НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря СІКОРСЬКОГО»
Навчально-науковий фізико-технічний інститут
Кафедра математичних методів захисту інформації

«На правах рукопису»
УДК (519.67)

«До захисту допущено»
Завідувач кафедри
_____ Сергій ЯКОВЛЄВ
«__» _____ 2024 р.

Дипломна робота

на здобуття ступеня бакалавра
за освітньо-професійною програмою
«Математичні методи криптографічного захисту інформації»

зі спеціальності: 113 Прикладна математика
на тему: **«Геопросторовий аналіз та графове дослідження сільської інфраструктури»**

Виконав:
студент IV курсу, групи ФІ-03
Свірш Влада Русланівна _____

Керівник:
Доктор технічних наук, професор
Савчук Михайло Миколайович _____

Консультант:
Доктор технічних наук, професор
Куссуль Наталія Миколаївна _____

Рецензент:
Доктор філософії, старший викладач
Яйлимова Ганна Олексіївна _____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____

Київ — 2024

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря СІКОРСЬКОГО»
Навчально-науковий фізико-технічний інститут
Кафедра математичних методів захисту інформації

Рівень вищої освіти — перший (бакалаврський)
Спеціальність — 113 Прикладна математика,
ОПП «Математичні методи криптографічного захисту інформації»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Сергій ЯКОВЛЄВ
«__» _____ 2024 р.

ЗАВДАННЯ
на дипломну роботу

Студент: Свірш Влада Русланівна

1. Тема роботи: «*Геопросторовий аналіз та графове дослідження сільської інфраструктури*», науковий керівник дипломної роботи: Доктор технічних наук, професор Савчук Михайло Миколайович,

затверджені наказом по університету №__ від «__» _____ 2024 р.

2. Термін подання студентом роботи: «__» _____ 2024 р.

3. Об'єкт дослідження: Розвиток сільських територій України.

4. Предмет дослідження: Геопросторовий аналіз доступності інфраструктури в сільських поселеннях України.

5. Перелік завдань:

– Провести статистичний аналіз інфраструктури України.

– Розробити покращені дескриптори доступності сільських поселень до критичної інфраструктури, такої як медичні, освітні, транспортні та інші об'єкти громадського користування.

– Провести числовий та графічний статистичний аналіз для підтвердження узгодженості між початковими розподілами доступності та графовими представленнями.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: Презентація доповіді

7. Орієнтовний перелік публікацій: Integrated Geospatial Analysis for Rural Development Metrics, CoLInS 2024, Lviv, Ukraine; Geospatial Framework for Enhancing Rural Infrastructure Development, ISAG 2024, Wroclaw, Poland; Enhancing Rural Infrastructure In Ukraine: an Integrated Geospatial Approach, Студентська конференція КПІ 2024, Київ, Україна; Плануються публікації у наукових журналах.

8. Дата видачі завдання: 10 вересня 2023 р.

Календарний план

№ з/п	Назва етапів виконання дипломої роботи	Термін виконання	Примітка
1	Узгодження теми роботи із науковим керівником	01-15 вересня 2023 р.	Виконано
2	Огляд опублікованих джерел за тематикою дослідження	Вересень-жовтень 2023 р.	Виконано
3	Аналіз наявних джерел та підготовка даних	Листопад-грудень 2023 р.	Виконано
4	Розробка дескрипторів доступності інфраструктури	Січень-лютий 2024 р.	Виконано
5	Впровадження графових представлень інфраструктурних зв'язків	Березень 2024 р.	Виконано
6	Статистичний аналіз та перевірка узгодженості даних	Квітень 2024 р.	Виконано
7	Візуалізація та картографування результатів	Квітень 2024 р.	Виконано
8	Підготовка статей та презентацій для виступу на конференціях	Квітень-травень 2024 р.	Виконано
9	Оформлення дипломної роботи	Травень-червень 2024 р.	Виконано

Студент _____ Влада СВІРШ

Керівник _____ Михайло САВЧУК

РЕФЕРАТ

Кваліфікаційна робота містить: 61 стор., 9 рисунки, 3 таблиць, 13 джерел.

Метою даної роботи є розробка та застосування нових методів геопросторового аналізу для оцінки доступності інфраструктури в сільських поселеннях України. У ході дослідження було розроблено нові дескриптори доступності, які враховують реальні зв'язки близькості до критичної інфраструктури. Вперше в Україні застосовано графові моделі для представлення зв'язності сільської інфраструктури, використовуючи геопросторові дані з гібридних джерел, таких як OpenStreetMap та Humanitarian Data Exchange.

Проведено багатофазний аналіз, що включає статистичний профіль базових геопросторових даних, реалізацію спеціалізованих алгоритмів для побудови графових зв'язків та кількісний аналіз для оцінки якості трансформації даних. Основні результати дослідження включають виявлення прогалин та нерівностей у доступності інфраструктури, а також розробку локалізованих карт зв'язків для кожного села, що дозволяє визначити пріоритетні області для інвестицій та розвитку.

Це дослідження сприяє ефективному плануванню інфраструктурних проєктів і підвищенню якості життя в сільських районах України.

ГЕОПРОСТОРОВИЙ АНАЛІЗ, РОЗВИТОК СІЛЬСЬКИХ ТЕРИТОРІЙ, ДОСТУПНІСТЬ ІНФРАСТРУКТУРИ, ГРАФОВІ МОДЕЛІ

ABSTRACT

The qualification work contains: 61 pages, 9 figures, 3 tables, 13 references.

The goal of this work is to develop and apply new geospatial analysis methods for assessing infrastructure accessibility in rural settlements of Ukraine. In this study, new accessibility descriptors were developed, taking into account real proximity connections to critical infrastructure. For the first time in Ukraine, graph models were used to represent the connectivity of rural infrastructure, utilizing geospatial data from hybrid sources such as OpenStreetMap and Humanitarian Data Exchange.

A multi-phase analysis was conducted, including a statistical profile of basic geospatial data, the implementation of specialized algorithms for building graph connections, and a quantitative analysis to evaluate the quality of data transformation. The main results of the study include identifying gaps and inequalities in infrastructure accessibility, as well as developing localized connectivity maps for each village, allowing for the identification of priority areas for investment and development.

This research contributes to effective planning of infrastructure projects and improving the quality of life in rural areas of Ukraine.

GEOSPATIAL ANALYSIS, RURAL DEVELOPMENT, INFRASTRUCTURE ACCESSIBILITY, GRAPH MODELS

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

GDF — GeoDataFrame.

OSM — OpenStreetMap.

HDX — Humanitarian Data Exchange.

POI — Point of Interest, точка інтересу.

SEI — Social-Ecological Index, індекс соціально-екологічної оцінки.

KDE — Kernel Density Estimation, оцінка ядерної густини.

SHDI — Social-Human Development Index, індекс соціально-людського розвитку.

ГІС — Геоінформаційна система.

OCHA — United Nations Office for the Coordination of Humanitarian Affairs, Офіс Організації Об'єднаних Націй з координації гуманітарних питань.

ВСТУП

Ефективна політика розвитку сільських територій залежить від розв'язання ключових проблем, з якими стикаються сільські громади, зокрема прогалин в інфраструктурі та доступу до неї. Це дослідження спрямоване на розробку рішень, заснованих на доказах, через застосування інноваційних методів геопросторового аналізу, спеціально адаптованих для сільських поселень України.

Основні дослідницькі питання включають:

1) Як створити покращені дескриптори доступності сіл до критичної інфраструктури?

2) Чи можуть графові представлення зв'язності сільської інфраструктури зберегти статистичні властивості оригінальних розподілів доступності?

3) Які локалізовані карти зв'язків інфраструктури можуть виявити прогалини та пріоритети розвитку?

Для розв'язання цих питань ми використовуємо різноманітні шари геопросторових даних, здебільшого отриманих з OpenStreetMap. Наші методи включають розробку спеціалізованих алгоритмів для обробки даних та проведення ретельних статистичних тестів. Завдяки цьому, наші інфраструктурні графи ефективно пов'язують тисячі українських сіл із навколишніми об'єктами інфраструктури, надаючи глибоке розуміння рівня доступності та визначаючи можливості для розвитку.

Розвиток сільських територій стикається з численними викликами, включаючи обмежений доступ до ключової інфраструктури. Геопросторовий аналіз, що застосовується в даному дослідженні, забезпечує детальну картину інфраструктурних потреб, що є основою для більш раціонального та ефективного розподілу ресурсів.

Мета дослідження полягає в розробці та впровадженні інноваційної системи геопросторового аналізу для оцінки доступності

сільських поселень до критичної інфраструктури.

Для досягнення мети необхідно розв'язати такі **завдання дослідження**:

1) Провести огляд опублікованих джерел за тематикою геопросторового аналізу та розвитку сільських територій.

2) Розробити методологію створення покращених дескрипторів доступності сільських поселень до критичної інфраструктури.

3) Реалізувати алгоритми для побудови графових зв'язків між сільськими поселеннями та точками інтересу.

4) Провести кількісний та графічний аналіз для оцінки збереження статистичних властивостей оригінальних розподілів доступності.

5) Створити локалізовані карти зв'язків інфраструктури для кожного села та визначити пріоритетні області для розвитку.

Об'єктом дослідження є інфраструктурна забезпеченість та розвиток сільських територій України.

Предметом дослідження є геопросторовий аналіз доступності інфраструктури в сільських поселеннях України.

Для розв'язання поставлених завдань використовувались такі *методи дослідження*: методи геопросторового аналізу, графові моделі, статистичні методи аналізу даних, методи комп'ютерного моделювання.

Наукова новизна отриманих результатів полягає у вперше в Україні розроблених дескрипторах доступності сільських поселень, що враховують реальні зв'язки близькості до критичної інфраструктури. Застосовано графові моделі для представлення зв'язності сільської інфраструктури, використовуючи геопросторові дані з гібрида джерел, таких як OpenStreetMap, Humanitarian Data Exchange та інші.

Практичне значення результатів полягає в тому, що вони можуть бути використані для планування інфраструктурних проєктів, спрямованих на покращення доступності сільських поселень до критичних послуг. Це може сприяти підвищенню якості життя в сільських районах та залученню приватних інвестицій. Ця робота

виконана в рамках проєкту «Інформаційні технології геопросторового аналізу для розвитку сільських територій та громад», номер державної реєстрації: 0123U102838. Даний проєкт виконаний на замовлення Міністерства освіти і науки України, це дослідження спрямоване на розробку та впровадження нових методів покращення інфраструктури сільських територій України.

Апробація результатів та публікації. Основні результати дослідження були представлені на конференціях та опубліковані у наукових журналах:

– Integrated Geospatial Analysis for Rural Development Metrics, CoLInS 2024, Lviv, Ukraine. [5]

– Geospatial Framework for Enhancing Rural Infrastructure Development, ISAG 2024, Wroclaw, Poland. [10]

– Enhancing Rural Infrastructure In Ukraine: an Integrated Geospatial Approach, Студентська конференція КПІ 2024, Київ, Україна. [9]

1 ТЕОРЕТИЧНІ ВІДОМОСТІ

Застосування геопросторового аналізу в розвитку сільських територій зазнало значної еволюції за останні кілька років, що було зумовлено вдосконаленням методів збору даних, технік обробки та дедалі більшою доступністю відкритих джерел даних. Цей розділ оглядає нещодавні внески, які сформували наше сучасне розуміння та методології, зокрема зосереджуючись на роботах, які використовують геопросторовий аналіз, відкриті джерела даних, такі як OpenStreetMap (OSM) [1], та інноваційні аналітичні фреймворки для розв'язання питань розвитку сільської інфраструктури.

1.1 Геопросторова характеристика сільських поселень

У цьому розділі ми досліджуємо еволюцію геопросторового аналізу у розвитку сільських територій, підкреслюючи значні досягнення в цій галузі, як це показано в дослідженні «Геопросторова характеристика сільських поселень і потенційні цілі для ревіталізації за допомогою геоінформаційних технологій» Юксюань Лю та колег [6]. Їх дослідження є ключовим внеском, використовуючи передові методи просторового аналізу, такі як ядерна щільність, просторові автокореляції та регресійний аналіз, для розгляду сільської структури провінції Цзянсі, Китай. Інтегруючи дані дистанційного зондування, топографічні та соціально-економічні дані, Лю та колеги виявляють характерний просторовий розподіл сільських поселень, що характеризується густішими районами на півночі та менш густими на півдні, сформованими фізичними та соціально-економічними чинниками.

Методи аналізу

1) **Аналіз ядерної щільності** (Kernel Density Analysis): Для оцінки розподілу сільських поселень використовується метод ядерної щільності, який дозволяє визначити щільність об'єктів у певному радіусі. Формула ядерної щільності:

$$f(x, y) = \frac{1}{nh^2} \sum_{i=1}^{n} K\left(\frac{d_i}{h}\right)$$

де $f(x, y)$ — оцінка щільності в точці(x, y), n— кількість поселень, h— ширина ядерного вікна, K— ядерна функція, d_i— відстань від i-го поселення до точки(x, y).

2) **Просторова автокореляція** (Spatial Autocorrelation): Метод аналізу автокореляції, такий як індекс Морена (Moran's I), використовується для визначення ступеня взаємозв'язку між об'єктами в просторі. Формула індексу Морена:

$$I = \frac{n}{W} \frac{\sum_i \sum_j w_{ij}(x_i - \bar{x})(x_j - \bar{x})}{\sum_i (x_i - \bar{x})^2} \tag{1.1}$$

де I— індекс Морена, n— кількість поселень, W— сума всіх вагових коефіцієнтівw_{ij}, x_iі x_j— значення змінної для поселень iтаj, $\bar{x}$— середнє значення змінної.

3) **Регресійний аналіз** (Regression Analysis): Для визначення факторів, що впливають на розподіл сільських поселень, використовуються прості та множинні лінійні регресійні моделі. Формула множинної лінійної регресії:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \epsilon$$

де Y— залежна змінна (розподіл поселень), $X_1, X_2, \ldots, X_n$— незалежні змінні (соціально-економічні та екологічні фактори), β_0— вільний член, $\beta_1, \beta_2, \ldots, \beta_n$ — коефіцієнти регресії, ϵ— похибка моделі.

