дозволяє виявляти приховану структуру даних.

Основна ідея кластеризації полягає у розбитті множини об'єктів на групи (кластери) таким чином, щоб об'єкти всередині одного кластера були максимально подібними між собою, а об'єкти з різних кластерів – максимально відмінними. Формально, задача кластеризації формулюється наступним чином: для заданої множини об'єктів $X = \{x_1, x_2, ..., x_n\}$ необхідно знайти розбиття на k підмножин $C_1, C_2, ..., C_k$ таких, що кожен об'єкт належить рівно одному кластеру (для жорсткої кластеризації) або має ступені приналежності до різних кластерів (для нечіткої кластеризації).

У контексті аналізу транспортних систем кластеризація відкриває широкі можливості для вирішення різноманітних практичних задач:

- Виявлення маршрутів із подібними характеристиками пасажиропотоків для оптимізації розкладів
- Групування зупинок за інтенсивністю використання для планування інфраструктури
- Сегментація часових періодів для адаптивного управління частотою руху
- Ідентифікація просторових зон із подібними патернами мобільності
- Виявлення аномальних ситуацій та відхилень від нормальної роботи

Ефективність застосування кластеризації до транспортних даних підтверджується численними дослідженнями у галузі інтелектуальних транспортних систем [18].

У задачах аналізу пасажиропотоків об'єктами кластеризації можуть бути різні сутності транспортної системи, що характеризуються певними ознаками: транспортні маршрути, зупинки громадського транспорту, часові інтервали, просторові координати з інтенсивністю пасажиропотоку.

Транспортні маршрути можуть групуватися за подібністю профілів пасажиропотоків протягом дня, тижня або року. Ознаками для кластеризації маршрутів можуть бути:

- Часові ряди пасажиропотоків (погодинні, добові показники)
- Коефіцієнт нерівномірності завантаження
- Співвідношення між піковими та міжпіковими періодами
- Середня наповнюваність транспортних засобів
- Варіабельність пасажиропотоків (стандартне відхилення, коефіцієнт варіації)
- Загальна протяжність маршруту та кількість зупинок
- Тип обслуговуваних територій (житлові, промислові, адміністративні)

Зупинки класифікуються за інтенсивністю використання, типом пасажирів та функціональним призначенням прилеглих територій. Ознаки для кластеризації зупинок:

- Загальна кількість посадок/висадок за період
- Розподіл пасажиропотоку за часом доби
- Співвідношення посадок до висадок (баланс потоків)
- Кількість маршрутів, що обслуговують зупинку
- Щільність населення в радіусі 500 метрів
- Наявність поблизу ключових об'єктів (школи, лікарні, торгові центри)
- Пішохідна доступність від житлових масивів

Групування зупинок допомагає оптимізувати розміщення інфраструктури (навіси, лавки, інформаційні табло), планувати капітальні вкладення та визначати пріоритетні місця для покращення умов очікування транспорту [7].

Кластеризація часових періодів виявляє типові режими роботи транспортної системи протягом доби, тижня або року. Ознаки для групування часових інтервалів:

- Сумарний пасажиропотік у мережі
- Кількість активних транспортних засобів
- Середня швидкість руху транспорту
- Завантаженість окремих ділянок мережі
- Показники регулярності руху (відхилення від розкладу)
- Кількість затримок та інцидентів

Результати кластеризації часу дозволяють виділити пікові та міжпікові періоди, нічні години, вихідні дні, що є основою для диференційованого планування частоти руху та оптимізації використання рухомого складу.

Геопросторова кластеризація виявляє зони концентрації попиту на транспортні послуги. Об'єктами виступають точки на карті міста, характеризовані:

- Географічними координатами (широта, довгота)
- Інтенсивністю пасажиропотоку (посадки + висадки)
- Напрямками переміщень (вектори Origin-Destination)
- Часовою динамікою активності
- Типом функціональної зони (житлова, ділова, рекреаційна)

Просторова кластеризація корисна для виявлення "гарячих точок" – місць із найвищим попитом на транспорт, що потребують першочергової уваги при плануванні маршрутної мережі.

1.3.2 Класифікація методів кластеризації

Існує кілька підходів до класифікації алгоритмів кластеризації залежно від способу формування кластерів [24-26]:

1. Методи розбиття (Partitioning methods) – ділять об'єкти на заздалегідь задану кількість кластерів шляхом ітеративного покращення початкового розбиття. Приклади: k-means, k-medoids, k-modes.

2. Ієрархічні методи (Hierarchical methods) – будують деревоподібну структуру кластерів, що дозволяє аналізувати дані на різних рівнях деталізації. Поділяються на агломеративні (знизу вгору) та дивізивні (зверху вниз).

3. Щільнісні методи (Density-based methods) – виявляють кластери як області з високою щільністю об'єктів, відокремлені областями з низькою щільністю. Приклади: DBSCAN, OPTICS, DENCLUE.

4. Сіткові методи (Grid-based methods) – розбивають простір ознак на комірки сітки та виконують кластеризацію на основі статистики комірок. Приклади: STING, CLIQUE.

5. Імовірнісні методи (Model-based methods) – припускають, що дані генеруються статистичною моделлю (сумішшю розподілів) та виконують кластеризацію шляхом оцінки параметрів моделі. Приклад: алгоритм EM (Expectation-Maximization) для гаусівських сумішей.

Для транспортних даних найбільш затребуваними є методи розбиття (завдяки ефективності на великих даних), щільнісні методи (для роботи з просторовими даними) та ієрархічні методи (для багаторівневого аналізу).

Найбільш поширеними методами кластеризації є: K-means, ієрархічний, DBSCAN та OPTICS. В таблиці 1.1 приведені характеристики цих методів. В таблиці 1.2 проведено порівняння методів кластеризації за деякими критеріями.

Таблиця 1.1 – Найпоширеніші методи кластеризації

Метод	Характеристика	Переваги	Недоліки	Застосування
K-means	Принцип роботи: 1.Вибір k початкових центроїдів 2.Призначення об'єктів до найближчого центру 3.Перерахунок центрів кластерів 4.Повторення до збіжності	– висока швидкодія – простота реалізації – масштабованість	– потрібно знати k – чутливість до викидів – лише сферичні кластери	Групування маршрутів за профілями пасажиропотоків
ієрархічний	Підходи: Агломеративний: знизу вгору, об'єднання кластерів Дивізивний: зверху вниз, поділ кластерів Результат: Дендрограма (дерево кластерів)	– не потрібно k – візуалізація структури – детермінований результат	– висока складність $O(n^3)$ – погано масштабується	Класифікація зупинок за типами та інтенсивністю

Продовження табл. 1.1

DBSCAN	Параметри: ε: радіус околу MinPts: мін. точок у кластері Типи точок: ядрові, прикордонні, шумові	– довільна форма – виявлення шуму – авто визначення k	– чутливість до ε, MinPts – різна щільність – складність $O(n^2)$	Виявлення зон концентрації пасажиропотоків
OPTICS	Особливості: Упорядкування замість розбиття Reachability plot для візуалізації Багаторівнева структура щільності	– різна щільність – не потрібно ε – багаторівневість	– складна інтерпретація – вища складність – потрібна постобробка	Аналіз зупинок з різною інтенсивністю

Таблиця 1.2 – Порівняльна таблиця методів кластеризації

Критерій	K-means	Ієрархічна	DBSCAN	OPTICS
Потрібно задавати k	Так	Ні	Ні	Ні
Складність	$O(n \cdot k \cdot t)$	$O(n^3)$	$O(n^2)$	$O(n^2)$
Форма кластерів	Сферична	Будь-яка	Будь-яка	Будь-яка
Виявлення шуму	Ні	Ні	Так	Так
Масштабованість	Висока	Низька	Середня	Середня
Ідеально для транспорту	Великі дані	Мала вибірка	Геодані	Різна щільність

Ефективність кластеризації значною мірою залежить від вибору метрики для вимірювання подібності об'єктів [17-28]. Для різних типів даних підходять різні метрики:

1) Евклідова відстань – найпоширеніша метрика для числових даних:

$$d(x, y) = \sqrt{(\Sigma_i (x_i - y_i)^2)}$$

Підходить для k-means, ієрархічної кластеризації, коли ознаки мають порівнянні шкали.

2) Манхеттенська відстань (відстань міських кварталів):

$$d(x, y) = \Sigma_i |x_i - y_i|$$

Менш чутлива до викидів, корисна для розріджених даних.

3) Косинусна подібність – вимірює кут між векторами:

$$\cos(x, y) = (x \cdot y) / (||x|| \cdot ||y||)$$

Ефективна для текстових даних та високовимірних розріджених векторів.

4) Відстань Мінковського – узагальнення евклідової та манхеттенської:

$$d(x, y) = (\Sigma_i |x_i - y_i|^p)^{\wedge}(1/p).$$

При p=2 отримуємо евклідову відстань, при p=1 – манхеттенську.

1.4 Порівняльний аналіз програмних засобів реалізації кластеризації

Для реалізації програмного модуля кластеризації пасажиропотоків можуть бути використані різні програмні середовища та інструментальні засоби, які відрізняються за рівнем автоматизації, гнучкості, масштабованості та можливостями інтеграції з іншими інформаційними системами.

Використання мови програмування Python [29-31].

Найбільш доцільним для спеціальності «Комп'ютерна інженерія» є використання мови програмування Python, оскільки вона забезпечує:

- широкий набір бібліотек для обробки та аналізу великих обсягів даних;
- зручні інструменти для реалізації алгоритмів машинного навчання та кластеризації;
- можливість створення модульних і масштабованих програмних рішень;
- просту інтеграцію з базами даних, веб-сервісами та інформаційними системами реального часу.

Найбільш поширеними бібліотеками Python для реалізації задач кластеризації є:

- pandas – для завантаження, очищення та агрегування табличних даних;
- NumPy – для ефективних числових обчислень;
- scikit-learn – для реалізації алгоритмів кластеризації (k-means, DBSCAN, агломеративна кластеризація);
- matplotlib та seaborn – для візуалізації результатів аналізу.