Висновки дослідження

Одним з ключових аспектів роботи Лю та колег є введення Індексу соціально-екологічної оцінки (SEI) — нової метрики для оцінки нерівності розвитку сільських територій та спрямування цільових зусиль щодо ревіталізації. Цей підхід не лише збагачує наше розуміння динаміки сільських поселень, але й пропонує методологічну основу для визначення пріоритетів ревіталізації на основі всебічної оцінки соціально-екологічних чинників. Формула SEI:

$$SEI = \frac{1}{1 + e^{-(a+bX_1+cX_2+dX_3+eX_4)}}$$

де SEI — індекс соціально-екологічної оцінки, X_1— дохід на душу населення, X_2— ресурси орних земель, X_3— середня висота, X_4— середній нахил, a, b, c, d, e— коефіцієнти моделі.

Висновки дослідження щодо розподілу «густа північ – рідкий південь» та розробки SEI представляють методологічний крок вперед у геопросторовому аналізі сільських територій, пропонуючи тонший погляд на виклики та можливості розвитку сільських територій.

Інтеграція з іншими дослідженнями

Дослідження Лю та колег узгоджується з ширшими тенденціями у геопросторовому аналізі, підкреслюючи критичну роль інтеграції екологічних та соціально-економічних даних для формування стратегій розвитку сільських територій. Їх висновки резонують з сучасними дослідженнями, що вивчають просторову гетерогенність інтеграції міських та сільських територій, концептуальне розширення міських досліджень та соціально-просторові нерівності у різних географічних контекстах. Розміщуючи своє дослідження у цьому еволюціонуючому ландшафті, Лю та колеги роблять внесок у більш поінформоване та тонке розуміння просторових моделей сільських поселень, підкреслюючи важливість геопросторового аналізу у створенні цільових та ефективних політик ревіталізації сільських територій.

1.2 Просторово-часовий аналіз глобальних даних про міські будівлі в OSM

На основі геопросторової характеристики сільських поселень, обговореної у розділі 1.1, цей розділ зосереджується на аналізі повноти даних про міські будівлі в системі OpenStreetMap (OSM). Він представляє просторово-часове дослідження для оцінки масштабу та розподілу міських будівельних слідів у глобальному масштабі, вивчаючи відмінності у наявності та якості даних, які можуть вплинути на комплексний міський аналіз та формування політик.

Дослідження, проведене Херфортом та ін. (2023) [4], використовує модель машинного навчання для оцінки повноти будівельного фонду OSM у 13189 міських агломераціях. Модель заснована на регресійному методі випадкових лісів, що дозволяє прогнозувати площу будівельних слідів у кожному з урбаністичних центрів. Формально, модель можна описати наступним чином:

$$\hat{A}_{\text{building}} = f(X) + \epsilon,$$

де $\hat{A}_{\text{building}}$ — прогнозована площа будівель, X— набір предикторів, що включає дані дистанційного зондування, індекс людського розвитку (SHDI), щільність дорожньої мережі та інші змінні, а ϵ— випадкова помилка.

Результати показують, що хоча дані про сліди будівель OSM мають понад 80% повноти для підмножини міських центрів, що представляють 16% міського населення, більшість міст, що охоплюють 48% міського населення, мають менше ніж 20% повноти. Ця різниця підкреслює важливість вирішення нерівностей у даних в рамках платформи OSM для забезпечення неупередженого розуміння розвитку міст.

Для кількісного оцінювання повноти даних будівельного фонду OSM використовується наступна формула:

$$C_{\text{OSM}} = \frac{A_{\text{OSM}}}{A_{\text{total}}},$$

де C_{OSM} — повнота даних OSM, A_{OSM}— площа будівельних слідів, зафіксованих в OSM, A_{total} — загальна прогнозована площа будівель.

Оцінка нерівностей у даних OSM є вирішальною, оскільки вона безпосередньо впливає на використання геопросторової інформації в міському плануванні та досягнення цілей сталого розвитку. Автори вводять комплексний фреймворк для оцінки повноти даних OSM про будівлі, враховуючи такі фактори, як індекс людського розвитку, розмір населення та географічне розташування для виявлення складних моделей просторової упередженості в покритті даних.

Інтегральною частиною аналізу є використання коефіцієнта Джині (G) та коефіцієнта Морана (I) для вимірювання рівномірності та просторової автокореляції відповідно:

$$G = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |x_i - x_j|}{2n \sum_{i=1}^{n} x_i},$$

де x_i і x_j— значення повноти даних у двох різних урбаністичних центрах, n— загальна кількість центрів.

$$I = \frac{n \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij}(x_i - \bar{x})(x_j - \bar{x})}{(\sum_{i=1}^{n}(x_i - \bar{x})^2) \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij}},$$

де w_{ij} — просторові ваги між об'єктами i та j, $\bar{x}$ — середнє значення повноти даних.

Таким чином підкреслюється необхідність інтеграції різних джерел даних та аналітичних методів для всебічного розуміння як сільської, так і міської інфраструктури. Це не лише доповнює сільський фокус попередніх розділів, але й розширює перспективу застосування відкритих геопросторових даних для аналізу інфраструктури на різних масштабах.

Дослідження Херфорт та ін. щодо міських даних OSM про будівлі надає критичну оцінку поточного стану повноти геопросторових даних, закликаючи до більш справедливого розподілу зусиль збирання даних. Такий підхід узгоджується з інноваційним фреймворком, представленим у цій статті, який підкреслює точність, неупередженість та різноманітність даних у геопросторовому аналізі для метрик розвитку сільських територій.

1.3 Геопросторовий аналіз якості життя у сільських районах

Розширюючи тему розвитку сільських територій через геопросторовий аналіз, робота Яйлимової та ін. [11]. представляє алгоритм для оцінки якості життя у сільських районах України. Їх методологія включає комплексну оцінку віддаленості сіл від основної інфраструктури та природних екосистем, а також враховує близькість до зон конфлікту. Цей інноваційний підхід охоплює не лише фізичні, але й соціально-політичні аспекти, що впливають на якість життя у сільських районах.

Методологія оцінки

Методологія дослідження передбачає розрахунок відстаней від кожного села до важливих інфраструктурних об'єктів та природних екосистем. Відстані розраховуються за допомогою географічної інформаційної системи (ГІС) і формують індекс якості життя.

Для кожного села jобчислюється відстань d_{ij}до інфраструктурного об'єктаi. Використовуючи правило трьох сигм (3σ), визначається верхня межа інтервалу:

$$\text{up\_val}_i = \text{mean}_i + 3 \cdot \text{std}_i$$

де mean_i — середнє значення відстаней до об'єктаi, std_i— стандартне відхилення.

Інтервал $[0, \text{up_val}_i]$ділиться на три рівні частини для визначення градації якості життя:

$$\text{Gradation}_{ij} = \begin{cases} 0, & \text{if } d_{ij} \in [0; \frac{\text{up\_val}_i}{3}] \\ 1, & \text{if } d_{ij} \in (\frac{\text{up\_val}_i}{3}; \frac{2 \cdot \text{up\_val}_i}{3}] \\ 2, & \text{if } d_{ij} > \frac{2 \cdot \text{up\_val}_i}{3} \end{cases}$$

Загальна якість життя у селі визначається сумою градацій для всіх показників:

$$\text{Life\_level}_j = \sum_{i=1}^{N} \text{Gradation}_{ij}$$

де N — кількість показників. Кінцева класифікація села за якістю життя:

$$\text{Life\_quality}_j = \begin{cases} \text{Позитивне}, & \text{if Life\_level}_j \in [0; 3] \\ \text{Середнє}, & \text{if Life\_level}_j \in (3; 10] \\ \text{Депресивне}, & \text{if Life\_level}_j > 10 \end{cases}$$

Результати дослідження

Дослідження Яйлимової та ін. вказує на значну різницю в якості життя, причому багато сіл, особливо у східній та південній Україні, стикаються з проблемами, погіршеними конфліктом, що триває. Алгоритмічна оцінка дослідження узгоджується з зусиллями спрямувати ревіталізаційні зусилля туди, де вони найбільше потрібні, пропонуючи даними обґрунтовану основу для прийняття політичних рішень.

У поєднанні з геопросторовими оцінками, представленими у попередніх розділах, це дослідження додатково підкреслює дихотомію між розвитком сільської та міської інфраструктури. Воно також висвітлює гострі виклики, з якими стикаються регіони, постраждалі від війни, представляючи нагальну необхідність цільової інфраструктурної та соціальної інтервенції. Внесок Яйлимової та ін. є таким чином виразним

нагадуванням про складну взаємодію між географією, інфраструктурою та соціально-політичними факторами у формуванні життя у сільських районах.

1.4 Використання геопросторових технологій для соціально-інфраструктурного картографування на рівні сіл

Маряда та Татіпарті [7] представляють дослідження випадку у селі Чіннапендіяла, ілюструючи, як геопросторові технології можуть ефективно картографувати соціальні та інфраструктурні об'єкти на мікрорівні. Поєднуючи просторові та непросторові дані, їхнє дослідження створює детальну геобазу даних, яка служить важливим інструментом для планувальників та політиків у розумінні та задоволенні потреб на рівні місцевих громад.

Їхня методологія демонструє потенціал ГІС у візуалізації та управлінні планами розвитку на рівні сіл, підкреслюючи інтеграцію різних параметрів, таких як зручності, доходи та соціальні показники. Цей підхід узгоджується з ширшою метою сталого та справедливого розвитку сільських територій, демонструючи практичне застосування геопросторових технологій у покращенні умов життя у сільських районах.

Висновки до розділу 1

Загалом, ці дослідження підкреслюють трансформаційну силу геопросторового аналізу у розвитку сільських територій. Вони представляють спектр методологій, від регіональних оцінок до аналізів на рівні окремих сіл, кожна з яких вносить унікальні інсайти у багатогранну природу сільського життя та його покращення через цільові стратегії розвитку. Проте всі вони зосереджуються на одному — дані відіграють важливу роль у формуванні та оцінці рівня розвитку сільських територій.

Тож важливим є детальний аналіз наявних даних, як статистично та аналітично, а також отримання нових можливих описів даних що корелювали б з найпростішими можливими описами, і як можна було побачити в даних дослідженнях значущу роль відіграє взаєморозміщення даних точок інфраструктури, через що в наступних розділах спробуємо ввести та перевірити на коректність графовий тип опису даних.

2 ТЕОРЕТИЧНІ ОСНОВИ ОБРОБКИ ТА АНАЛІЗУ ГЕОПРОСТОРОВИХ ДАНИХ

Як було показано в першому розділі, дані для подальших є значущими, а отже в даній частині роботи будуть розглянуті наявні дані, і розглянемо методи, за допомогою яких буде проведено аналіз та обробка цих дані.

2.1 Матеріали

Для цього аналізу різні геопросторові шари даних були вилучені з OpenStreetMap (OSM) [1] у GeoDataFrame (GDF). OpenStreetMap (OSM) — це спільний проєкт зі створення редагованої карти світу, що будується волонтерами за допомогою аерофотозйомок, GPS-пристроїв та низькотехнологічних польових карт [1]. Дані OSM є відкритими та включають всебічне глобальне покриття доріг, будівель, природних об'єктів, що надають багатий набір даних для геопросторового аналізу. У нашому дослідженні шари даних, вилучені з OSM, включають основні, вторинні та сільські дороги; класифікацію земельного покриву; та місцезнаходження шкіл, коледжів, університетів, готелів, лікарень, клінік, аптек, супермаркетів, торгових центрів, банків, церков, бібліотек, дитячих садків та місцевих, національних і регіональних парків.

Додаткові дані про розташування сіл та міст надійшли з Humanitarian Data Exchange (HDX) [3], відкритої платформи для обміну даними про кризи та країни. HDX керується Офісом ООН з координації гуманітарних питань (OCHA). Для нашого аналізу використовувалися дані HDX про розташування населених пунктів в Україні станом на середину 2021 року. Повний список геопросторових даних по Україні, використаних для нашого аналізу, представлений у Таблиці 2.1.

Дані	**Шари**	**Джерело**
Села	Села	The Humanitarian Data Exchange (станом на 17.07.2021) [3]
Міста	Міста	The Humanitarian Data Exchange (станом на 17.07.2021) [3]
Елеватори	Елеватори	Elevators in Ukraine (станом на 23.02.2022) [2]
Дороги	Національні дороги, районні дороги, сільські дороги	OSM [1]
Освіта	Школи, коледжі, університети	OSM [1]
Готелі	Готелі, гостьові будинки, притулки	OSM [1]
Медицина	Лікарні, клініки, аптеки	OSM [1]
Магазини	Супермаркети, торгові центри, магазини одягу, ринки	OSM [1]
Банки	Банки	OSM [1]
Церкви	Церкви	OSM [1]
Бібліотеки	Бібліотеки	OSM [1]
Дитячі садки	Дитячі садки	OSM [1]
Парки	Місцеві парки, національні парки, регіональні парки	OSM [1]
Пошта	«Нова пошта» та «Укрпошта»	Ресурси поштових операторів [12, 13]
Мобільні мережі	«Київстар», «Vodafone», «lifecell», «Trimob»	Mobua.net [8]

Таблиця 2.1 – Геопросторові дані

2.2 Методи

Методи, які були розглянуті та описані можна поділити на 3 типи, а саме:

1) Методи аналізу геопросторових даних.

2) Методи обробки геопросторових даних.

3) Методи візуалізації геопросторових даних.

2.2.1 Методи аналізу геопросторових даних

Першим розділом, позначимо методи аналізу геопросторових даних. Для підтримки всебічної оцінки геопросторових наборів даних, пов'язаних з сільською інфраструктурою в Україні, використовуються різноманітні числові та графічні статистичні методи. Поєднання кількох підходів забезпечує ретельний, всебічний погляд на розподіли даних.