Зазначений програмний стек дозволяє реалізувати гнучкий, розширюваний та повторно використовуваний програмний модуль, що є важливим для інженерних застосувань.

Використання середовища R [32-33].

Мова програмування R традиційно використовується для статистичного аналізу та обробки даних і має потужний набір інструментів для кластерного аналізу. До її переваг належать:

- наявність спеціалізованих статистичних пакетів;
- розвинуті засоби візуалізації;
- зручна реалізація класичних методів кластеризації.

Для задач кластерного аналізу в середовищі R часто використовуються пакети:

- stats (hclust, kmeans);
- cluster (PAM, CLARA);
- factoextra – для візуалізації та оцінювання якості кластерів.

Разом з тим, R менш орієнтована на розробку повноцінних програмних модулів і промислових інформаційних систем, що обмежує її використання у проєктах, де необхідна тісна інтеграція з іншими програмними компонентами або обробка потокових даних.

Використання програмного комплексу STATISTICA [34-35].

Програмний комплекс STATISTICA є комерційним статистичним середовищем, яке широко застосовується для аналізу даних, зокрема для кластеризації. Його перевагами є:

- наявність графічного інтерфейсу користувача;
- реалізація стандартних алгоритмів кластерного аналізу;
- зручність для навчальних та прикладних статистичних досліджень.

Водночас STATISTICA має низку обмежень:

- обмежені можливості автоматизації та масштабування;
- складність інтеграції з іншими інформаційними системами;
- залежність від ліцензійного програмного забезпечення.

У контексті комп'ютерної інженерії STATISTICA доцільно використовувати для попереднього аналізу даних або перевірки гіпотез, але не як основу для розробки програмного модуля.

Порівняльна характеристика програмних засобів

Таким чином, Python є найбільш придатним інструментом для реалізації програмного модуля кластеризації пасажиропотоків, оскільки поєднує:

- алгоритмічну гнучкість;
- можливість масштабування;
- відкритість і розширюваність;
- зручність інтеграції з іншими програмними компонентами.

Середовище R та програмний комплекс STATISTICA доцільно розглядати як допоміжні інструменти для статистичного аналізу та верифікації результатів.

Таблиця 1.3 порівнює характеристики програмних засобів для реалізації кластеризації.

Таблиця 1.3 – Порівняльна характеристика програмних засобів для реалізації кластеризації

Критерій	Python	R	STATISTICA
Тип середовища	Мова програмування загального призначення	Статистична мова програмування	Комерційний статистичний пакет
Орієнтація	Інженерні та прикладні програмні системи	Статистичний аналіз і дослідження	Статистичний аналіз з GUI
Підтримка алгоритмів кластеризації	k-means, DBSCAN, агломеративна, спектральна	k-means, ієрархічна, PAM, CLARA	k-means, ієрархічна кластеризація
Гнучкість алгоритмічної реалізації	Висока (повна кастомізація)	Середня	Низька
Обробка великих обсягів даних	Висока (NumPy, pandas, Dask)	Середня	Обмежена
Інтеграція з БД та веб-сервісами	Повна підтримка (SQL, REST API)	Обмежена	Практично відсутня
Автоматизація та масштабування	Висока	Середня	Низька
Візуалізація результатів	matplotlib, seaborn, plotly	ggplot2, factoextra	Вбудовані засоби
Ліцензія	Відкрита (open source)	Відкрита (open source)	Комерційна

Для реалізації кластерного аналізу використовуються різні програмні засоби, зокрема бібліотека *scikit-learn* мови Python [36], статистичне середовище R [37].

1.5 Постановка задачі дослідження

Метою даної кваліфікаційної роботи є розробка програмного модуля кластеризації пасажиропотоків міста Тернополя, який дозволяє виявляти характерні групи маршрутів або зон з подібною інтенсивністю перевезень.

Для досягнення поставленої мети необхідно розв’язати такі задачі:

- проаналізувати предметну область міських пасажирських перевезень;
- дослідити методи кластеризації, придатні для транспортних даних;
- розробити архітектуру програмного модуля;
- реалізувати алгоритм кластеризації;
- провести експериментальне дослідження на реальних або модельних даних.

У першому розділі проведено аналіз транспортної системи міста Тернополя як об’єкта дослідження та розглянуто основні джерела даних про пасажиропотоки. Проаналізовано методи кластеризації даних і програмні засоби їх реалізації. На основі виконаного аналізу сформульовано постановку задачі розробки програмного модуля кластеризації пасажиропотоків.

2 РОЗРОБКА АРХІТЕКТУРИ ТА МАТЕМАТИЧНОГО ЗАБЕЗПЕЧЕННЯ ПРОГРАМНОГО МОДУЛЯ КЛАСТЕРИЗАЦІЇ

2.1 Формалізація вхідних даних та постановка задачі кластеризації

Для реалізації програмного модуля кластеризації пасажиропотоків необхідно виконати формалізацію вхідних даних, які використовуються в процесі аналізу.

У даній роботі вхідні дані представлені у вигляді середньої кількості пасажирів на кожній зупинці міста Тернополя, агрегованої за фіксованими часовими інтервалами доби: ранок, обід та вечір. Такий підхід дозволяє зменшити вплив випадкових коливань і забезпечити стабільність результатів кластеризації.

Кожна зупинка громадського транспорту розглядається як окремий об’єкт кластеризації та описується вектором ознак:

$$\mathbf{x}_i = (x_i^{(r)}, x_i^{(o)}, x_i^{(v)}),$$

де $x_i^{(r)}$ – середня кількість пасажирів у ранковий період,

$x_i^{(o)}$ – середня кількість пасажирів в обідній період,

$x_i^{(v)}$ – середня кількість пасажирів у вечірній період

для *i*-тої зупинки.

Сукупність даних для всіх зупинок може бути представлена у вигляді матриці:

$$\mathbf{X} = \begin{pmatrix} x_1^{(r)} & x_1^{(o)} & x_1^{(v)} \\ x_2^{(r)} & x_2^{(o)} & x_2^{(v)} \\ \text{M} & \text{M} & \text{M} \\ x_n^{(r)} & x_n^{(o)} & x_n^{(v)} \end{pmatrix},$$

де n – кількість зупинок громадського транспорту.

Метою кластеризації є групування зупинок міста за подібністю добових профілів пасажиропотоків, що дозволяє виявити типові категорії зупинок, наприклад зупинки з переважно ранковим, рівномірним або вечірнім навантаженням.

2.2 Нормалізація даних та вибір метрики подібності

Перед застосуванням алгоритмів кластеризації необхідно виконати попередню обробку вхідних даних, оскільки значення ознак можуть суттєво відрізнятися за масштабом, що впливає на результати групування.

У даній роботі кожна зупинка громадського транспорту описується вектором з трьох числових ознак, які відповідають середній кількості пасажирів у різні часові періоди доби: ранок, обід та вечір. Значення цих ознак можуть мати різні діапазони, тому без попередньої нормалізації окремі часові інтервали можуть непропорційно впливати на процес кластеризації.

Нормалізація даних

Для приведення значень ознак до порівнюваного масштабу в роботі використовується нормалізація методом мін–макс, яка визначається формулою:

$$x'_{ij} = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j)},$$

де x_{ij} – початкове значення j-тої ознаки для i-тої зупинки;

$\min(x_j)$ – мінімальне значення j-тої ознаки серед усіх зупинок;

$\max(x_j)$ – максимальне значення j-тої ознаки серед усіх зупинок;

x'_{ij} – нормалізоване значення ознаки.

У результаті нормалізації всі значення ознак належать інтервалу [0; 1], що забезпечує рівний внесок кожного часового інтервалу в обчислення міри подібності між об’єктами.

Вибір метрики подібності

Одним із ключових етапів кластеризації є вибір метрики, яка використовується для оцінювання подібності між об’єктами. У задачах аналізу пасажиропотоків доцільно застосовувати метрики, які враховують не лише абсолютні значення, а й форму добового профілю навантаження.

Однією з найпоширеніших метрик для кластерного аналізу є евклідова відстань, яка визначає геометричну відстань між двома точками у багатовимірному просторі ознак.

Для двох зупинок, описаних векторами ознак

$\mathbf{x}_i = (x_{i1}, x_{i2}, x_{i3})$ та

$\mathbf{x}_k = (x_{k1}, x_{k2}, x_{k3})$,

евклідова відстань обчислюється за формулою:

$$d_E(\mathbf{x}_i, \mathbf{x}_k) = \sqrt{\sum_{j=1}^{3} (x_{ij} - x_{kj})^2}.$$

Евклідова метрика враховує абсолютні відмінності між значеннями пасажиропотоків у відповідних часових інтервалах. Тому вона є доцільною для задач, у яких важливим є загальний рівень навантаження зупинок.

Разом з тим, навіть після нормалізації, евклідова відстань залишається чутливою до сумарної інтенсивності пасажиропотоку, що може ускладнювати виділення зупинок з подібною формою добового профілю.

Для аналізу подібності структури добового розподілу пасажиропотоків у роботі також використовується косинусна подібність, яка визначається як косинус кута між двома векторами ознак:

$$\cos(\mathbf{x}_i, \mathbf{x}_k) = \frac{\mathbf{x}_i \cdot \mathbf{x}_k}{\| \mathbf{x}_i \| \| \mathbf{x}_k \|},$$

де $\mathbf{x}_i$, $\mathbf{x}_k$ – нормалізовані вектори ознак для *i*-тої та *k*-тої зупинок;

$\mathbf{x}_i \cdot \mathbf{x}_k$ – скалярний добуток векторів;

$\| \mathbf{x}_i \|$, $\| \mathbf{x}_k \|$ – їхні евклідові норми.

Косинусна подібність дозволяє оцінювати схожість структури розподілу пасажиропотоку за часовими інтервалами, незалежно від загальної інтенсивності перевезень. Це є важливим для виділення зупинок із подібними добовими профілями, наприклад зупинок із переважно ранковим або вечірнім навантаженням.

Для використання косинусної подібності в алгоритмах кластеризації вводиться відповідна косинусна відстань, яка визначається як:

$$d_{\cos}(\mathbf{x}_i, \mathbf{x}_k) = 1 - \cos(\mathbf{x}_i, \mathbf{x}_k).$$

Порівняльний аналіз показує, що:

- евклідова метрика є доцільною для оцінювання загального рівня пасажирського навантаження;
- косинусна метрика краще відображає подібність форми добового профілю пасажиропотоку.

З огляду на мету роботи – групування зупинок за характером добового розподілу пасажирів, у подальших дослідженнях основну увагу приділено використанню косинусної відстані, тоді як евклідова метрика застосовується для порівняльного аналізу результатів кластеризації.

Порівняно з евклідовою відстанню, косинусна відстань:

- менш чутлива до абсолютних значень пасажиропотоків;
- дозволяє враховувати форму добового навантаження;
- забезпечує кращу інтерпретованість результатів кластеризації.

Таким чином, поєднання нормалізації даних методом мін–макс та використання косинусної відстані створює коректні передумови для подальшого застосування алгоритмів кластеризації зупинок міста Тернополя.

2.3 Опис алгоритму кластеризації

Для групування зупинок громадського транспорту за подібністю добових профілів пасажиропотоків у даній роботі використовується алгоритм k-means, який є одним із найбільш поширених і ефективних методів кластерного аналізу.

Алгоритм k-means належить до методів неконтрольованого навчання та передбачає поділ множини об’єктів на k кластерів таким чином, щоб мінімізувати сумарну внутрішньокластерну відстань між об’єктами та центрами кластерів.

Математична постановка задачі

Нехай задано множину об’єктів кластеризації

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\},$$

де кожен об’єкт $\mathbf{x}_i$ є вектором ознак:

$$\mathbf{x}_i = (x_i^{(r)}, x_i^{(o)}, x_i^{(v)}),$$

що відповідають середній кількості пасажирів у ранковий, обідній та вечірній періоди.

Необхідно розбити множину $\mathbf{X}$ на k неперетинних кластерів $\{C_1, C_2, \ldots, C_k\}$ так, щоб мінімізувати цільову функцію:

$$J = \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in C_j} d(\mathbf{x}_i, \boldsymbol{\mu}_j),$$

де $\boldsymbol{\mu}_j$ – центр (центроїд) *j*-го кластера,

$d(\cdot)$ – обрана метрика відстані (евклідова або косинусна).

Приведемо етапи роботи алгоритму k-means.

Алгоритм k-means реалізується ітеративно та включає такі основні етапи:

1. Ініціалізація центрів кластерів

Випадковим чином або за спеціальною процедурою (наприклад, k-means++) обираються початкові центроїди $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k$.

2. Призначення об’єктів до кластерів

Кожен об’єкт $\mathbf{x}_i$ відноситься до того кластера, центр якого є найближчим відповідно до обраної метрики:

$$C_j = \arg\min_j d(\mathbf{x}_i, \boldsymbol{\mu}_j).$$

3. Перерахунок центрів кластерів

Для кожного кластера обчислюється новий центр як середнє значення всіх об’єктів, що належать до нього:

$$\boldsymbol{\mu}_j = \frac{1}{|C_j|} \sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i.$$

4. Перевірка умови зупинки

Алгоритм завершується, якщо:

- центри кластерів не змінюються;
- або кількість ітерацій досягає заданого максимуму;
- або зміна цільової функції J є меншою за наперед заданий поріг.

Приведемо псевдокод алгоритму k-means

```
Вхід:
    X – матриця даних (n × 3)
    k – кількість кластерів
    d – метрика відстані
    max_iter – максимальна кількість ітерацій

Вихід:
    C – множина кластерів
    μ – центри кластерів

1. Ініціалізувати центри μ1, μ2, …, μk
2. Для iter = 1 до max_iter виконати:
3.     Для кожного об’єкта xi ∈ X:
4.         Призначити xi до найближчого центру μj
5.     Для кожного кластера Cj:
6.         Оновити центр μj як середнє значення об’єктів Cj
7.     Якщо центри не змінюються:
8.         Перервати цикл
9. Повернути C та μ
```

У контексті даної роботи алгоритм k-means застосовується для кластеризації зупинок міста Тернополя за подібністю добових профілів пасажиропотоків. Невелика розмірність простору ознак (три часові інтервали) забезпечує:

- високу швидкодію алгоритму;
- можливість наочної візуалізації результатів;
- інтерпретованість отриманих кластерів.

Використання косинусної відстані дозволяє виділяти зупинки з подібною структурою добового навантаження, тоді як евклідова метрика застосовується для порівняльного аналізу результатів.

2.4 Вибір кількості кластерів та критерії оцінювання якості кластеризації

Одним із ключових етапів застосування алгоритму k-means є вибір кількості кластерів *k*. Неправильний вибір цього параметра може призвести до втрати важливої інформації або, навпаки, до надмірної деталізації результатів кластеризації.

У даній роботі для визначення оптимальної кількості кластерів та оцінювання якості кластеризації використовуються метод «ліктя» (elbow method) та коефіцієнт силуету (silhouette score).

Метод «ліктя» (elbow method)

Метод «ліктя» базується на аналізі залежності значення цільової функції алгоритму k-means від кількості кластерів. У ролі такої функції зазвичай використовується сума внутрішньокластерних відстаней (within-cluster sum of squares, WCSS):

$$\mathrm{WCSS}(k) = \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in C_j} d(\mathbf{x}_i, \boldsymbol{\mu}_j)^2,$$

де C_j – j-тий кластер,

$\boldsymbol{\mu}_j$ – центр відповідного кластера,

$d(\cdot)$ – обрана метрика відстані.

Зі збільшенням кількості кластерів значення WCSS монотонно зменшується, оскільки об’єкти стають ближчими до своїх центрів. Оптимальне значення k визначається в точці, де подальше збільшення кількості кластерів призводить лише до незначного покращення якості кластеризації. На графіку ця точка має вигляд характерного «зламу», або «ліктя».

Метод «ліктя» є простим у реалізації та широко застосовується для попереднього оцінювання кількості кластерів.

Коефіцієнт силуету (silhouette score)

Для більш формальної оцінки якості кластеризації використовується коефіцієнт силуету, який характеризує ступінь подібності об’єкта до власного кластера порівняно з іншими кластерами.

Для кожного об’єкта $\mathbf{x}_i$ коефіцієнт силуету визначається як:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}},$$

де $a(i)$ – середня відстань між об’єктом $\mathbf{x}_i$ та всіма іншими об’єктами його кластера;

$b(i)$ – мінімальна середня відстань між об’єктом $\mathbf{x}_i$ та об’єктами найближчого сусіднього кластера.

Значення коефіцієнта силуету належить інтервалу $[-1; 1]$ і має таку інтерпретацію:

- $s(i) \approx 1$ – об’єкт добре віднесений до свого кластера;
- $s(i) \approx 0$ – об’єкт розташований на межі між кластерами;
- $s(i) < 0$ – можливе неправильне віднесення об’єкта.

Середнє значення коефіцієнта силуету для всіх об’єктів використовується як узагальнений показник якості кластеризації для заданого значення k.

Обґрунтування вибору кількості кластерів

У даній роботі вибір кількості кластерів здійснюється шляхом комбінованого використання методу «ліктя» та коефіцієнта силуету. Метод «ліктя» дозволяє звузити діапазон можливих значень k, тоді як коефіцієнт силуету забезпечує кількісну оцінку якості кластеризації.

Такий підхід дозволяє обґрунтовано визначити оптимальну кількість кластерів зупинок міста Тернополя за характером добових пасажиропотоків та підвищити надійність отриманих результатів.

2.5 Архітектура програмного модуля кластеризації пасажиропотоків

Програмний модуль кластеризації пасажиропотоків призначений для автоматизованого аналізу агрегованих даних щодо середньої кількості пасажирів на зупинках міста та їх групування за подібністю добових профілів навантаження.

Архітектура модуля побудована за модульним принципом, що забезпечує гнучкість, розширюваність та можливість повторного використання окремих компонентів. Основні функціональні блоки програмного модуля наведено на блок-схемі (рисунок 2.1).

Опишемо функціональні компоненти програмного модуля.

Перший блок – введення даних.

Забезпечує завантаження вхідних даних у вигляді таблиці, яка містить інформацію про середню кількість пасажирів на кожній зупинці в різні часові інтервали доби (ранок, обід, вечір). Дані можуть надходити з файлів форматів CSV або Excel.

Другий – блок попередньої обробки даних.

Виконує очищення та нормалізацію даних. На цьому етапі здійснюється:

- перевірка коректності вхідних значень;
- нормалізація ознак методом мін–макс;
- формування матриці ознак для подальшої кластеризації.

Третій – блок вибору параметрів кластеризації.

Дозволяє задати параметри алгоритму кластеризації, зокрема:

- кількість кластерів k;
- тип метрики подібності (евклідова або косинусна);
- максимальну кількість ітерацій алгоритму k-means.

Четвертий блок – блок кластеризації.

Реалізує алгоритм k-means відповідно до описаних у підрозділах 2.3–2.4 етапів. На цьому етапі здійснюється ітеративне групування зупинок за подібністю їх добових профілів пасажиропотоків.

Блок оцінювання якості кластеризації є п’ятим блоком.

Здійснює обчислення показників якості кластеризації, зокрема:

- значення суми внутрішньокластерних відстаней (WCSS);
- середнього коефіцієнта силуету.

Отримані значення використовуються для аналізу доцільності вибраної кількості кластерів.

Шостий блок – блок візуалізації та інтерпретації результатів.

Забезпечує подання результатів кластеризації у вигляді графіків, діаграм або таблиць. Кожна зупинка відображається з урахуванням її приналежності до певного кластера, що полегшує інтерпретацію результатів.

Блок виведення результатів – сьомий блок.

Формує вихідні дані програмного модуля, які можуть бути використані для подальшого аналізу або прийняття управлінських рішень щодо оптимізації транспортної мережі.