Також широко використовуються гістограми в аналітичному процесі. Візуалізація частоти метричних відстаней до інфраструктури серед сіл на гістограмах швидко передає форму їх розподілів. Гістограми допомагають оперативно визначити нормально розподілені особливості у порівнянні з тими, що мають зсув у доступності до вищих чи нижчих значень. Викиди також чітко проявляються у гістограмах. Наприклад, сільське поселення розташоване аномально далеко від найближчої медичної клініки у порівнянні з більшістю інших сіл.

Формально, гістограма визначається як графік, що показує розподіл набору даних. Вона побудована за формулою:

$$H(x) = \frac{1}{n} \sum_{i=1}^{n} I\left(x_i \in [x_j, x_{j+1})\right)$$

де $H(x)$ — висота стовпчика гістограми для інтервалу$[x_j, x_{j+1})$, n — загальна кількість даних, I — індикаторна функція, яка дорівнює 1, якщо значення x_i потрапляє у відповідний інтервал, і 0 в іншому випадку.

Доповнюючи графічні інсайти гістограм, коробкові діаграми лаконічно представляють внутрішнє розповсюдження даних, що лежить в основі цих розподілів: ключові квартилі, викиди та діапазон екстремальних значень. Порівняння коробкових діаграм дозволяє швидко розпізнати медіанні схожості та відмінності між категоріями інфраструктури. Наприклад, можна виявити, яка частина українських сіл знаходиться в межах 10 км від школи та яка середня відстань до великого міста.

Коробкові діаграми (або діаграми розмаху) показують п'ять основних характеристик розподілу:

1) Нижній квартиль ($Q1$)
2) Медіана ($Q2$)
3) Верхній квартиль ($Q3$)
4) Мінімальне значення (нижній ус)
5) Максимальне значення (верхній ус)

Коробкову діаграму можна формально описати як:

$$\text{IQR} = Q3 - Q1$$

де IQR — міжквартильний розмах, який представляє центральну частину розподілу даних.

У той час як графічні підходи вказують на властивості розподілу, кореляційні матриці безпосередньо виявляють зв'язки між доступністю лікарень, шкіл, доріг та інших об'єктів. Кореляційні коефіцієнти виявляють аспекти сільської інфраструктури з найтіснішими зв'язками, спрямовуючи глибші дослідження у ці вирівняні прогалини в доступності.

Кореляція між двома змінними X та Y визначається як:

$$\rho_{X,Y} = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y}$$

де $\text{cov}(X, Y)$ — коваріація між X та Y, σ_X та σ_Y — стандартні відхилення X та Y.

На додаток до навчання на основі візуалізації даних, використовуються дві фундаментальні статистичні функції. Підсумкові статистичні дані, такі як середні значення, медіани та стандартні відхилення, підкреслюють центральні тенденції та варіації для відстаней до інфраструктури та додаткових особливостей. Була використана бібліотека Pandas для Python, яка пропонує зручні вбудовані функції для обчислення описової статистики через синтаксис на кшталт `dataframe.describe()`.

Основні описові статистики можна формально визначити як:

– Середнє значення:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

– Медіана — значення, яке розділяє набір даних на дві рівні частини

– Стандартне відхилення:

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2}$$

Ці статистичні описи надають важливу інформацію про центральні тенденції та розсіювання даних. Наприклад, середнє значення вказує на типове значення в наборі даних, тоді як медіана дає уявлення про центральну точку розподілу. Стандартне відхилення описує, наскільки значення в наборі даних відхиляються від середнього, що важливо для оцінки варіації у доступності інфраструктури.

2.2.2 Методи обробки геопросторових даних

По-перше, найпростішим і найзрозумілішим є банальний пошук дистанції між селом та різноманітними точками інфраструктури, дана логіка була розглянута в рамках дослідження [11]. Проте, наразі є прагнення розширити даний опис, імплементувавши нові дескриптори даних. Основна суть даного нововведення полягає в отриманні інформації про 5 найближчих об'єктів в певному радіусі від села. Для цього недостатньо просто використати вбудовані методи бібліотеки, оскільки дані методи не роблять пошук ефективно, а починають робити повний перебір, що при 28000 тисяч сіл та 50000 об'єктів інфраструктури означає розрахунок 1.4 мільярдів відстаней, що потребує 388 годин обробки при обрахунку 1000 відстаней в секунду. Тому потрібно було ввести власний метод обробки даних, що дозволить отримати результати в коротший час.

Методи створення розширеного набору даних з новими дескрипторами

Для розробки більш комплексного набору даних з додатковими дескрипторами для сільських районів України ми впроваджуємо систематичний процес виявлення та інтеграції точок інтересу (POI) з вихідних геопросторових шарів даних у групи для кожного села, представлені у форматі графу. Для кожного типу POI спочатку сегментуємо всі об'єкти на буферні коробки однакового заздалегідь визначеного розміру, що відповідає максимальній відстані, визначеній для кожного типу у вхідних даних (приклад буде наведено в наступній частині). Після створення цих квадратних буферів переходимо до виявлення найближчих об'єктів для кожного села. Це включає визначення, у якому буфері розташоване село, та визначення набору з восьми сусідніх коробок. Процес буферизації POI та визначення буферів пошуку ілюструється на Рисунку 2.1, а процес розрахунку відстаней до POI та визначення найближчих описаний на Рисунку 2.2.

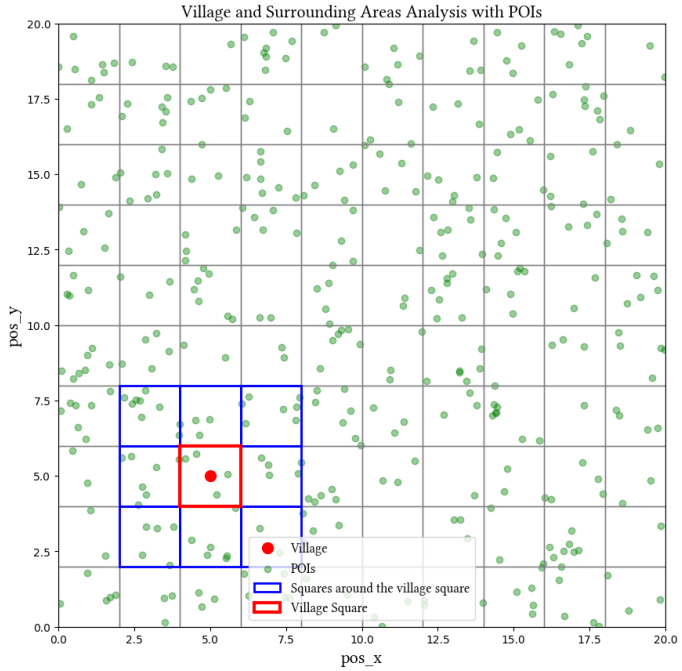

Рисунок 2.1 – Ілюстрація методології, використаної для визначення регіонів для просторового аналізу.

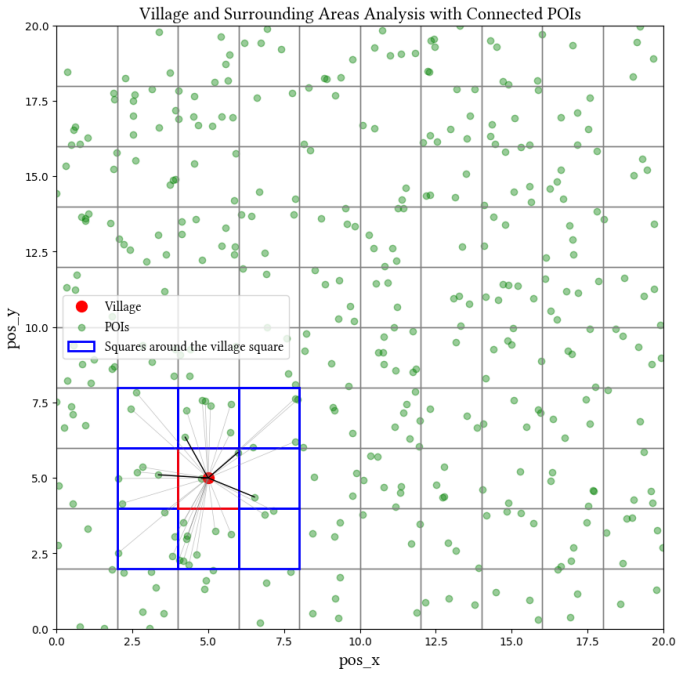

Рисунок 2.2 – Цей рисунок демонструє розрахунок відстаней у визначених регіонах, відповідно до методології.

Цей підхід дозволяє створити всеосяжний набір POI поблизу кожного села, охоплюючи як різноманітність найближчих об'єктів (включаючи міста, парки тощо), так і забезпечуючи їх розташування на доступній відстані. Систематично каталогізуючи найближчі POI різних типів до кожного сільського поселення, розробляємо розширені дескриптори, що охоплюють доступність і наявність ключової інфраструктури та послуг.

Кінцевим результатом є набір даних у структурі графу, що з'єднує села з їх найближчими POI по визначених категоріях разом з метриками відстані. Цей вихідний набір даних надає детальні інсайти щодо доступу до важливих зручностей на національному рівні. Структура графу, яка з'єднує сільські поселення з найближчими POI різних типів через зв'язки на основі відстані, формує основу наших розширених геопросторових дескрипторів для аналізу сільської інфраструктури.

Основні кроки включають:

– Розбиття даних POI на блоки. Кожен блок має розмір, що відповідає максимальній відстані, визначеній для кожного типу POI.

– Визначення найближчих об'єктів для кожного села. Це включає визначення, у якому блоці розташоване село, та аналіз восьми сусідніх блоків.

– Обмеження кількості об'єктів до заздалегідь визначеної кількості найближчих об'єктів для кожного села.

Формула для обчислення евклідової відстані між селом і точкою інтересу (POI) виглядає наступним чином:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

де d — відстань між селом і POI, (x_1, y_1) — координати села, а (x_2, y_2) — координати POI.

Цей підхід дозволяє значно скоротити час обчислень і забезпечує точність результатів. Важливо відзначити, що для побудови цих графів було взято за ідею ефективні алгоритми пошуку найближчих сусідів, такі як KD-дерева (k-d trees) та Ball-дерева, які забезпечують пошук у логарифмічний час:

$$O(n \log n)$$

де n - кількість об'єктів у наборі даних.

В цьому ж випадку, в новій імплементації кількість об'єктів для аналізу зменшилась до 500 об'єктів (в середньому) для кожного села, що покращило швидкість обчислення в 100 разів, додатково покращивши обробку за допомогою паралельного виконання на 8 ядрах процесора отримано додаткове покращення в близько 5 разів, і на обраній системі обчислення займали 0.75 години, що є прийнятним часом для такого обсягу даних.

Висновки до розділу 2

В даному розділі розглянуто та описано, основну теорію методів та матеріалів роботи з геопросторовими даними. Було описано матеріали якими оперуємо в рамках даної роботи, а також методи за допомогою яких робимо оцінку даних та їх розподіл, а також розглянули метод створення нових дескрипторів.

3 ЕКСПЕРИМЕНТАЛЬНИЙ АНАЛІЗ ТА ЗАСТОСУВАННЯ МЕТОДІВ ГЕОПРОСТОРОВОГО АНАЛІЗУ

На результатах першого розділу, де було визначено на основі попередніх досліджень, що аналіз наявних даних є важливим та необхідним, в другому розділі розглянуто доступні дані, методи аналізу і методи створення нових дескрипторів. В даному ж розділі, застосовуються отриманні знання для експериментального аналізу даних та експериментального застосування методу пошуку точок інфраструктури в сільських районах.

Основною метою експериментального зусилля є розробка розширеного набору геопросторових дескрипторів для представлення сільської доступності до критичної інфраструктури на основі моделі графової бази даних. Для досягнення цього проведемо багатофазний аналіз, що включає:

1) Статистичний профіль базових геопросторових наборів даних про сільські території України для перевірки цілісності та узгодженості з пріоритетами розвитку.

2) Експериментальну реалізацію спеціалізованих алгоритмів для побудови графових зв'язків між сільськими поселеннями та навколишніми багатокатегорійними точками інтересу (POI), що представляють доступну інфраструктуру.

3) Кількісний та графічний статистичний аналіз для оцінки, чи зберігає графова трансформація якісні властивості попередніх розподілів доступності сільських територій.

Вихідним результатом є вдосконалений графовий набір даних, що кількісно оцінює доступність на рівні сіл до ключових зручностей у межах критичних порогів відстані. Що закладає фундамент для майбутньої аналітики для визначення прогалин у сільській інфраструктурі, що обмежують розвиток.

3.1 Перевірка базових геопросторових даних

Спочатку перевіримо повноту та дійсність зібраних геопросторових шарів даних про сільську інфраструктуру в Україні. Для цього було обрано використання GeoDataFrames (GDF), точки яких можна побачити на Рисунку 3.1.

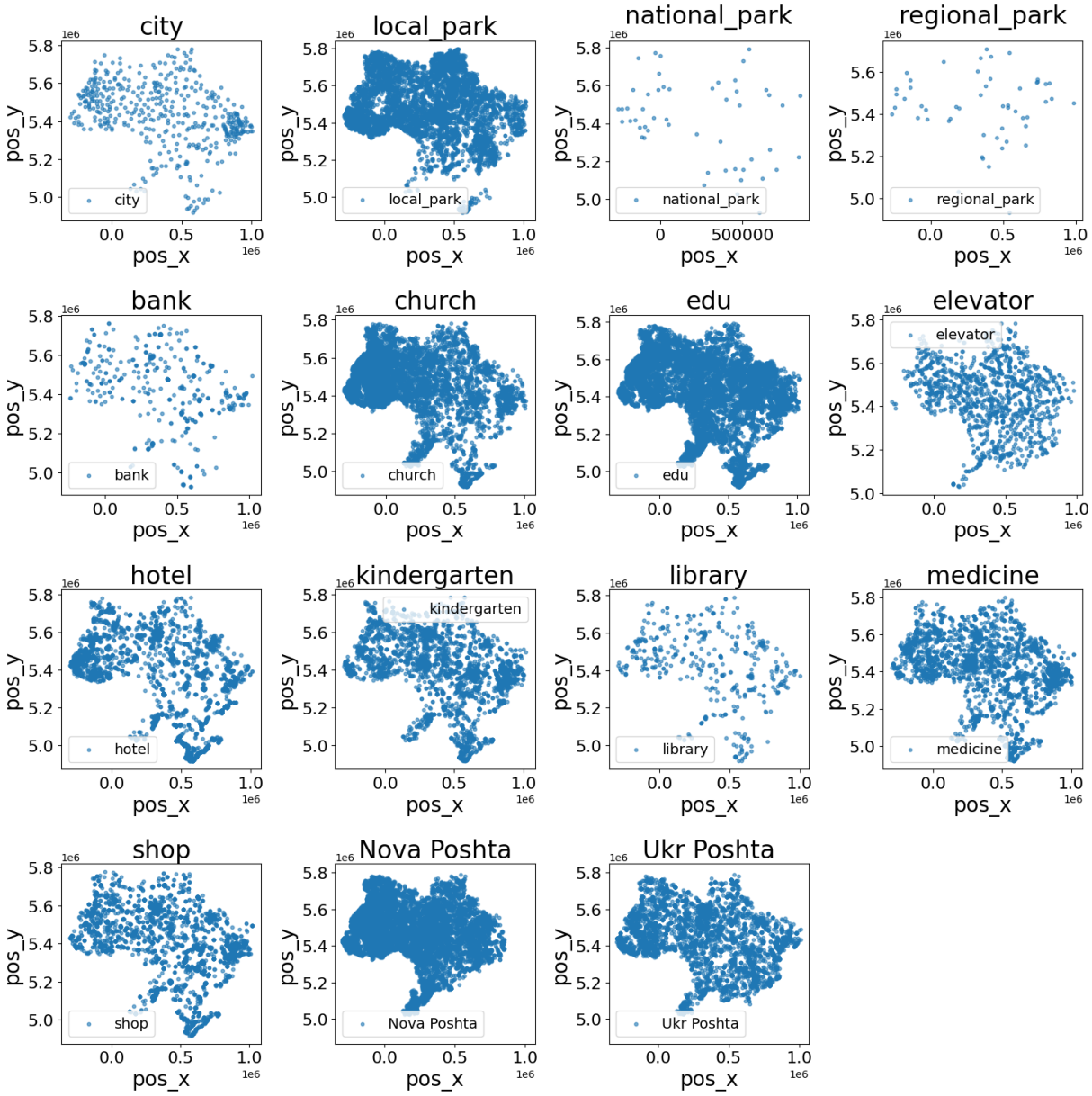

Рисунок 3.1 – Графічне представлення груп точок інтересу (POI). Кожний рисунок представляє один тип POI, як категоризовано в Таблиці 2.1.

Після огляду стає очевидним, що більшість цих графіків демонструють більш-менш нормальний розподіл по Україні. «Нормальний» у цьому контексті означає, що немає значних прогалин, смуг або будь-яких інших відсутніх областей у даних, окрім даних поштових операторів, в зв'язку з закриттям своїх відділень через війну на сході України. Після аналізу цього набору даних і перевірки його точності переходимо до наступних кроків.

3.2 Створення дескриптора доступності сільських територій на основі графових даних

Маючи чисті базові дані, реалізуємо запропоновану методологію сегментації зони обслуговування сільських територій, вилучення та відбору POI, розрахунку відстаней та побудови графової бази даних для перетворення попередніх наборів даних у розширені дескриптори доступності. Як було описано в секції 2, процес створення графової бази даних включає декілька ключових етапів. Спочатку визначимо всі можливі точки інтересу (POI) в заданому радіусі навколо кожного села. Далі, використовуємо алгоритми просторового аналізу для розрахунку евклідових відстаней між селами та кожною POI, забезпечуючи тим самим точні географічні зв'язки. Ці відстані конвертуються в атрибути графових вершин і ребер, що дозволяє створити деталізовану мережу доступності. Ця мережа відображає реальні шляхи доступу до різних об'єктів інфраструктури і забезпечує глибоке розуміння просторових структур та можливих прогалин у наявності критичних ресурсів. Експеримент передбачає використання таблиці, яка описує, які дані були включені у новостворені дескриптори. Для кожного GDF визначено конкретні значення, що визначають, які стовпці повинні бути включено у вихідні дані. Ці деталі можна знайти у Таблиці 3.1.

Дані	**Опис у вихідному графі**	**Максимальна відстань**	**Максимальна кількість всередині**
Міста	Id, pos_x, pos_y, distance	50 км	5
Місцеві парки	Id, type, area, pos_x, pos_y, distance	30 км	5
Національні парки	Id, area, pos_x, pos_y, distance	50 км	5
Регіональні парки	Id, area, pos_x, pos_y, distance	50 км	5
Банк	Id, pos_x, pos_y, distance	30 км	5
Церква	Id, type, pos_x, pos_y, distance	30 км	5
Освіта	Id, type, pos_x, pos_y, distance	30 км	5
Елеватори	Id, pos_x, pos_y, distance	30 км	5
Готелі	Id, type, pos_x, pos_y, distance	30 км	5
Дитячі садки	Id, pos_x, pos_y, distance	30 км	5
Бібліотеки	Id, pos_x, pos_y, distance	30 км	5
Медицина	Id, type, pos_x, pos_y, distance	30 км	5
Магазини	Id, type, pos_x, pos_y, distance	30 км	5

Таблиця 3.1 – Схема створення

3.3 Кількісна оцінка відстані початкових данних

Даний тип оцінки сіл був частково попередньо розрахований членами проекту [11] і включає різні атрибути інфраструктури, детальний опис яких наведено у Таблиці 3.2, частково розрахований в рамках даної роботи.

Збір та обробка цих даних дозволили створити детальну картину доступності сільських територій до критичної інфраструктури, такої як дороги, міста, елеватори, освітні та медичні заклади, а також інші важливі об'єкти. За допомогою найпростішого типу опису, що дозволяє закласти певний найпростіший опис доступності інфраструктури, розкриваються основні характеристики та кореляції, які можемо використати в наступних розділах при аналізі новостворених дескрипторів доступності.

Назва	**Опис**
RD_m1_NEAR	Відстань до найближчої основної дороги
RD_m2_NEAR	Відстань до найближчої регіональної дороги
RD_m3_NEAR	Відстань до найближчої сільської дороги
CITY2_NEAR	Відстань до найближчого міста
elevators_	Відстань до найближчого елеватора
Kiyv_NEAR_	Відстань до столиці України, Київ
LokPark_NE	Відстань до найближчого місцевого парку
regPark_NE	Відстань до найближчого регіонального парку
kinder_NEA	Відстань до найближчого дитячого садка
bank_NEAR_	Відстань до найближчого банку
cerkva_NEA	Відстань до найближчої церкви
education_	Відстань до найближчого освітнього закладу
hotels_NEA	Відстань до найближчого готелю
library_NE	Відстань до найближчої бібліотеки
likarni_NE	Відстань до найближчої лікарні
magaz_NEAR	Відстань до найближчого магазину
NP_Min_Dist	Відстань до найближчого відділення Нової Пошти
UP_Min_Dist	Відстань до найближчого відділення Укрпошти

Таблиця 3.2 – Відстані до найближчих об'єктів

3.4 Графові дескриптори доступності

В результаті успішного виконання експерименту в частині 3.2 було отримано графові дескриптори доступності для сільських територій. В результаті трансформації початкових оцінок відстаней у детальне графове з'єднання між сільськими селами та навколишніми багатокатегорійними POI, успішно сконструйовано розширені дескриптори доступності. В результаті розроблено структуру, що відображена нижче на вставці 3.4.1.

Зауваження. Коли потрібні додаткові дескриптори в рамках певної категорії інфраструктури, такої як регіональні парки, у графовій структурі вводяться нові ключові значення. Це розширення забезпечує всебічність графу, як зазначено в Таблиці 3.1.

3.4.1 Приклад графового дескриптора доступності


```
{
    "id_type": "admin4Pcod",
    "id": "UA2111000000",
    "distance": 3111.931012554032,
    "pos_x": 52060.12938924221,
    "pos_y": 5420926.907282681
},
{
    "id_type": "admin4Pcod",
    "id": "UA2110100000",
    "distance": 23160.7396750182,
    "pos_x": 60818.384628206666,
    "pos_y": 5441084.188000026
},
{
    "id_type": "admin4Pcod",
    "id": "UA2123210100",
    "distance": 41187.70115749954,
    "pos_x": 76317.64176459657,
    "pos_y": 5451978.68918336
},
{
    "id_type": "admin4Pcod",
    "id": "UA2110400000",
    "distance": 41376.3746009097,
    "pos_x": 90058.87699085698,
    "pos_y": 5416411.645983889
},
{
    "id_type": "admin4Pcod",
    "id": "UA2110200000",
    "distance": 43577.79427222703,
    "pos_x": 80594.92931475205,
    "pos_y": 5391219.696301765
}
```


3.5 Статистичний аналіз

Одним з початкових висновків цього статистичного аналізу є ілюстрація на Рисунку 3.2. Досліджуючи RD_m2_NEAR та RD_m3_NEAR, видно, що майже всі села мають задовільний доступ до

доріг будь-якого типу. Однак, є значний простір для покращення в RD_m1_NEAR, який демонструє право зміщений розподіл. Kyiv_NEAR показує трикутний розподіл, що вказує на те, що більшість сіл розташовані на відстані 150–500 км, що сприятливо, оскільки це свідчить про рівномірний доступ до столиці та, відповідно, до можливостей бізнесу з компаніями, які там базуються. Розподіли інших змінних ілюструють схожість на логнормальний розподіл, що вказує на те, що хоча деякі села розташовані близько до POI, багато інших мають простір для покращення в плані розташування та доступу до різних зручностей.

Зауваження. Схожість в даному контексті була отримана, як результат проведення тесту в підборі параметрів, для нормального, логнормального гамма та експоненціального розподілів. Після підбору параметрів і проведення тесту Колмогорова-Смірнова, було визначено, що навіть для найкращого підходу, значення p досить низьке $(0.05 - 0.1)$, що не дозволяє казати про те, що дані розподіли повторюють обрані розподіли точно, проте можна стверджувати про їх схожість.

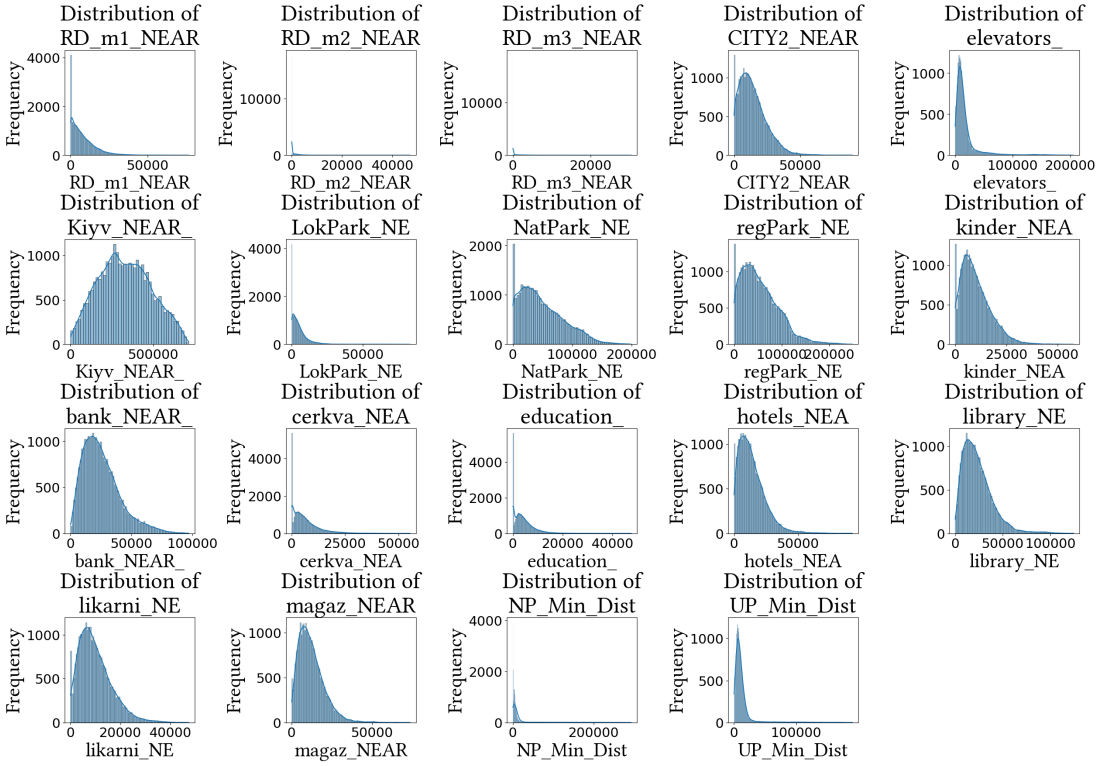

Рисунок 3.2 – Гістограма відстані для кожного типу інфраструктури.

Додатковий аналіз, що показано на Рисунку 3.3, надає підтвердження відносно коректності використаних відстаней при побудові графових дескрипторів, а саме в таблиці 3.1 колонці «Максимальна відстань». Коробкові діаграми чітко зображують розподіл відстаней.