З метою підвищення якості результатів кластеризації в архітектурі програмного модуля передбачено механізм зворотного зв’язку між блоком оцінювання якості кластеризації та блоком вибору параметрів кластеризації.

Після виконання алгоритму k-means здійснюється оцінювання якості отриманих кластерів із використанням методу «ліктя» та коефіцієнта силуету. У разі, якщо значення показників якості не відповідають заданим критеріям або не забезпечують достатньої інтерпретованості результатів, відбувається перевибір параметрів кластеризації, насамперед кількості кластерів *k*.

Такий ітеративний підхід дозволяє автоматизувати процес підбору параметрів алгоритму та забезпечити більш обґрунтований вибір оптимальної структури кластерів.

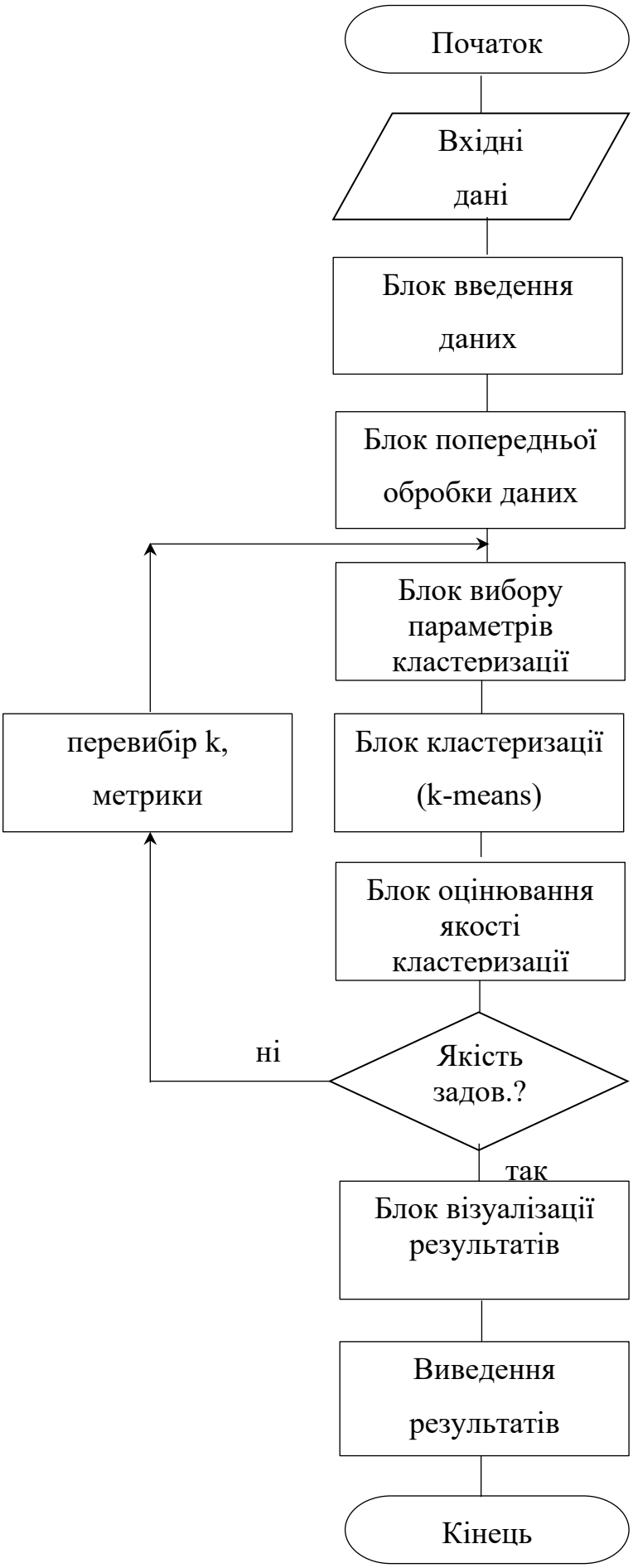

Рисунок 2.1 – Архітектура програмного модуля кластеризації

Архітектура модуля є модульною, що дозволяє змінювати алгоритми кластеризації, метрики подібності та джерела даних без зміни загальної структури системи.

У другому розділі кваліфікаційної роботи розроблено математичне та архітектурне забезпечення програмного модуля кластеризації пасажиропотоків міста Тернополя.

Проведено формалізацію вхідних даних, у яких кожна зупинка громадського транспорту описується вектором середніх значень пасажиропотоків за трьома часовими інтервалами доби. Обґрунтовано доцільність використання агрегованих показників для зменшення впливу випадкових коливань та підвищення стабільності результатів кластеризації.

Розглянуто методи попередньої обробки даних, зокрема нормалізацію ознак методом мін–макс, що забезпечує коректне застосування метрик подібності. Проаналізовано евклідову та косинусну метрики, визначено їх переваги та обмеження у задачі аналізу пасажиропотоків, а також обґрунтовано вибір косинусної відстані як основної для групування зупинок за формою добового навантаження.

Детально описано алгоритм кластеризації k-means, наведено його математичну постановку, основні етапи реалізації та псевдокод. Показано доцільність застосування цього алгоритму для задачі кластеризації зупинок у просторі малої розмірності.

Розглянуто методи вибору кількості кластерів і оцінювання якості кластеризації з використанням методу «ліктя» та коефіцієнта силуету. Запропоновано комбінований підхід до визначення оптимального числа кластерів, що підвищує надійність отриманих результатів.

Розроблено архітектуру програмного модуля кластеризації, яка включає блоки введення та попередньої обробки даних, кластеризації, оцінювання якості, візуалізації результатів та механізм зворотного зв’язку для перевибору параметрів кластеризації. Запропонована архітектура забезпечує модульність, гнучкість і можливість подальшого розширення системи.

Отримані у розділі результати створюють теоретичну та інженерну основу для програмної реалізації модуля кластеризації, яка буде представлена у третьому розділі роботи.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОГРАМНОГО МОДУЛЯ КЛАСТЕРИЗАЦІЇ ПАСАЖИРОПОТОКІВ М. ТЕРНОПОЛЯ

3.1 Технічні та програмні вимоги до функціонування програмного модуля

Для коректної роботи програмного модуля кластеризації пасажиропотоків необхідно визначити технічні та програмні вимоги, які забезпечують стабільність, продуктивність і можливість подальшого розширення системи.

Приведемо технічні вимоги.

Програмний модуль орієнтований на використання на персональному комп’ютері середнього рівня продуктивності та не потребує спеціалізованого апаратного забезпечення.

Мінімальні технічні вимоги:

- процесор з тактовою частотою не менше 2,0 ГГц;
- оперативна пам’ять обсягом не менше 4 ГБ;
- вільний дисковий простір не менше 1 ГБ;
- пристрої введення та виведення (клавіатура, миша, монітор).

Рекомендовані технічні вимоги:

- багатоядерний процесор;
- оперативна пам’ять обсягом 8 ГБ і більше;
- твердотільний накопичувач (SSD) для прискорення доступу до даних.

Зазначених ресурсів достатньо для обробки агрегованих даних щодо пасажиропотоків зупинок міста та виконання алгоритмів кластеризації у прийнятний час.

Розглянемо програмні вимоги.

Програмний модуль реалізується з використанням мови програмування Python, яка забезпечує кросплатформеність, відкритість і широкий вибір бібліотек для аналізу даних та машинного навчання.

Для коректної роботи модуля необхідне таке програмне забезпечення:

- операційна система: Windows, Linux або macOS;

– інтерпретатор Python версії 3.8 або вище;

– середовище розробки або виконання (Visual Studio Code, PyCharm, Jupyter Notebook).

Нижче приведемо використані програмні бібліотеки.

У програмному модулі застосовуються такі бібліотеки Python:

– pandas – для зчитування, зберігання та обробки табличних даних;

– NumPy – для виконання числових обчислень;

– scikit-learn – для реалізації алгоритму кластеризації k-means та обчислення показників якості кластеризації;

– matplotlib та seaborn – для візуалізації результатів кластеризації.

Використання відкритих бібліотек дозволяє уникнути ліцензійних обмежень і спрощує подальший розвиток програмного модуля.

Розглянемо вимоги до вхідних та вихідних даних.

Вхідні дані повинні бути представлені у вигляді таблиці, що містить такі поля:

– ідентифікатор або назву зупинки;

– середню кількість пасажирів у ранковий період;

– середню кількість пасажирів в обідній період;

– середню кількість пасажирів у вечірній період.

Вихідними даними програмного модуля є:

– номер кластера для кожної зупинки;

– координати центрів кластерів;

– значення показників якості кластеризації;

– графічні матеріали для інтерпретації результатів.

Опишемо вимоги до надійності та масштабованості.

Програмний модуль повинен забезпечувати:

– коректну обробку вхідних даних без аварійного завершення роботи;

– можливість зміни кількості зупинок без модифікації структури програми;

– можливість розширення набору ознак або заміни алгоритму кластеризації.

Дотримання наведених вимог забезпечує придатність програмного модуля для використання в задачах аналізу міських пасажиропотоків та його подальшого розвитку.

3.2 Програмна реалізація програмного модуля кластеризації пасажиропотоків

Програмна реалізація модуля кластеризації виконана з використанням мови програмування Python та бібліотек аналізу даних і машинного навчання. Реалізація побудована за модульним принципом, що відповідає розробленій у розділі 2 архітектурі.

3.2.1 Структура програмного проєкту

Програмний модуль реалізовано у вигляді набору логічно пов’язаних компонентів, кожен з яких відповідає за окремий етап обробки даних.

Приблизна структура проєкту має вигляд:

```
passenger_clustering/
│
├── data/
│   └── stops_passengers.csv
│
├── preprocessing.py
├── clustering.py
├── evaluation.py
├── visualization.py
└── main.py
```

де preprocessing.py – модуль попередньої обробки та нормалізації даних;

clustering.py – реалізація алгоритму k-means;

evaluation.py – оцінювання якості кластеризації;

visualization.py – візуалізація результатів;

main.py – керуючий модуль, який поєднує всі етапи.