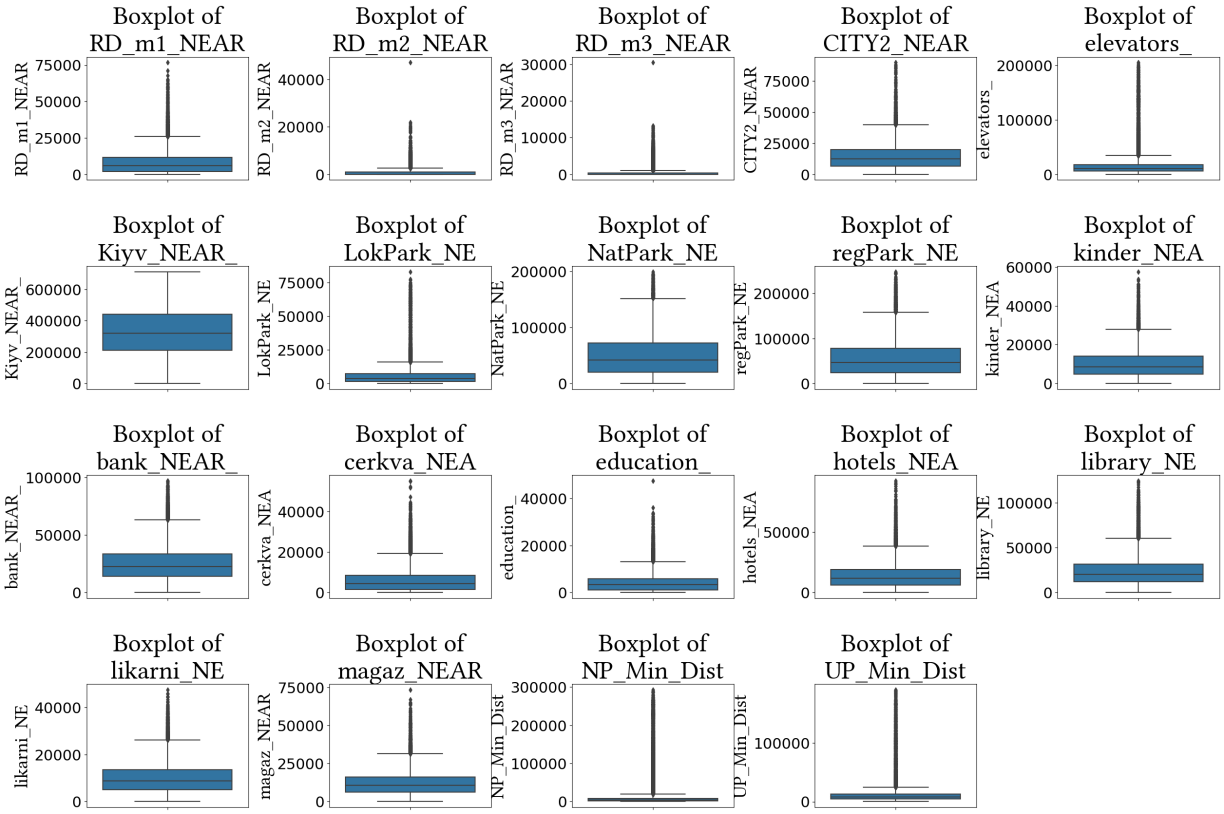

Рисунок 3.3 – Коробкові діаграми досліджуваних колонок.

Наступним кроком розглядаємо кореляції між дескрипторами відстаней, що зображені на Рисунку 3.4. Важливим спостереженням є кореляція таких об'єктів, як дитячі садки, банки, церкви та освітні послуги, з близькістю до найближчого міста, що логічно, оскільки міста зазвичай пропонують більше та кращі зручності, ніж села. Додатковий статистичний аналіз показав, що інші категорії інфраструктури, такі як лікарні та магазини, також демонструють високий рівень кореляції з міською близькістю. Це свідчить про те, що сільські райони, які мають кращий доступ до міської інфраструктури, мають значні переваги в доступі до ключових послуг. Крім того, результати показують, що

зручності, які вважаються менш критичними, такі як місцеві парки та готелі, також демонструють значні кореляції з міською близькістю. Це підкреслює важливість комплексного підходу до розвитку сільської інфраструктури, включаючи як критичні, так і некритичні послуги, для забезпечення високої якості життя в сільських громадах.

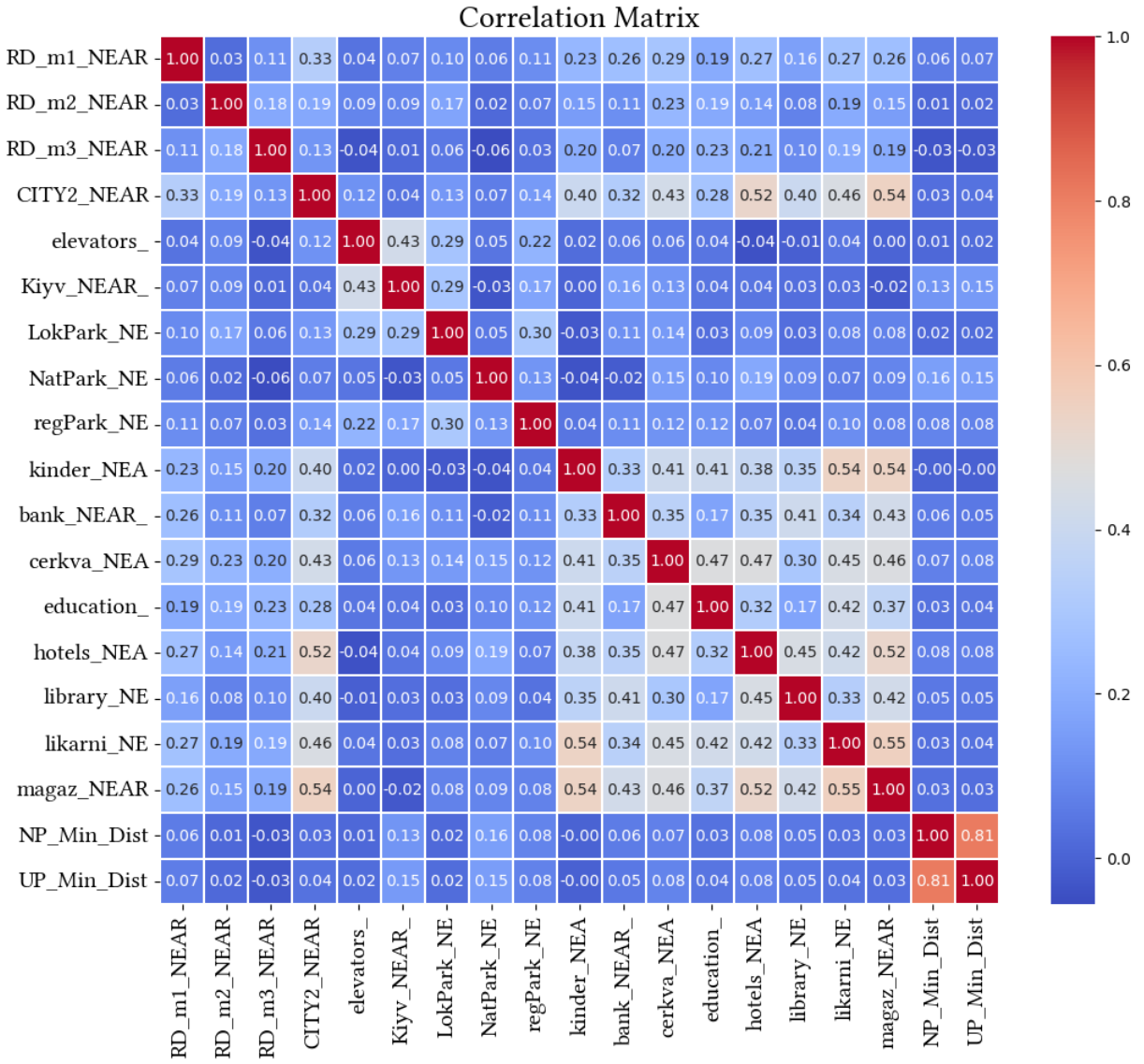

Рисунок 3.4 – Кореляційна матриця раніше розглянутих колонок.

Як було сказано раніше, важливо, щоб новостворені дані не втрачали цілісність або не демонстрували широко варіативні розподіли. Рисунок 3.4 підтверджує продовження актуальності доступності соціальних зручностей. Далі потрібно перевірити, чи залишаються

розподіли даних послідовними або майже такими, створюючи графіки розподілу, як показано на Рисунку 3.5. Хоча ці графіки містять багато даних та містять розбіжності, все ж можна спостерігати, що загальний розподіл не змінився суттєво в новому методі опису даних. Це свідчить про те, що наш підхід забезпечує стабільність і надійність отриманих результатів, зберігаючи ключові характеристики початкових даних.

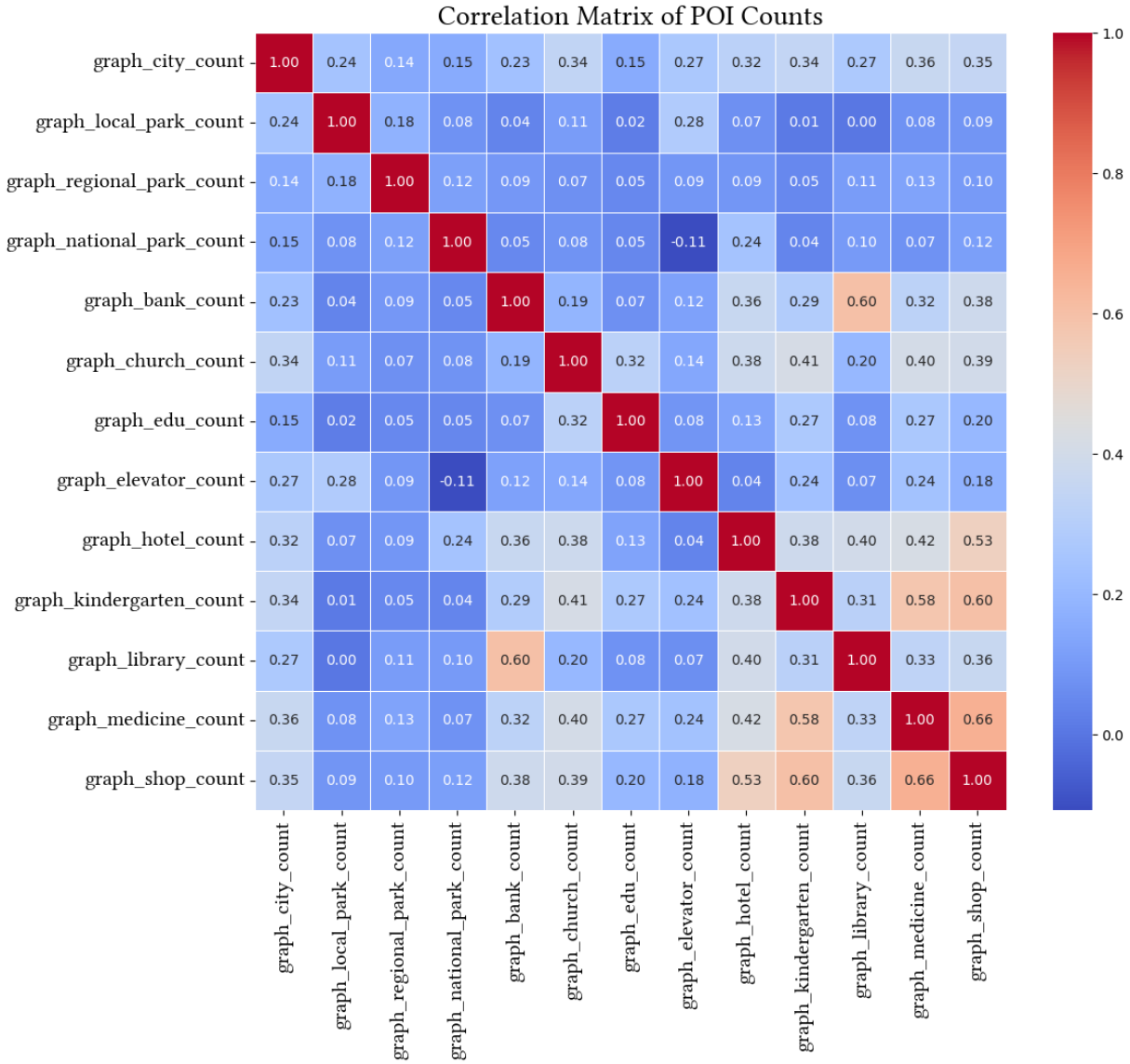

Рисунок 3.5 – Кореляційна матриця новостворених дискрипторів.

Також необхідно оцінити чи зберігається структура в розподілах в графових структурах, а саме чи стверджується факт того, що села з більшою кількістю POI в радіусі серед однієї групи будуть мати перший

об'єкт ближче, ніж села з меншою кількістю POI. Та чи стверджується факт того, що для сіл з декількома POI в радіусі, з переходом від першого до останнього об'єкту будуть віддалятися. Для цього було використано коробкові діаграми. Одну з таких можна побачити на Рисунку 3.6. І як легко можна побачити, дані правила виконуються.

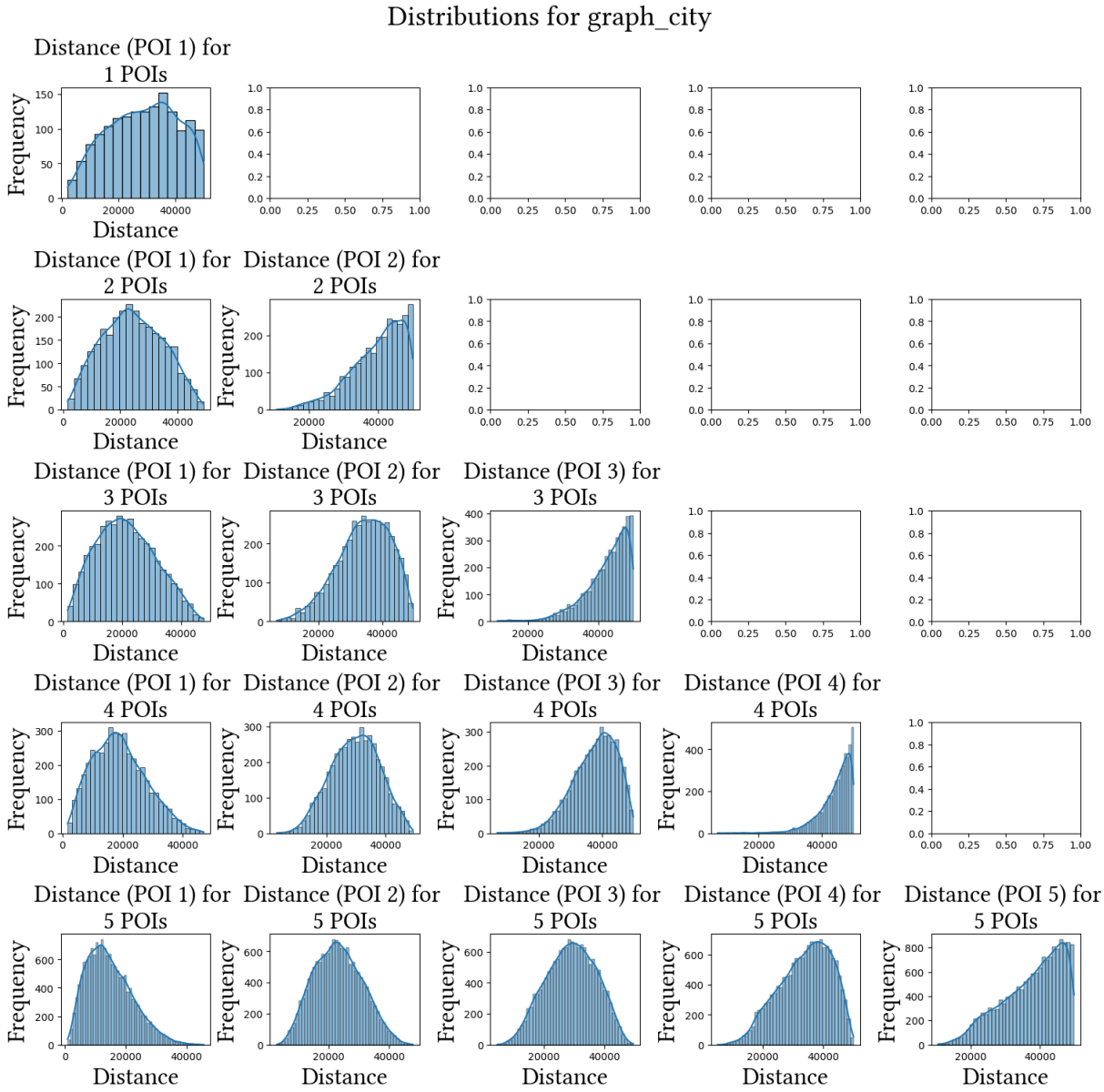

Рисунок 3.6 – Матриця розподілу об'єктів у «граф міста» на основі кількості POI у радіусі для кожного конкретного порядку точки з цього масиву.

3.6 Обговорення

Комплексний набір даних та введення нових дескрипторів, як детально описано в розділі з результатами, значно покращують розуміння сільських районів. Включення близькості до POI таких як: дороги, міста та соціальні об'єкти, графові структури дескрипторів з їх детальними геопросторовими координатами та метриками відстаней — забезпечують глибоке дослідження доступності та розподілу основної інфраструктури.

Цей статистичний аналіз виявляє кілька ключових тенденцій. Право зміщений розподіл RD_m1_NEAR свідчить про те, що хоча багато сіл підключені до основних доріг, значна кількість не підключена, що підкреслює потенційні області для розвитку інфраструктури. На гістограмі, представленій на Рисунку 3.2, можемо чітко побачити, що більшість сіл мають прямий доступ до вторинних та сільських доріг. Цей доступ усуває необхідність виділення ресурсів на такі типи доріг і полегшує доступ, що сприяє розвитку інфраструктури та бізнесу. Аналогічно, логнормальні розподіли для відстаней до різних POI виявляють диспропорції в доступі до важливих послуг, таких як освіта, охорона здоров'я та роздрібна торгівля, вказуючи на необхідність цілеспрямованих покращень. Зручності, такі як освіта, церкви, лікарні та дитячі садки, а також менш важливі, але все ж важливі об'єкти для комфорту людей, такі як місцеві парки, магазини, готелі та національні парки, демонструють високі перші квартилі. Це свідчить про те, що багато сіл мають добре розвинену інфраструктуру для цих типів POI. Трикутний розподіл для Kyiv_NEAR вказує на відносно рівномірний доступ до столиці, що може сприяти економічним можливостям та доступу до послуг.

Коробкова діаграма на Рисунку 3.3 підтверджує наведені вище описи. Вона також надає додаткові інсайти, підтверджуючи, що відстань до всіх типів зручностей зазвичай менше 50 кілометрів, за винятком

регіональних та національних парків або відстані до столиці. Це відкриття є важливим з точки зору здатності суспільства швидко та легко отримувати доступ до соціальних послуг. Аналіз підкріплює ці інсайти, надаючи візуальні докази відстаней до різних об'єктів та визначаючи області, де диспропорції можуть бути найбільш ефективно усунені. Кореляційна матриця додатково підкреслює логічний зв'язок між близькістю села до міст і наявністю зручностей, підтверджуючи, що міські центри зазвичай пропонують більш комплексну інфраструктуру.

Вирівнювання нових графових дескрипторів з оригінальним набором даних вказує на те, що наш новаторський підхід не тільки зберігає, але й збагачує описову якість даних без зміни їх фундаментальних характеристик розподілу. Ця послідовність є життєво важливою для забезпечення того, щоб будь-які рекомендації щодо політики на основі цих дескрипторів точно відображали реальні умови. Крім того, це дослідження дало змістовні інсайти, такі як: кореляція між наявністю магазинів, медичних закладів, дитячих садків та готелів. Враховуючи, що сектори магазинів та готелів в Україні переважно приватні, ці інсайти надають чіткий напрямок для покращення загальної привабливості села. Покращення у сфері державних послуг, таких як: медицина, дитячі садки та церкви — ймовірно, призведуть до зростання інтересу до регіону, а отже, і до більшої приватної інвестиційної активності. На додаток до нових графових дескрипторів, було створено Рисунок 3.7, що відображає кількість усіх типів POI з градієнтною шкалою, що дає попередні вказівки щодо найбільш недорозвинених районів України, які потребують покращень. Ця карта закладає основу для подальших детальних аналізів у майбутніх дослідженнях.

Ця методологія підтримує мету Міністерства освіти і науки України використовувати технології для розвитку сільських територій шляхом виявлення просторових нерівностей та потреб у інфраструктурі. Тим не менш, рекомендації дослідження повинні також враховувати соціально-економічні та політичні виклики, пов'язані з реалізацією

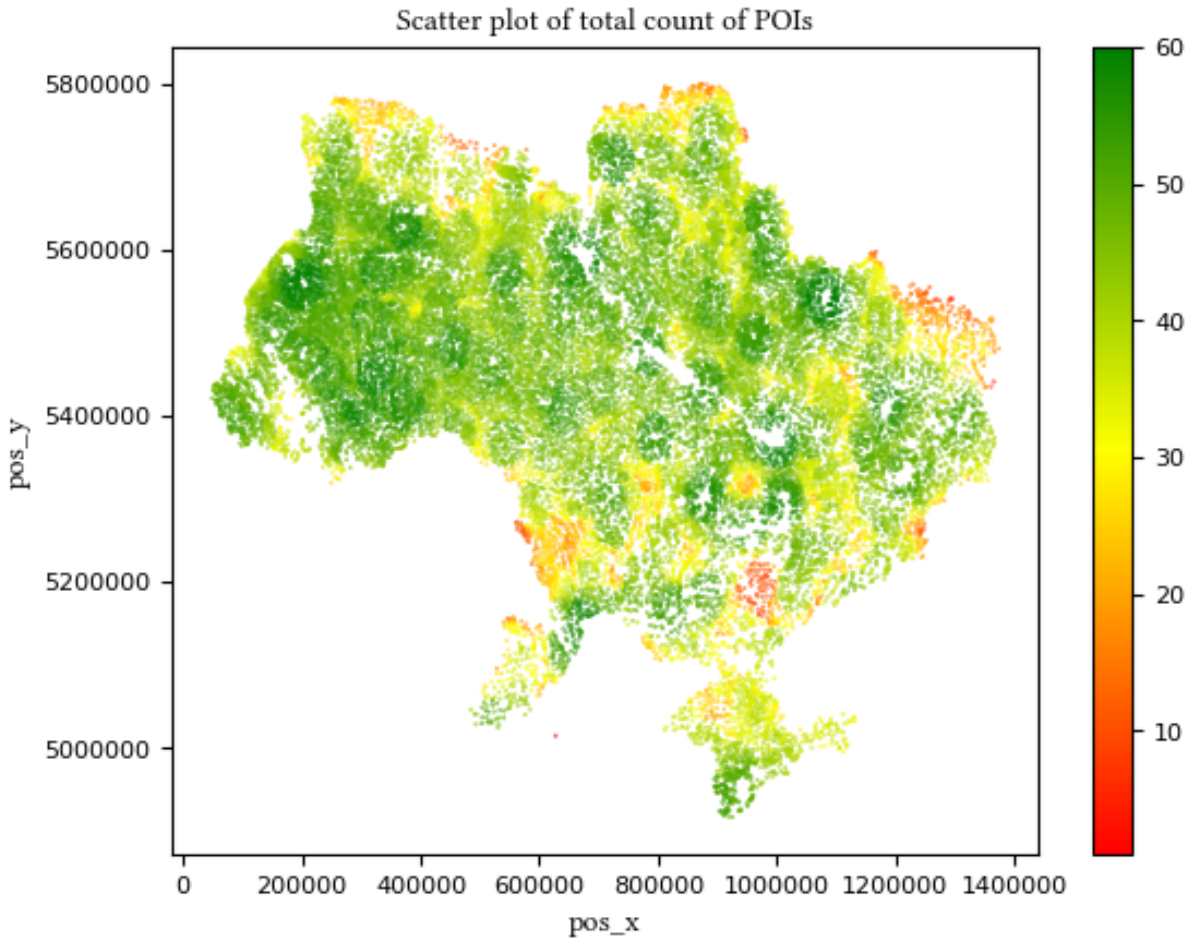

Рисунок 3.7 – Загальна кількість усіх типів точок інтересу (POI) для кожного села в новоствореній графовій структурі, візуалізована за допомогою градієнта.

покращень інфраструктури. Ці виклики включають мобілізацію необхідних ресурсів та залучення політичної підтримки. Крім того, хоча використання даних OpenStreetMap є вигідним, це викликає занепокоєння щодо повноти та точності даних, як обговорюють Herfort та ін. [4]. Ці потенційні розбіжності в якості даних, особливо в сільських умовах, можуть вплинути на висновки дослідження і повинні бути визнані як обмеження.

Висновки до розділу 3

В даному розділі було проведено експеримент, розглянуто результати та обговорено їх. Спираючись на закладені в першому і другому розділах теоретичні відомості та нові методи дослідження була змога виявити ключові аспекти доступності між об'єктами інфраструктури, а також наведено перші підсумки, як точні відносно кожної категорії інфраструктури, так і більш широкі, наводячи перші оцінки доступності та різноманітності POI в рамках усієї України.

ВИСНОВКИ

Це дослідження підкреслило цінність заснованого на даних геопросторового аналізу в просуванні розвитку сільських територій, особливо в унікально складному поточному контексті, з яким стикається Україна. Основними результатами нашого дослідження є:

1) **Розробка розширених дескрипторів доступності**: Використовуючи геопросторові дані, зібрані з OpenStreetMap (OSM) та Humanitarian Data Exchange (HDX), ми створили нові дескриптори, які включають близькість до доріг, міст, соціальних об'єктів та інших точок інтересу (POI). Ці дескриптори надають детальну інформацію про доступність та розподіл основної інфраструктури в сільських районах.

2) **Кількісний та графічний аналіз**: Статистичний аналіз показав, що нові дескриптори зберігають основні характеристики попередніх метрик доступності, забезпечуючи цілісність та правдивість даних. Гістограми, коробкові діаграми та кореляційні матриці допомогли ідентифікувати нормально розподілені особливості та викиди. Наприклад, розподіл відстаней до медичних клінік показав значний правий зсув, що вказує на те, що багато сіл розташовані далеко від цих об'єктів.

3) **Інфраструктурні карти зв'язків**: Створені детальні карти зв'язків інфраструктури для кожного села дозволяють визначити пріоритетні області для інфраструктурних інвестицій та планування розвитку. Ці карти відображають реальні умови доступності та допомагають виявити просторові нерівності.

4) **Ефективність нових методів обробки даних**: Запропоновані методи обробки даних, такі як сегментація на буферні коробки та використання алгоритмів пошуку найближчих POI, дозволили значно скоротити час обчислень та забезпечити точність результатів. Це дало змогу обробляти великі обсяги даних з високою ефективністю.

5) **Статистичний аналіз нових дескрипторів**: Аналіз нових

дескрипторів, таких як відстані до найближчих POI, показав, що вони зберігають узгодженість з початковими даними та надають додаткові інсайти щодо доступності інфраструктури. Наприклад, коробкові діаграми підтвердили, що розподіл відстаней до інфраструктури зберігає свою форму після перетворення до графової моделі.

Подальші дослідження можуть бути спрямовані на:

1) **Розширення набору даних**: Включення додаткових соціально-економічних даних для більш комплексного аналізу потреб сільських громад. Це може включати дані про доходи, рівень зайнятості та інші важливі показники.

2) **Інтеграція реальних даних**: Використання реальних даних у режимі реального часу, таких як сенсорні мережі Інтернету речей, для покращення точності та актуальності аналізу. Наприклад, дані про рух транспорту можуть допомогти краще оцінити доступність доріг та транспортних послуг.

3) **Вивчення впливу політичних та економічних факторів**: Дослідження впливу політичних та економічних змін на розвиток сільської інфраструктури та пошук шляхів для подолання можливих перешкод у реалізації інфраструктурних проєктів. Це може включати аналіз впливу фінансування, політичної підтримки та регуляторних змін на розвиток інфраструктури.

ПЕРЕЛІК ПОСИЛАНЬ

[1] OpenStreetMap contributors. *OpenStreetMap*. Англ. 2021. URL: https://download.geofabrik.de/europe/ukraine.html.

[2] Elevatorist.com. *Elevators in Ukraine*. Англ. 2022. URL: https://elevatorist.com/karta-elevatorov-ukrainy.

[3] The Humanitarian Data Exchange. *Settlement Locations in Ukraine*. Англ. 2021. URL: https://data.humdata.org/.