Така структура забезпечує зрозумілість коду та можливість його подальшого розширення.

3.2.2 Зчитування та підготовка вхідних даних

Вхідні дані зберігаються у табличному форматі CSV та містять інформацію про середню кількість пасажирів на кожній зупинці міста в різні часові періоди доби.

Приклад структури вхідного файлу приведена в таблиці 3.1.

Таблиця 3.1 – Приклад структури вхідного файлу

Stop	Morning	Noon	Evening
Зупинка 1	120	85	140
Зупинка 2	60	55	70
…	…	…	…

Зчитування даних реалізується за допомогою бібліотеки pandas:

```
import pandas as pd

data = pd.read_csv("data/stops_passengers.csv")
features = data[["Morning", "Noon", "Evening"]]
```

Отримана матриця features використовується як вхід для подальших етапів аналізу.

3.2.4 Реалізація алгоритму k-means

Кластеризація зупинок виконується за допомогою алгоритму k-means. Кількість кластерів задається параметром *k*, який може змінюватися під час експериментів.

```
from sklearn.cluster import KMeans

k = 3
kmeans = KMeans(n_clusters=k, random_state=42)
labels = kmeans.fit_predict(X_norm)
```

Результатом роботи алгоритму є:

- масив labels, що містить номер кластера для кожної зупинки;
- координати центрів кластерів kmeans.cluster_centers_.

3.2.5 Оцінювання якості кластеризації

Для оцінювання якості кластеризації використовується коефіцієнт силуету:

```
from sklearn.metrics import silhouette_score

score = silhouette_score(X_norm, labels)
```

Отримане значення коефіцієнта силуету дозволяє оцінити доцільність вибраної кількості кластерів та використовується для подальшого аналізу.

3.2.6 Формування результатів кластеризації

Результати кластеризації додаються до початкової таблиці даних та можуть бути збережені у файл для подальшого використання:

```
data["Cluster"] = labels
data.to_csv("data/clustering_results.csv",
index=False)
```

Це забезпечує можливість використання результатів кластеризації для подальшого аналізу або прийняття управлінських рішень.

3.2.3 Нормалізація даних

Для забезпечення коректної роботи алгоритму кластеризації здійснюється нормалізація ознак методом мін–макс із використанням бібліотеки scikit-learn:

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
X_norm = scaler.fit_transform(features)
```

У результаті кожна з ознак набуває значень у межах інтервалу [0;1], що відповідає вимогам, визначеним у підрозділі 2.2.

3.3 Візуалізація результатів та тестування програмного модуля

Візуалізація результатів кластеризації є важливим етапом аналізу, оскільки дозволяє наочно інтерпретувати отримані кластери та оцінити коректність роботи програмного модуля. Тестування, у свою чергу, забезпечує перевірку правильності реалізації основних функцій і стабільності роботи системи.

3.3.1 Візуалізація результатів кластеризації

Оскільки кожна зупинка описується трьома числовими ознаками (ранок, обід, вечір), для візуалізації результатів кластеризації використовується:

- двовимірна проєкція простору ознак;
- графіки розподілу пасажиропотоків за кластерами;
- діаграми центрів кластерів.

Двовимірна проєкція кластерів

Для зручності сприйняття даних виконується візуалізація у двох вимірах (наприклад, «ранок – вечір»), де кожна точка відповідає окремій зупинці, а колір визначає номер кластера.

```
import matplotlib.pyplot as plt

plt.figure()
plt.scatter(
    X_norm[:, 0], X_norm[:, 2],
    c=labels
)
plt.xlabel("Ранковий пасажиропотік")
plt.ylabel("Вечірній пасажиропотік")
plt.title("Розподіл зупинок за кластерами")
plt.show()
```

Такий графік дозволяє виявити компактність кластерів і наявність можливих перетинів між ними.

Візуалізація центрів кластерів

Для інтерпретації результатів кластеризації будується графік середніх значень ознак для кожного кластера:

```
import pandas as pd

centers = pd.DataFrame(
    kmeans.cluster_centers_,
    columns=["Morning", "Noon", "Evening"]
)

centers.plot(kind="bar")
plt.title("Центри кластерів за часовими інтервалами")
plt.ylabel("Нормалізоване значення")
```

```
plt.show()
```

Ця візуалізація дозволяє інтерпретувати кожен кластер як окремий тип зупинок, наприклад з переважно ранковим, рівномірним або вечірнім пасажиропотоком.

3.3.2 Тестування програмного модуля

Тестування програмного модуля здійснювалося з метою перевірки коректності його функціонування, відповідності поставленим вимогам та стабільності роботи.

Функціональне тестування

У процесі функціонального тестування перевірялися такі аспекти:

- коректність зчитування вхідних даних;
- правильність виконання нормалізації;
- стабільність роботи алгоритму k-means для різних значень кількості кластерів k;
- коректність формування вихідних файлів із результатами кластеризації.

За результатами тестування встановлено, що програмний модуль коректно обробляє вхідні дані та формує очікувані результати без аварійних збоїв.

Тестування якості кластеризації

Для різних значень кількості кластерів обчислювався коефіцієнт силуету, що дозволило оцінити якість кластеризації та перевірити доцільність вибору параметра k.

```
from sklearn.metrics import silhouette_score

for k in range(2, 6):
    model = KMeans(n_clusters=k, random_state=42)
    labels_k = model.fit_predict(X_norm)
    score_k = silhouette_score(X_norm, labels_k)
```