[4] B. Herfort та ін. «A spatio-temporal analysis investigating completeness and inequalities of global urban building data in OpenStreetMap». Англ. В: *Nature Communications* 14 (2023), с. 3985. URL: https://doi.org/10.1038/s41467-023-39698-6.

[5] Nataliia Kussul, Vlada Svirsh та Bohdan Potuzhnyi. «Integrated Geospatial Analysis for Rural Development Metrics». В: *COLINS-2024: 8th International Conference on Computational Linguistics and Intelligent Systems*. Institute of Physics та ін. Lviv, Ukraine, 2024. URL: http://ceur-ws.org/Vol-3664/paper11.pdf.

[6] Y. Liu та ін. «Geospatial characterization of rural settlements and potential targets for revitalization by geoinformation technology». Англ. В: *Scientific Reports* 12 (2022), с. 8399. URL: https://doi.org/10.1038/s41598-022-12294-2.

[7] A. Maryada та V. L. Thatiparthi. «Geospatial technology for mapping and analysis of social and infrastructural facilities at village level: a case study of Chinnapendyala village». Англ. В: *Modeling Earth Systems and Environment* 6 (2020), с. 1763—1781. URL: https://doi.org/10.1007/s40808-020-00788-9.

[8] Mobua.net. *Інформаційний портал мобільного зв'язку України*. 2023. URL: https://mobua.net/.

[9] B. V. Potuzhnyi, V. R. Svirsh та N. M. Kussul. «Enhancing Rural Infrastructure in Ukraine: An Integrated Geospatial Approach». В: *XXII Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених: Теоретичні і прикладні проблеми фізики, математики та інформатики*. Institute of Physics, Technology; Engineering та Informational Technology Department, Bern University of Applied Science; Space Research Institute NASU-SSAU. Kyiv, Ukraine, 2024, с. 353—359. URL: https://drive.google.com/file/d/1gCPaXLRwF2L_qQkwE3Nf9-GmNVBae6zs/view.

[10] Vlada Svirsh, Bohdan Potuzhnyi та Nataliia Kussul. «Geospatial Framework for Enhancing Rural Infrastructure Development». В: *Book of Abstracts: IV International Symposium on Applied Geoinformatics*. Wroclaw University of Science та Technology, Yildiz Technical University, Technical University of Crete. 2024, с. 30. URL: https://www.kongresistemi.com/panel/UserUploads/Files/95ef88acb8bd857.pdf.

[11] H. Yailymova та ін. «Geospatial Analysis of Life Quality in Ukrainian Rural Areas». Англ. В: *Proceedings of the 13th International Conference on Dependable Systems, Services and Technologies (DESSERT)* (2023), с. 1—5. URL: https://doi.org/10.1109/DESSERT61349.2023.10416517.

[12] Нова Пошта. *Національний оператор експрес-доставки України*. 2023. URL: https://novaposhta.ua/.

[13] Укрпошта. *Національний оператор поштового зв'язку України*. 2023. URL: https://www.ukrposhta.ua/.