```
print(f"k={k}, silhouette={score_k:.3f}")
```

Отримані значення підтвердили, що обрана кількість кластерів забезпечує достатній рівень роздільності та інтерпретованості результатів.

Перевірка механізму зворотного зв’язку

У межах тестування перевірено механізм зворотного зв’язку між блоком оцінювання якості та блоком вибору параметрів кластеризації. У випадках, коли коефіцієнт силуету зменшувався при зміні k, здійснювався повторний запуск алгоритму з іншим значенням параметра, що підтверджує коректність реалізації ітераційного підбору кількості кластерів.

3.3.3 Аналіз результатів тестування

Результати тестування показали, що програмний модуль:

- стабільно працює з агрегованими даними пасажиропотоків;
- забезпечує коректну кластеризацію зупинок;
- формує наочні візуальні представлення результатів;
- дозволяє обґрунтовано підбирати кількість кластерів.

Це підтверджує придатність розробленого програмного модуля для використання в задачах аналізу міських пасажиропотоків.

На рисунку 3.1 наведено блок-схему роботи програмного модуля кластеризації пасажиропотоків, яка відображає послідовність етапів: завантаження та валідація даних, нормалізація, підбір кількості кластерів за WCSS і silhouette, фінальна кластеризація, формування вихідних результатів та візуалізація.

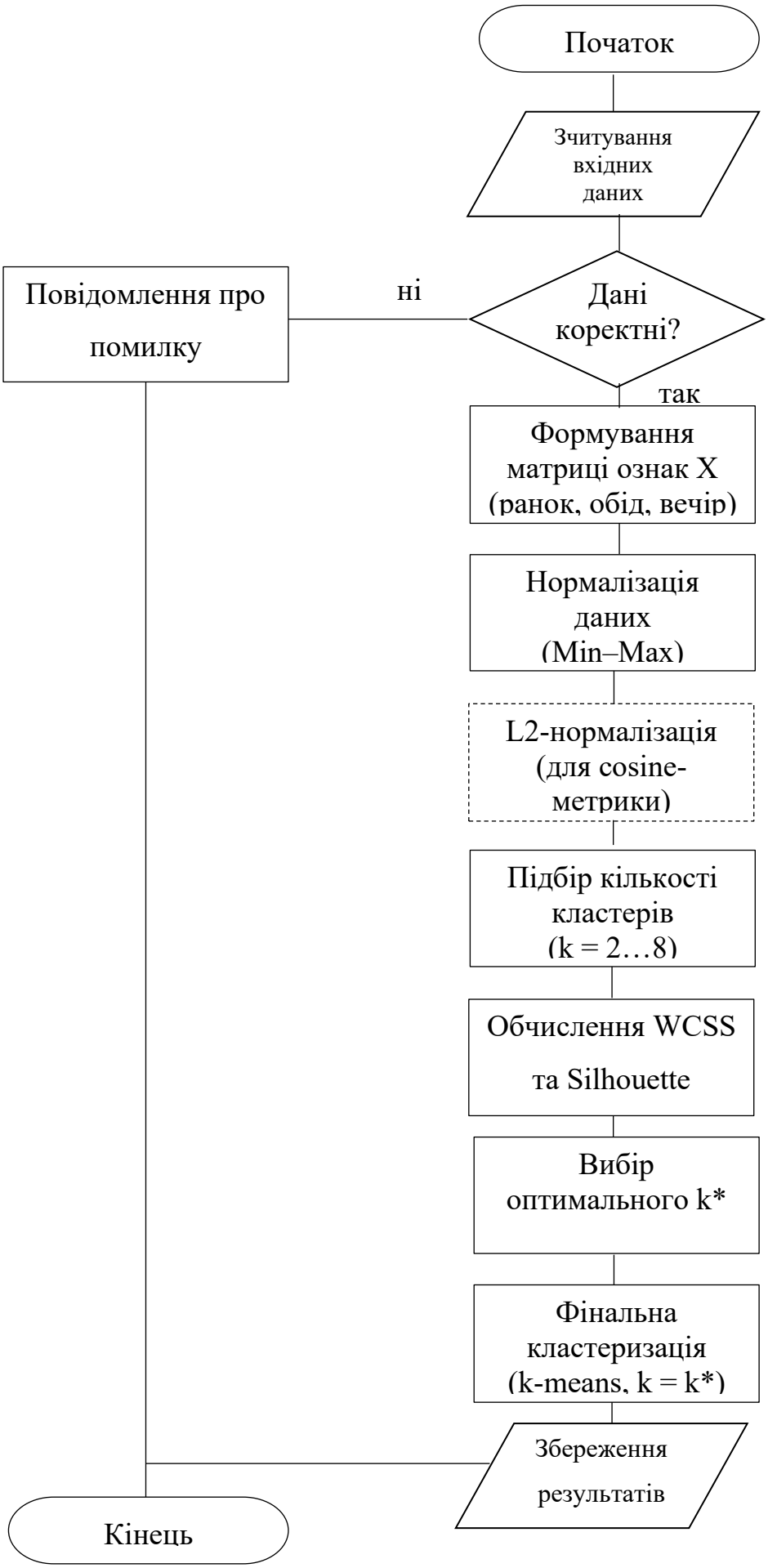

Рисунок 3.1 – Блок-схема роботи програмного модуля кластеризації

3.4 Експериментальна частина

3.4.1 Експерименти при k=3

Кожна зупинка описується вектором:

$$\mathbf{x}_i = (x_i^{(\text{ранок})}, x_i^{(\text{обід})}, x_i^{(\text{вечір})})$$

Наприклад: «Автовокзал» → (90, 78, 58)

Перед кластеризацією:

- виконано нормалізацію Min–Max;
- застосовано алгоритм k-means;
- оцінювання якості – коефіцієнт силуету.

Для демонстрації роботи модуля було виконано кластеризацію вибірки з використанням трьох кластерів.

Коефіцієнт силуету – Silhouette≈0,41. Значення > 0,4 для реальних соціально-транспортних даних – абсолютно нормальне. В таблиці 3.2 приведено фрагмент результатів кластеризації для k = 3.

Таблиця 3.2 – Фрагмент результатів кластеризації для k = 3

Зупинка	Ранок	Обід	Вечір	Кластер
Збруч	17	34	12	3
11 школа (від центру)	23	21	29	3
11 школа (до центру)	18	16	14	1
13 школа (від центру)	5	9	7	1
13 школа (до центру)	29	32	27	3
14 школа	17	7	9	1
18 школа	10	13	18	1
Автовокзал	90	78	58	2

Згідно кластеризації маємо:

– кластер 1 – зупинки з низьким та помірним пасажиропотоком (житлові райони, локальні зупинки);

– кластер 2 – зупинки з дуже високим пасажиропотоком (транспортні вузли: автовокзал, вокзали, великі ТЦ);

– кластер 3 – зупинки з помірно-високим та рівномірним навантаженням (школи, університети, центральні вулиці).

У результаті застосування алгоритму k-means до нормалізованих даних було виділено три кластери зупинок за характером добового пасажиропотоку. Отримані кластери відповідають зупинкам з низьким, помірним та високим рівнем пасажирського навантаження. Значення середнього коефіцієнта силуету підтверджує задовільну якість кластеризації та інтерпретованість результатів.

3.4.2 Експерименти по підбору оптимального k

Для визначення кількості кластерів було виконано серію експериментів для $k \in [2; 8]$. Як критерії якості використовувалися метод «ліктя» (WCSS – коефіцієнт інерції) та коефіцієнт силуету. За результатами експериментів максимальне значення коефіцієнта силуету (для косинусної метрики) отримано при $k = 8$, тому подальший аналіз виконано для восьми кластерів. В таблиці 3.3 приведені значення коефіцієнта інерції та коефіцієнта силуету.

Таблиця 3.3 – Експерименти по підбору оптимального k

k	WCSS (інерція)	Silhouette (cosine)	Silhouette (euclidean)
2	8,777	0,509	0,348
3	6,333	0,499	0,340
4	5,054	0,492	0,341
5	4,183	0,475	0,327
6	3,493	0,500	0,346
7	2,849	0,523	0,365
8	2,488	**0,528**	0,366

Візуалізація у площині «ранок–вечір» та аналіз середніх профілів кластерів підтверджують інтерпретованість отриманих груп зупинок.

На рисунку 3.2 бачимо, що Elbow (WCSS) дає плавне зменшення, без дуже різкого “ліктя” при k = 8. На рисунку 3.3 є силует-графік для різних k (cosine та euclidean).

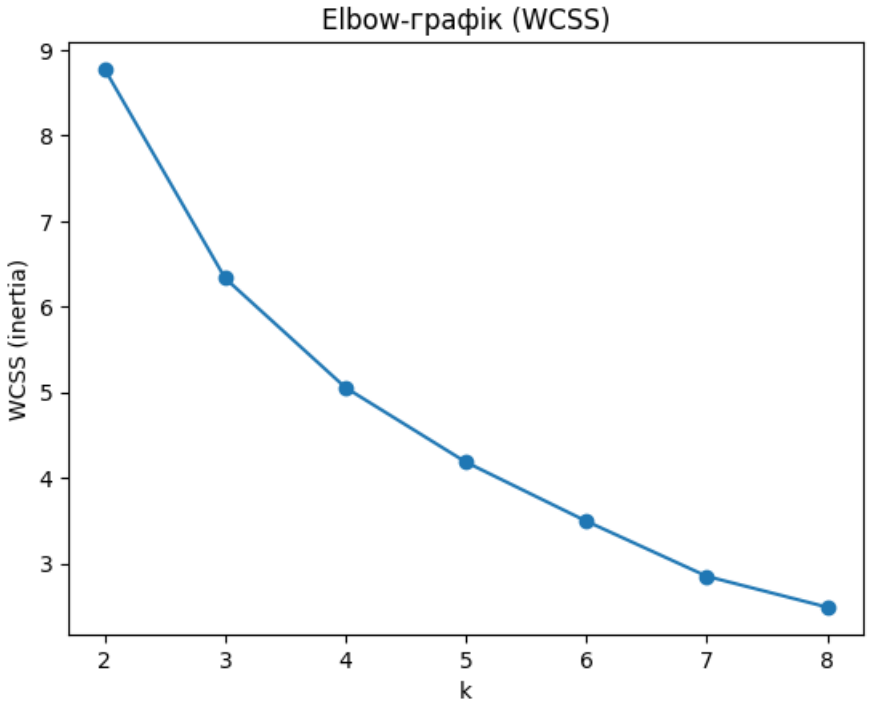

Рисунок 3.2 – Графік ліктя (WCSS)

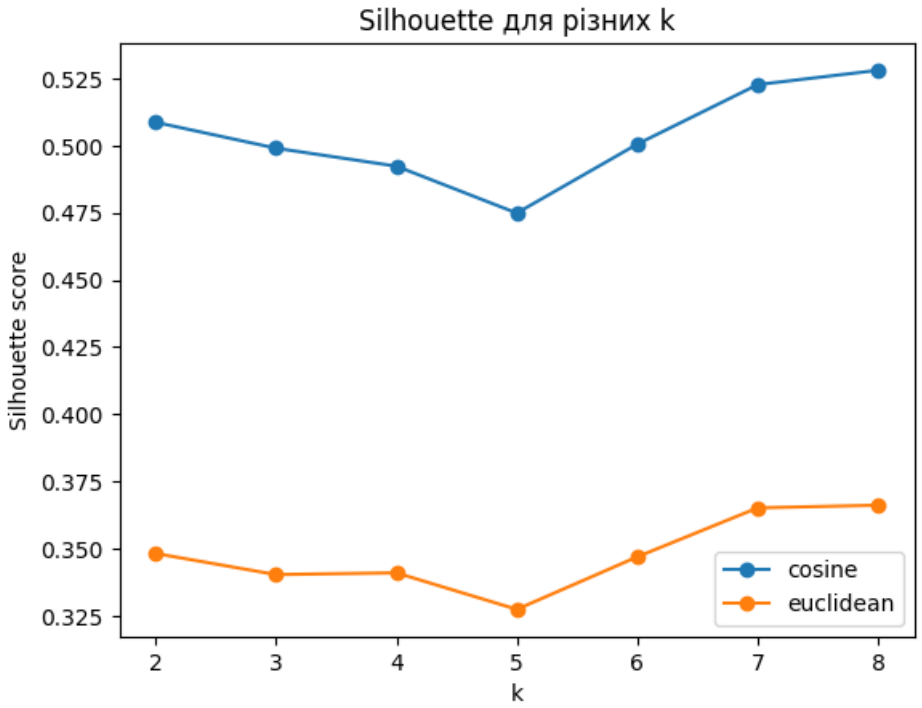

Рисунок 3.3 – Силует-графік для різних k (cosine та euclidean)

На рисунку 3.4 приведено розсіювання (Morning–Evening) з розфарбуванням за кластерами (k=8).

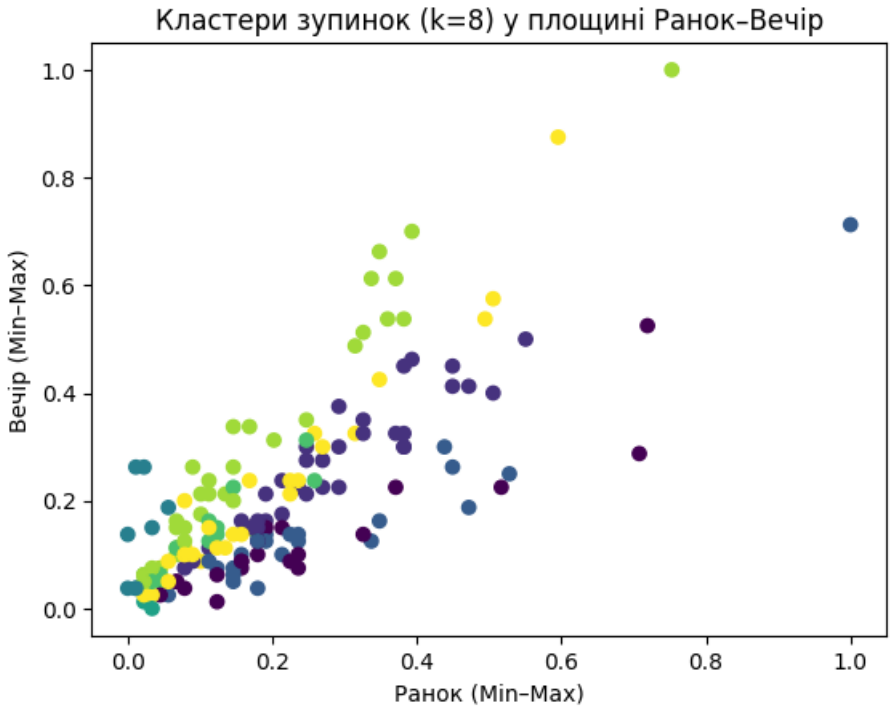

Рисунок 3.4 – Розсіювання за кластерами (k=8) у площині Ранок-Вечір

На рисунку 3.5 приведені середні профілі “ранок–обід–вечір” для кожного кластера (стовпчиковий графік).

В таблиці 3.4 Приведені характеристики кожного кластера за кількістю зупинок, що увійшли до нього, середньою кількістю пасажирів зранку, в обід і ввечері.

Таблиця 3.4 – Підсумок кластерів (k=8)

Кластер	Розмір кластера	Ранок	Обід	Вечір
0	21	22,52	9,9	11,19
1	33	26,73	21,39	23,09
2	22	25,73	21,77	13,36
3	8	2,62	6,5	12,12
4	3	8	15	5
5	13	10,38	4,46	11,69
6	34	16,47	18,41	24,47
7	28	17,54	23,61	18,29

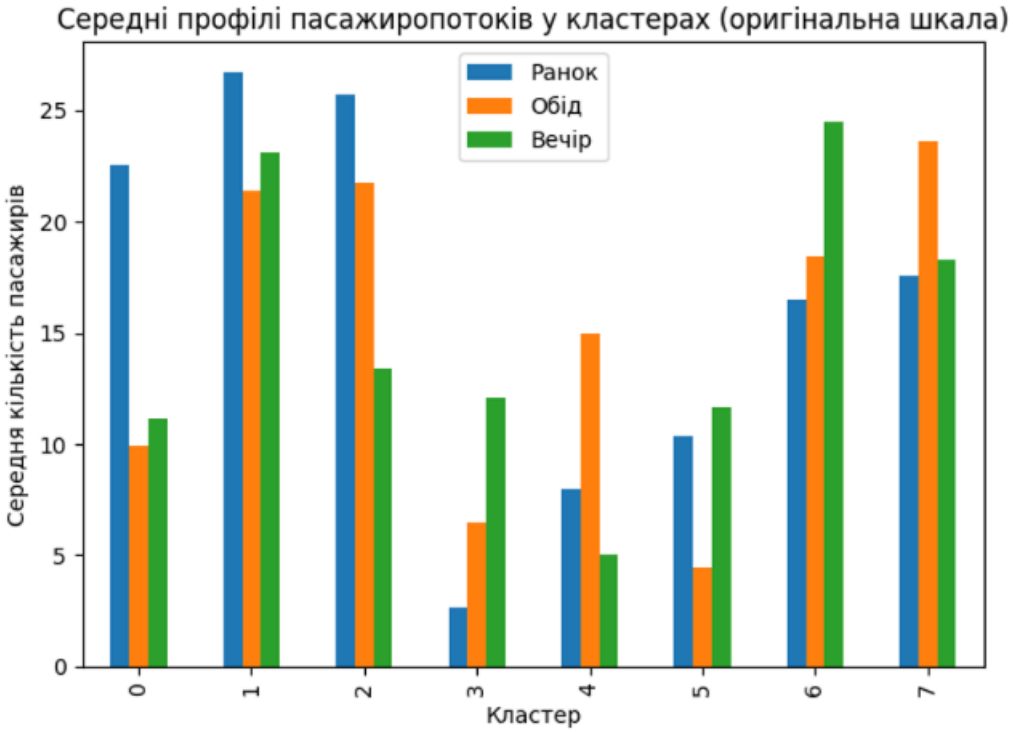

Рисунок 3.5 – Середні профілі “ранок–обід–вечір” для кожного кластера

Таблиця 3.5 демонструє класифікацію кластерів і приклади зупинок.

В додатку Б приведено повну кластеризацію усіх зупинок м. Тернополя.

Таблиця 3.5 – Класифікація кластерів

Кластер	Тип кластера	Приклади (ранок/обід/вечір)
1	2	3
0	Ранково-активні (високий ранок, низькі обід/вечір)	вул.Леся Курбаса (22/11/9); Новий світ (20/10/13); вул. Л. Українки (65/33/43); Кемпінг (64/30/24)
1	Високонавантажені збалансовані (високі значення протягом дня)	вул. Тролейбусна (25/20/23); вул. Ш. Руставелі (від центру) (23/23/23); Галицький коледж (50/44/41); вул. Бережанська (41/36/34)
2	Ранок+обід високі, вечір нижчий (денний профіль)	вул. Винниченка (22/18/12); Орнава (21/25/12); Автовокзал (90/78/58); вул. Симоненка (48/33/21)

Продовження таблиці 3.5

1	2	3
3	Вечірній пік при низьких ранок/обід (периферійні/повернення додому)	Ремзавод (1/6/12); вул. Лозовецька (4/7/13); ТЦ “Епіцентр” (3/13/22); Столярна фабрика (2/11/22)
4	Обідній пік при низьких ранок/вечір (локальні денні поїздки)	Тернопільелектротранс (4/6/1); вул. Весела (3/5/2); Збруч (17/34/12)
5	Низький обід, підвищений вечір (вечірня активність)	Електросвіт (до центру) (11/5/11); вул. Стуса (12/5/13); просп. С. Бандери (23/10/26); с. Острів (24/4/20)
6	Вечірньо-інтенсивні (помірний ранок/обід, високий вечір)	Авторинок (до центру) (14/18/22); Духовний центр (14/16/28); Центральна бібліотека (68/73/81); вул. Збаразька (від центру) (36/43/57)
7	Обідньо-активні (обід вище ранку й вечора)	Савич парк (21/22/18); вул. А. Шептицького (16/27/20); Центральний ринок (54/97/71); Лікарня швидкої допомоги (46/54/47)

3.4.3 Аналіз отриманих результатів

У ході експериментальних досліджень було виконано кластеризацію 162 зупинок громадського транспорту міста Тернополя за показниками середньої кількості пасажирів у ранковий, обідній та вечірній періоди доби. Для забезпечення коректності результатів вхідні дані були попередньо

нормалізовані, а кластеризація виконувалася з використанням алгоритму *k-means*.

Вибір кількості кластерів здійснювався на основі комбінованого використання методу «ліктя» та коефіцієнта силуету. Аналіз залежності суми внутрішньокластерних відстаней від кількості кластерів не виявив чітко вираженої точки зламу, що характерно для реальних соціально-транспортних даних. Водночас максимальне значення середнього коефіцієнта силуету для косинусної метрики було отримано при k=8, що й визначило вибір оптимальної кількості кластерів для подальшого аналізу.

Отримані кластери відрізняються між собою як за рівнем загального пасажирського навантаження, так і за формою добового профілю пасажиропотоку. Аналіз середніх значень показників у межах кожного кластера дозволив виділити типові групи зупинок, зокрема:

- зупинки з низьким та рівномірним пасажиропотоком, характерні для периферійних житлових районів;
- зупинки з вираженим ранковим або вечірнім піком, пов’язані з маятниковими міграціями населення;
- зупинки з високим та відносно рівномірним навантаженням упродовж доби, що відповідають центральним районам міста, навчальним закладам, торговельно-розважальним центрам та транспортним вузлам.

Візуалізація результатів кластеризації у двовимірній проєкції простору ознак підтвердила достатню компактність більшості кластерів і наявність чітких відмінностей між ними. Це свідчить про те, що обрані ознаки адекватно описують добовий профіль пасажиропотоків, а застосований алгоритм кластеризації забезпечує інтерпретовані результати.

Практичне значення отриманих результатів полягає у можливості використання кластерів для подальшої оптимізації транспортної мережі міста, зокрема при коригуванні розкладів руху, перерозподілі транспортних засобів між маршрутами та плануванні інфраструктурних змін. Розроблений

програмний модуль може бути використаний як інструмент підтримки прийняття рішень у сфері міського транспортного планування.

Таким чином, результати експериментального дослідження підтверджують доцільність застосування методів кластерного аналізу для аналізу пасажиропотоків міського громадського транспорту та ефективність розробленого програмного модуля.

У третьому розділі кваліфікаційної роботи здійснено програмну реалізацію та експериментальну перевірку розробленого модуля кластеризації пасажиропотоків міста Тернополя.

Реалізація програмного модуля виконана з використанням мови програмування Python та сучасних бібліотек аналізу даних і машинного навчання. Запропонована структура програмного проєкту є модульною, що забезпечує зручність супроводу, розширюваність та можливість подальшої модифікації системи.

На основі реальних даних щодо середньої кількості пасажирів на 162 зупинках громадського транспорту проведено серію експериментальних досліджень. Для підвищення якості результатів виконано нормалізацію даних та застосовано алгоритм кластеризації k-means із використанням механізму ітераційного підбору кількості кластерів.

Вибір оптимальної кількості кластерів здійснено з використанням методу «ліктя» та коефіцієнта силуету. За результатами експериментів обґрунтовано вибір восьми кластерів, що забезпечують найкращий компроміс між компактністю кластерів та їх інтерпретованістю.

Отримані результати кластеризації підтверджують наявність різних типів зупинок за характером добового пасажиропотоку, зокрема зупинок із переважно ранковою, обідньою або вечірньою активністю, а також зупинок зі стабільно високим навантаженням упродовж дня. Аналіз середніх профілів кластерів дозволив надати змістовну інтерпретацію кожній виділеній групі.

Розроблені засоби візуалізації та проведене тестування підтвердили коректність роботи програмного модуля, стабільність алгоритмів та придатність

отриманих результатів для подальшого використання. Практичне значення роботи полягає у можливості застосування розробленого модуля як інструменту підтримки прийняття рішень у сфері планування та оптимізації міських транспортних систем.

Фрагменти програмного коду розміщені в додатку А.

Таким чином, поставлені у третьому розділі завдання виконано в повному обсязі, а отримані результати підтверджують ефективність розробленого програмного модуля кластеризації пасажиропотоків.

ВИСНОВКИ

У кваліфікаційній роботі розв’язано актуальну задачу аналізу пасажиропотоків міського громадського транспорту шляхом розробки програмного модуля кластеризації зупинок за характером добового навантаження. Актуальність роботи зумовлена необхідністю підвищення ефективності функціонування транспортних систем міст та обґрунтованого прийняття управлінських рішень у сфері міського транспортного планування.

У першому розділі виконано аналіз предметної області та огляд сучасних методів кластерного аналізу, які застосовуються для обробки транспортних і соціально-економічних даних. Розглянуто основні підходи до кластеризації, метрики подібності та програмні засоби реалізації, що дозволило обґрунтувати вибір алгоритму k-means і мови програмування Python для реалізації програмного модуля.

У другому розділі розроблено математичну модель кластеризації пасажиропотоків. Проведено формалізацію вхідних даних, обґрунтовано застосування нормалізації ознак, розглянуто евклідову та косинусну метрики подібності. Детально описано алгоритм k-means, методи вибору кількості кластерів та критерії оцінювання якості кластеризації. Також розроблено архітектуру програмного модуля з механізмом зворотного зв’язку для ітераційного підбору параметрів кластеризації.

У третьому розділі здійснено програмну реалізацію розробленого модуля кластеризації та проведено експериментальні дослідження на основі реальних даних щодо середньої кількості пасажирів на 162 зупинках міста Тернополя. За результатами експериментів обґрунтовано вибір восьми кластерів, що забезпечують найкращу інтерпретованість та якість кластеризації відповідно до коефіцієнта силуету. Проведено візуалізацію та детальний аналіз отриманих кластерів, що дозволило виділити типові групи зупинок за характером добового пасажиропотоку.

Отримані результати підтверджують доцільність застосування методів кластерного аналізу для дослідження пасажиропотоків міського громадського транспорту. Розроблений програмний модуль є гнучким, масштабованим і може бути використаний як інструмент підтримки прийняття рішень при оптимізації маршрутної мережі, коригуванні розкладів руху та плануванні розвитку транспортної інфраструктури.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методичні вказівки до випускних кваліфікаційних робіт освітнього рівня “Бакалавр” спеціальності “Комп’ютерна інженерія”/ О.М. Березький, Г.М. Мельник, Л.О.Дубчак, Ю.М. Батько / Під ред. О.М. Березького. Тернопіль: ЗУНУ, 2021. 52 с.

2. Ілюшин В. В., Король В. Р. Програмний модуль кластеризації пасажиропотоків міста Тернополя. Інтелектуальні комп’ютерні системи та мережі : тези доп. IV Всеукр. наук.-практ. конф. студентів, аспірантів, молодих вчених (20 травня 2026 р.). Тернопіль : ЗУНУ, 2026. С. 117-119. https://ki.wunu.edu.ua/conference/archive/2026_1.pdf#page=117

3. Державна служба статистики України. Чисельність наявного населення України на 1 січня 2024 року. Київ, 2024. 82 с.

4. Офіційний сайт Тернопільської міської ради. Громадський транспорт. URL: https://www.ternopil-city.gov.ua (дата звернення: 15.11.2024).

5. Попович П.В., Маяк М.М., Розум Р.І., Буряк М.В., Березька К.М., Коваль Ю.Б., Мишко С.А. Дослідження стану транспортної інфраструктури міста Тернополя. Центральноукраїнський науковий вісник. Технічні науки. Вип. 7(38), Ч. II. 2023.С. 243-249. https://mapiea.kntu.kr.ua/pdf/7(38)_II/33.pdf

6. ДБН В.2.3-5:2018. Вулиці та дороги населених пунктів. Київ: Мінрегіон України, 2018. 53 с.

7. Шевчук О.С., Березька К.М., Фалович Н.М., Захарчук О.П., Фалович В. А., Мамрош І.М. Визначення рівня завантаження зупинок громадського транспорту на основі кластерного аналізу. Сучасні технології в машинобудуванні та транспорті. 2024, №1(22). С. 357-368. https://doi.org/10.36910/automash.v1i22.1379

8. Гаврилов Є. В., Дмитриченко М. Ф. Системологія на транспорті. Організація дорожнього руху: підручник. Київ: Знання України, 2019. 452 с.

9. Попович П.В., Розум Р.І., Мурований І.С., Буряк М.В., Березька К.М., Петринюк Н.А., Лоїк І.О. Дослідження безпеки дорожнього руху у м. Тернополі. Центральноукраїнський науковий вісник. Технічні науки. Вип. 7(38), Ч. II. 2023. С. 250-256. https://doi.org/10.32515/2664-262X.2023.7(38).2.250-256

10. Шевчук О.С., Захарчук О.П, Фалович Н.М., Березька К.М., Сіран Р. В. Концептуальні основи модернизації транспортної інфраструктури середніх міст в Україні. Сучасні технології в машинобудуванні та транспорті. 2024, №1(22). С. 369-377. DOI: https://doi.org/10.36910/automash.v1i22.1380

11. Березька К.М., Шевчук О.С., Фалович Н.М., Бубняк Ю.Р. Аналіз проблем і математичних методів для їх вирішення в транспортній логістиці. Центральноукраїнський науковий вісник. Технічні науки. Вип. 9(40), Ч. II. 2024. С. 174-181. https://doi.org/10.32515/2664-262X.2024.9(40).2.174-181

12. Шевчук О.С., Березька К.М., Захарчук О.П., Фалович Н.М., Сіран Р.В. Інституційні та нормативно-правові аспекти регулювання сталої міської мобільності. Центральноукраїнський науковий вісник. Технічні науки. Вип. 9(40), Ч. II. 2024. С. 247-256. DOI: https://doi.org/10.32515/2664-262X.2024.9(40).2.247-256

13. Chen C., Skabardonis A., Varaiya P. Travel-time reliability as a measure of service. Transportation Research Record. 2003. Vol. 1855. P. 74-79.

14. Zhang Y., Haghani A. A gradient boosting method to improve travel time prediction. Transportation Research Part C: Emerging Technologies. 2015. Vol. 58. P. 308-324.

15. Zheng Y., Zhang L., Xie X., Ma W.-Y. Mining interesting locations and travel sequences from GPS trajectories. Proceedings of the 18th International Conference on World Wide Web. 2009. P. 791-800.

16. Jiang S., Fiore G. A., Yang Y., Ferreira Jr J., Frazzoli E., González M. C. A review of urban computing for mobile phone traces: Current methods, challenges and opportunities. Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing. 2013. P.

17. Черняк О. І., Захарченко П. В. Інтелектуальний аналіз даних : підручник. – Київ : Знання, 2011. – 599 с.

18. Сергеєв-Горчинський О. О., Іщенко Г. В. Інтелектуальний аналіз даних. Комп'ютерний практикум : навч. посіб. – Київ : КПІ ім. Ігоря Сікорського, 2018. – 156 с.

19. Івахненко О. Г., Лаптєв О. О. Аналіз даних та машинне навчання : навчальний посібник. – Київ : Видавничий дім «Академперіодика», 2016. – 308 с.

20. Клебанова Т. С., Гур'янова Л. С., Чаговець Л. О. Бізнес-аналітика багатовимірних процесів : навч. посіб. – Харків : ХНЕУ ім. С. Кузнеця, 2018. – 272 с.

21. Пономаренко В. С., Мінухін С. В. Інтелектуальні системи підтримки прийняття рішень : навч. посіб. – Харків : ХНЕУ, 2010. – 344 с.

22. Литвин В. В., Висоцька В. А. Інформаційні технології інтелектуального аналізу даних : навчальний посібник. – Львів : Видавництво Львівської політехніки, 2013. – 232 с.

23. Згуровський М. З., Панкратова Н. Д. Основи системного аналізу : підручник. – Київ : Видавнича група BHV, 2007. – 544 с.

24. Jain A. K., Murty M. N., Flynn P. J. Data clustering: a review // *ACM Computing Surveys*. – 1999. – Vol. 31, No. 3. – P. 264–323.

25. Kaufman L., Rousseeuw P. J. Finding groups in data: an introduction to cluster analysis. – New York : John Wiley & Sons, 1990. – 342 p.

26. Jain A. K. Data clustering: 50 years beyond K-means // *Pattern Recognition Letters*. – 2010. – Vol. 31, No. 8. – P. 651–666.

27. Ji Y., Cao Y., Liu Y., Shan X., Li X. Clustering mixed numeric and categorical data with missing values: A novel approach. Information Sciences. 2019. Vol. 479. P. 111-131.

28. Xu R., Wunsch D. Survey of clustering algorithms. IEEE Transactions on Neural Networks. 2005. Vol. 16(3). P. 645-678.

29. Pedregosa F., Varoquaux G., Gramfort A. et al. Scikit-learn: Machine Learning in Python // Journal of Machine Learning Research. – 2011. – Vol. 12. – P. 2825–2830.

30. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. – 2nd ed. – New York : Springer, 2009. – 745 p.

31. James G., Witten D., Hastie T., Tibshirani R. An Introduction to Statistical Learning with Applications in R. – New York : Springer, 2013. – 426 p.

32. Kaufman L., Rousseeuw P. J. Finding Groups in Data: An Introduction to Cluster Analysis. – New York : John Wiley & Sons, 1990. – 342 p.

33. Everitt B. S., Landau S., Leese M., Stahl D. Cluster Analysis. – 5th ed. – Chichester : John Wiley & Sons, 2011. – 346 p.

34. Hill T., Lewicki P. Statistics: Methods and Applications. – Tulsa : StatSoft, Inc., 2006. – 832 p.

35. StatSoft Inc. STATISTICA (data analysis software system), version 10. – Tulsa, USA, 2011.

36. Scikit-learn Developers. Clustering – scikit-learn documentation [Electronic resource]. – Access mode: https://scikit-learn.org/stable/modules/clustering.html (дата звернення: 30.01.2026).

37. R Core Team. R: A Language and Environment for Statistical Computing. – Vienna : R Foundation for Statistical Computing, 2023. – URL: https://www.r-project.org (дата звернення: 30.01.2026).