ДОДАТОК А ТЕКСТИ ПРОГРАМ

А.1 Програма 1 (Створення нових дескрипторів)

```
#!pip install fastkml
#%%
#!pip install geopandas
#%%
# Standard imports
import os
import json
import random
import zipfile

# Data processing and scientific computing
import numpy as np
import pandas as pd
import geopandas as gpd
#import cudf
#import cuspatial

# Concurrency
import concurrent.futures
from concurrent.futures import ProcessPoolExecutor, as_completed

# Visualization
import matplotlib.pyplot as plt
import matplotlib.font_manager as font_manager
import folium
from fastkml import kml

# Utility
from tqdm import tqdm

# Configuration for pandas display
pd.set_option('display.max_columns', None)
#%% md
### Part 1: Definition of function for processing
#%%
# Some basic functions for the processing
def convert_gdf_to_df(gdf):
    df = pd.DataFrame(gdf)
    df['pos_x'] = df['pos_x'] + 350000
    return df

def calculate_distance(x1, y1, x2, y2):
    """Calculate Euclidean distance."""
    return np.sqrt((x1 - x2) ** 2 + (y1 - y2) ** 2)
#%%
# Block with functions to process the creation of the buffers
def assign_to_block(x, y, block_size):
    """Assign coordinates to a block identifier."""
    return int(x // block_size), int(y // block_size)

def partition_data(df, block_size):
    """Partition data into blocks."""
    blocks = {}
    for index, row in df.iterrows():
        block_id = assign_to_block(row['pos_x'], row['pos_y'], block_size)
        if block_id not in blocks:
            blocks[block_id] = []
        blocks[block_id].append(row)
    return {block: pd.DataFrame(data) for block, data in blocks.items()}

def get_neighboring_blocks(block_id):
    """Get identifiers for a block and its immediate neighbors."""
    x, y = block_id
    return [(x + dx, y + dy) for dx in [-1, 0, 1] for dy in [-1, 0, 1]]

#%%
# Functions which are doing the processing of villages
```

```
def count_nearby_points_optimized(village_df, poi_df, range_in_km, name_for_new_column, identifier,
↪  columns_to_include):
    range_in_meters = range_in_km * 1000
    block_size = range_in_meters

    # Partition POIs into blocks
    poi_blocks = partition_data(poi_df, block_size)

    # Prepare arguments for each village for parallel processing
    args_list = [(village, poi_blocks, block_size, range_in_meters, identifier, columns_to_include)
                 for _, village in village_df.iterrows()]

    # Use ProcessPoolExecutor to process each village in parallel
    with concurrent.futures.ProcessPoolExecutor() as executor:
        # Wrap executor.map with tqdm for progress tracking
        results = list(tqdm(executor.map(process_village, args_list), total=len(args_list)))

    village_df[name_for_new_column] = results

    return village_df

def process_village(args):
    village, poi_blocks, block_size, range_in_meters, identifier, columns_to_include = args
    return find_nearby_pois_for_village(village, poi_blocks, block_size, range_in_meters, identifier,
    ↪  columns_to_include)

def find_nearby_pois_for_village(village, blocks, block_size, range_in_meters, identifier,
↪  columns_to_include):
    """Find nearby POIs for a single village."""
    block_id = assign_to_block(village['pos_x'], village['pos_y'], block_size)
    nearby_pois = []
    for neighbor_block in get_neighboring_blocks(block_id):
        if neighbor_block in blocks:
            for _, poi in blocks[neighbor_block].iterrows():
                distance = calculate_distance(village['pos_x'], village['pos_y'], poi['pos_x'], poi['pos_y'])
                if distance <= range_in_meters:
                    poi_data = {"id_type": identifier, "id": poi[identifier], "distance": distance}
                    for column in columns_to_include:
                        if column not in [identifier]:
                            poi_data[column] = poi[column]
                    nearby_pois.append(poi_data)

    nearby_pois.sort(key=lambda x: x['distance'])
    if len(nearby_pois) > 5:
        nearby_pois = nearby_pois[:5]

    return json.dumps(nearby_pois)
#%%
# Main function to load JSON with data definition and process it
def load_and_process_json(data):
    main_gdf = gpd.read_file(data['path'])
    main_gdf['geometry'] = main_gdf['geometry'].centroid
    main_gdf['pos_x'] = main_gdf.geometry.x
    main_gdf['pos_y'] = main_gdf.geometry.y
    main_df = convert_gdf_to_df(main_gdf[data['columns_to_include']])
    main_df = main_df.sort_values(['pos_x'])

    for desc in data['description']:
        poi_gdf = gpd.read_file(desc['file'])
        poi_gdf['geometry'] = poi_gdf['geometry'].centroid
        poi_gdf['pos_x'] = poi_gdf.geometry.x
        poi_gdf['pos_y'] = poi_gdf.geometry.y
        poi_df = convert_gdf_to_df(poi_gdf[desc['columns_to_include']])
        poi_df_sorted = poi_df.sort_values(['pos_x', 'pos_y'])

        print(poi_df_sorted)

        main_df = count_nearby_points_optimized(
            village_df=main_df,
            poi_df=poi_df_sorted,
            range_in_km=desc['distance'],
            name_for_new_column=data['prefix_for_new_columns'] + desc["type"],
            identifier=desc['id'],
            columns_to_include=desc['columns_to_include']
        )
```

```
    return main_df
#%%
# This is the set-up that was used so far
data = {
  "main" : "villages",
  "path" : "/kaggle/input/diploma-dataset/diploma-dataset-v2/dataset-diploma/Data_for_distances"
           "/VILLAGE_Dist/VILLAGE_ok_Dist2.shp",
  "prefix_for_new_columns" : "graph_",
  "columns_to_include" : ["admin4Pcod", "RD_m1_NEAR", "RD_m2_NEAR", "RD_m3_NEAR",
    "CITY2_NEAR", "elevators_", "Kiyv_NEAR_", "Lis_NEAR_D", "LokPark_NE", "NatPark_NE",
    "regPark_NE", "TOWN_NEAR_", "Water_NEAR", "hz", "kinder_NEA",
    "bank_NEAR_", "cerkva_NEA", "education_", "hotels_NEA", "library_NE",
    "likarni_NE", "magaz_NEAR", "occup_NEAR", "power_NEAR", "swimpool_N",
    "hz2", "pos_x", "pos_y"
  ],
  "description" : [
    {
      "type" : "city",
      "distance" : 50,
      "max_quantity_in_distance" : 5,
      "file":
      ↪  "/kaggle/input/diploma-dataset/diploma-dataset-v2/dataset-diploma/Data_for_distances/CITY/CITY.shp",
      "id": "admin4Pcod",
      "columns_to_include": ["admin4Pcod", "pos_x", "pos_y"]
    },
    {
      "type" : "local_park",
      "distance" : 30,
      "max_quantity_in_distance" : 5,
      "file": "/kaggle/input/diploma-dataset/diploma-dataset-v2/dataset-diploma/Data_for_distances"
              "/PARKS/local_parks/local_parks.shp",
      "id": "KodPZF",
      "columns_to_include": ["KodPZF", "KatObPZF", "AreaPZF", "pos_x", "pos_y"]
    },
    {
      "type" : "national_park",
      "distance" : 50,
      "max_quantity_in_distance" : 5,
      "file": "/kaggle/input/diploma-dataset/diploma-dataset-v2/dataset-diploma/Data_for_distances"
              "/PARKS/national_parks/national_parks.shp",
      "id": "KodPZF",
      "columns_to_include": ["KodPZF", "AreaPZF", "pos_x", "pos_y"]
    },
    {
      "type" : "regional_park",
      "distance" : 50,
      "max_quantity_in_distance" : 5,
      "file": "/kaggle/input/diploma-dataset/diploma-dataset-v2/dataset-diploma/Data_for_distances"
              "/PARKS/regional_parks/regional_parks.shp",
      "id": "KodPZF",
      "columns_to_include": ["KodPZF", "AreaPZF", "pos_x", "pos_y"]
    },
    {
      "type" : "bank",
      "distance" : 30,
      "max_quantity_in_distance" : 5,
      "file": "/kaggle/input/diploma-dataset/diploma-dataset-v2/dataset-diploma/Data_for_distances"
              "/bank/bank.shp",
      "id": "osm_id",
      "columns_to_include": ["osm_id", "pos_x", "pos_y"]
    },
    {
      "type" : "church",
      "distance" : 30,
      "max_quantity_in_distance" : 5,
      "file": "/kaggle/input/diploma-dataset/diploma-dataset-v2/dataset-diploma/Data_for_distances"
              "/church/church.shp",
      "id": "osm_id",
      "columns_to_include": ["osm_id", "fclass", "pos_x", "pos_y"]
    },
    {
      "type" : "edu",
      "distance" : 30,
      "max_quantity_in_distance" : 5,
```

```
      "file": "/kaggle/input/diploma-dataset/diploma-dataset-v2/dataset-diploma/Data_for_distances"
              "/education/education.shp",
      "id": "osm_id",
      "columns_to_include": ["osm_id", "fclass", "pos_x", "pos_y"]
    },
    {
      "type" : "elevator",
      "distance" : 30,
      "max_quantity_in_distance" : 5,
      "file": "/kaggle/input/diploma-dataset/diploma-dataset-v2/dataset-diploma/Data_for_distances"
              "/elevators/elevators.shp",
      "id": "id",
      "columns_to_include": ["id", "pos_x", "pos_y"]
    },
    {
      "type" : "hotel",
      "distance" : 30,
      "max_quantity_in_distance" : 5,
      "file": "/kaggle/input/diploma-dataset/diploma-dataset-v2/dataset-diploma/Data_for_distances"
              "/hotels/hotels.shp",
      "id": "osm_id",
      "columns_to_include": ["osm_id", "fclass", "pos_x", "pos_y"]
    },
    {
      "type" : "kindergarten",
      "distance" : 30,
      "max_quantity_in_distance" : 5,
      "file": "/kaggle/input/diploma-dataset/diploma-dataset-v2/dataset-diploma/Data_for_distances"
              "/kindergarten/kindergarten.shp",
      "id": "osm_id",
      "columns_to_include": ["osm_id", "pos_x", "pos_y"]
    },
    {
      "type" : "library",
      "distance" : 30,
      "max_quantity_in_distance" : 5,
      "file": "/kaggle/input/diploma-dataset/diploma-dataset-v2/dataset-diploma/Data_for_distances"
              "/library/library.shp",
      "id": "osm_id",
      "columns_to_include": ["osm_id", "pos_x", "pos_y"]
    },
    {
      "type" : "medicine",
      "distance" : 30,
      "max_quantity_in_distance" : 5,
      "file": "/kaggle/input/diploma-dataset/diploma-dataset-v2/dataset-diploma/Data_for_distances"
              "/medicine/likarni.shp",
      "id": "osm_id",
      "columns_to_include": ["osm_id", "fclass", "pos_x", "pos_y"]
    },
    {
      "type" : "shop",
      "distance" : 30,
      "max_quantity_in_distance" : 5,
      "file": "/kaggle/input/diploma-dataset/diploma-dataset-v2/dataset-diploma/Data_for_distances"
              "/shops/shops.shp",
      "id": "osm_id",
      "columns_to_include": ["osm_id", "fclass", "pos_x", "pos_y"]
    }
  ]
}
#%% md
### Part 2: Data loading and visualisation major points
#%%
# Block to show POIs on the subplots

def load_and_convert_to_df(file_path, columns_to_include):
    gdf = gpd.read_file(file_path)
    gdf['geometry'] = gdf['geometry'].centroid
    gdf['pos_x'] = gdf.geometry.x
    gdf['pos_y'] = gdf.geometry.y
    df = gdf[columns_to_include]
    return df

fig, axs = plt.subplots(4, 4, figsize=(15, 15))
```

```
axs = axs.flatten()

for index, desc in enumerate(data['description']):
    file_path = desc['file']
    columns_to_include = desc['columns_to_include']

    df = load_and_convert_to_df(file_path, columns_to_include)

    axs[index].scatter(df['pos_x'], df['pos_y'], label=desc['type'], alpha=0.6, s=10)
    axs[index].set_title(desc['type'],  fontsize=24)
    axs[index].set_xlabel('pos_x', fontsize=20)
    axs[index].set_ylabel('pos_y', fontsize=20)
    axs[index].tick_params(axis='both', which='major', labelsize=16)
    axs[index].legend( fontsize=14)

# Hide any unused subplots
for i in range(len(data['description']), len(axs)):
    axs[i].axis('off')

plt.tight_layout()
plt.savefig("maps.png", bbox_inches='tight')
plt.show()
#%%
# File loading and processing
gdf = gpd.read_file("/kaggle/input/diploma-dataset/diploma-dataset-v2/dataset-diploma/Data_for_distances"
                    "/VILLAGE_Dist/VILLAGE_ok_Dist2.shp")
print(gdf.describe())
print(gdf.crs)
#%%
# Block to show all villages on hte map of Ukraine

ukraine_shapefile_path = '/kaggle/input/diploma-dataset/ua_shp/ua.shp'
ukraine_boundary = gpd.read_file(ukraine_shapefile_path)

fig, ax = plt.subplots(figsize=(10, 10))
gdf = gdf.to_crs(ukraine_boundary.crs)
ukraine_boundary.plot(ax=ax, color='lightgray')
gdf.plot(ax=ax, marker='o', color='red', markersize=5, alpha=0.6)
bounds = ukraine_boundary.total_bounds
ax.set_xlim(bounds[0], bounds[2])
ax.set_ylim(bounds[1], bounds[3])
ax_crs = ax.figure.canvas.get_renderer()
ukraine_boundary = ukraine_boundary.to_crs(epsg=3857)
gdf = gdf.to_crs(epsg=3857)
ctx.add_basemap(ax, source=ctx.providers.Stamen.TonerLite)
ax.set_axis_off()

plt.show()
#%% md
### Part 4: Processing of the dataset
#%%
# This block will create the graph-based deescription
dataset_df = load_and_process_json(data)
#%%
dataset_df.to_csv("dataset_graphed_final_edition_blocks_search.csv", index=False)
dataset_df
#%% md
#### Mobile network coverage
#%%
gdf = gpd.read_file("/kaggle/input/diploma-dataset/diploma-dataset-v2/dataset-diploma/Data_for_distances"
                    "/VILLAGE_Dist/VILLAGE_ok_Dist2.shp")
gdf.geometry = gdf.geometry.centroid
#%%
# Mapping the mobile network coverages on the map of Ukraine
def kmz_to_kml(kmz_filepath, output_folder):
    with zipfile.ZipFile(kmz_filepath, 'r') as zip_ref:
        zip_ref.extractall(output_folder)
        for filename in zip_ref.namelist():
            if filename.endswith('.kml'):
                return os.path.join(output_folder, filename)
    return None

def kml_to_gdf(kml_filepath):
    """Converts a KML file to a GeoDataFrame."""
    with open(kml_filepath, 'rt', encoding="utf-8") as myfile:
```

```
        doc = myfile.read().encode('utf-8')

    k = kml.KML()
    k.from_string(doc)

    features = list(k.features())
    geoms = []
    names = []
    descriptions = []

    for feature in features:
        for placemark in feature.features():
            geometry = placemark.geometry
            geoms.append(geometry)
            names.append(placemark.name)
            descriptions.append(placemark.description)

    gdf = gpd.GeoDataFrame({
        'name': names,
        'description': descriptions,
        'geometry': geoms
    })

    return gdf

def plot_kml_on_map(fmap, kml_filepath, color):
    with open(kml_filepath, 'rt', encoding="utf-8") as myfile:
        doc = myfile.read().encode('utf-8')

    k = kml.KML()
    k.from_string(doc)

    features = list(k.features())
    for feature in features:
        for placemark in feature.features():
            geometry = placemark.geometry
            if geometry.geom_type == 'Point':
                folium.Marker([geometry.y, geometry.x], icon=folium.Icon(color=color)).add_to(fmap)
            elif geometry.geom_type in ['LineString', 'LinearRing']:
                folium.PolyLine([(point[1], point[0]) for point in geometry.coords], color=color).add_to(fmap)
            elif geometry.geom_type == 'Polygon':
                exterior_coords = [(point[1], point[0]) for point in geometry.exterior.coords]
                folium.Polygon(locations=exterior_coords, color=color).add_to(fmap)
            elif geometry.geom_type == 'MultiPolygon':
                for polygon in geometry.geoms:
                    exterior_coords = [(point[1], point[0]) for point in polygon.exterior.coords]
                    folium.Polygon(locations=exterior_coords, color=color).add_to(fmap)

def find_kmz_files(directory):
    """Return a list of KMZ file paths in the given directory."""
    kmz_files = []
    for root, dirs, files in os.walk(directory):
        for file in files:
            if file.endswith('.kmz'):
                kmz_files.append(os.path.join(root, file))
    return kmz_files

def generate_random_color():
    """Generate a random hex color."""
    return "#{:06x}".format(random.randint(0, 0xFFFFFF))

kmz_directory = '/kaggle/input/diploma-dataset/diploma-dataset-v2/dataset-diploma/mobile_networks_from
↪  mobua.net'
temp_folder = '/kaggle/working/'

fmap = folium.Map(location=[20, 0], zoom_start=2)

kmz_files = find_kmz_files(kmz_directory)
legend_html = '<div style="position: fixed; bottom: 50px; left: 50px; background-color: white; border:2px
↪  solid grey; z-index:9999; font-size:14px; padding: 10px;"><b>Legend</b><br>'

for kmz_file in kmz_files:
    color = generate_random_color()
    kml_file = kmz_to_kml(kmz_file, temp_folder)
    if kml_file:
```

```
        plot_kml_on_map(fmap, kml_file, color)
        legend_html += f'  {os.path.basename(kmz_file)}   <i class="fa fa-map-marker fa-2x"
        ↪  style="color:{color}"></i><br>'

legend_html += '</div>'
fmap.get_root().html.add_child(folium.Element(legend_html))

fmap.save('map.html')
#%%
gdf = gdf.to_crs(epsg=4258)

def extract_lat_lon(geom):
    if geom.type == 'Point':
        return geom.y, geom.x
    else:
        centroid = geom.centroid
        return centroid.y, centroid.x

gdf['latitude'], gdf['longitude'] = zip(*gdf['geometry'].apply(extract_lat_lon))
#%%
# Another varient of the map
m = folium.Map(location=[gdf['latitude'].mean(), gdf['longitude'].mean()], zoom_start=6)

# Function to add polygons from a GeoDataFrame to a folium map
def add_polygons_to_map(fmap, polygons_gdf, layer_name="Polygons", color="blue"):
    for _, row in polygons_gdf.iterrows():
        sim_geo = gpd.GeoSeries(row['geometry']).simplify(tolerance=0.001)
        geo_j = sim_geo.to_json()
        geo_j = folium.GeoJson(data=geo_j,
                               style_function=lambda x: {'fillColor': color, 'color': color})
        geo_j.add_to(fmap)
        folium.LayerControl().add_to(fmap)

# Plot each village as a point on the map
for index, row in gdf.iterrows():
    folium.Circle(
        location=[row['latitude'], row['longitude']],
        radius=100,
        color='blue',
        fill=True,
        fill_color='blue',
        fill_opacity=0.5,
        tooltip=row.get('name', 'Unnamed')
    ).add_to(m)

# Loop through each KMZ file, convert to GeoDataFrame, and add polygons to the map
for kmz_file in kmz_files:
    column_name = os.path.splitext(kmz_file)[0]
    kmz_file_path = os.path.join(kmz_directory, kmz_file)
    kml_file = kmz_to_kml(kmz_file_path, temp_folder)

    if kml_file:
        polygons_gdf = kml_to_gdf(kml_file).set_crs(epsg=4326)  # Convert to EPSG:4326
        add_polygons_to_map(m, polygons_gdf, layer_name=column_name, color="blue")

# Save or display the map
m.save('villages_and_kmz_polygons_map.html')
#%%
# Block in which we are checking the villages for their connection to mobile network providers
results_df = gdf.copy()
for kmz_file in os.listdir(kmz_directory):
    if kmz_file.endswith('.kmz'):
        column_name = os.path.splitext(kmz_file)[0]
        results_df[column_name] = False  # Initialize with False

        kmz_file_path = os.path.join(kmz_directory, kmz_file)
        kml_file = kmz_to_kml(kmz_file_path, temp_folder)

        if kml_file:
            polygons_gdf = kml_to_gdf(kml_file).set_crs("EPSG:4258")

            # Perform spatial join: which villages are within any polygon from this KMZ file?
            joined = gpd.sjoin(gdf, polygons_gdf, how="inner", op='within')

            results_df.loc[joined.index, column_name] = True
```

```
#%%
columns_of_interest = ['lifecell-4g', 'trimob', 'vodafone-3g', 'kyivstar-4g', 'kyivstar-3g', 'vodafone-4g',
↪  'lifecell-3g']
df = pd.read_csv("/kaggle/input/diploma-dataset/dataset_graphed_final_edition_blocks_search.csv")
merged_df = df.join(results_df[columns_of_interest])
merged_df.to_csv("/kaggle/working/dataset_graphed_final_edition_blocks_with_mob_coverage.csv", index=False)
#%% md
#### Processing of the postal providers
#%%
tqdm.pandas()

firm1_df = pd.read_csv("/kaggle/input/diploma-dataset/diploma-dataset-v2/dataset-diploma"
                       "/post-departments/nova-poshta/departments_coordinates_nova_poshta.csv")
firm1_gdf = gpd.GeoDataFrame(firm1_df, geometry=gpd.points_from_xy(firm1_df.Longitude, firm1_df.Latitude))
firm1_gdf.crs = "EPSG:4258"

firm2_df = pd.read_csv("/kaggle/input/diploma-dataset/diploma-dataset-v2/dataset-diploma"
                       "/post-departments/ukr-poshta/ukr-post-dep-loc.csv")
firm2_gdf = gpd.GeoDataFrame(firm2_df, geometry=gpd.points_from_xy(firm2_df.Longitude, firm2_df.Latitude))
firm2_gdf.crs = "EPSG:4258"

firm1_gdf = firm1_gdf.to_crs("EPSG:32636")
firm1_gdf.describe()
firm2_gdf = firm2_gdf.to_crs("EPSG:32636")
firm2_gdf.describe()
gdf = gdf.to_crs("EPSG:32636")
#%%
firm1_gdf.geometry.x
#%%
fig, axs = plt.subplots(4, 4, figsize=(15, 15))
axs = axs.flatten()

all_pos_x = []
all_pos_y = []

for index, desc in enumerate(data['description']):
    file_path = desc['file']
    columns_to_include = desc['columns_to_include']

    df = load_and_convert_to_df(file_path, columns_to_include)
    all_pos_x.append(df['pos_x'].min())
    all_pos_x.append(df['pos_x'].max())
    all_pos_y.append(df['pos_y'].min())
    all_pos_y.append(df['pos_y'].max())

    axs[index].scatter(df['pos_x'], df['pos_y'], label=desc['type'], alpha=0.6, s=10)
    axs[index].set_title(desc['type'],  fontsize=24)
    axs[index].set_xlabel('pos_x', fontsize=20)
    axs[index].set_ylabel('pos_y', fontsize=20)
    axs[index].tick_params(axis='both', which='major', labelsize=16)
    axs[index].legend( fontsize=14)


min_pos_x = min(all_pos_x) - 50000
max_pos_x = max(all_pos_x) + 50000
min_pos_y = min(all_pos_y) - 50000
max_pos_y = max(all_pos_y) + 50000

axs[len(data['description'])].scatter(firm1_gdf.geometry.x, firm1_gdf.geometry.y, label='Nova Poshta',
↪  alpha=0.6, s=10)
axs[len(data['description'])].set_title('Nova Poshta', fontsize=24)
axs[len(data['description'])].set_xlim(min_pos_x, max_pos_x)
axs[len(data['description'])].set_ylim(min_pos_y, max_pos_y)
axs[len(data['description'])].set_xlabel('pos_x', fontsize=20)
axs[len(data['description'])].set_ylabel('pos_y', fontsize=20)
axs[len(data['description'])].tick_params(axis='both', which='major', labelsize=16)
axs[len(data['description'])].legend(fontsize=14)

axs[len(data['description'])+1].scatter(firm2_gdf.geometry.x, firm2_gdf.geometry.y, label='Ukr Poshta',
↪  alpha=0.6, s=10)
axs[len(data['description'])+1].set_title('Ukr Poshta', fontsize=24)
axs[len(data['description'])+1].set_xlim(min_pos_x, max_pos_x)
axs[len(data['description'])+1].set_ylim(min_pos_y, max_pos_y)
axs[len(data['description'])+1].set_xlabel('pos_x', fontsize=20)
axs[len(data['description'])+1].set_ylabel('pos_y', fontsize=20)
```

```
axs[len(data['description'])+1].tick_params(axis='both', which='major', labelsize=16)
axs[len(data['description'])+1].legend(fontsize=14)

# Hide any unused subplots
for i in range(len(data['description'])+2, len(axs)):
    axs[i].axis('off')

plt.tight_layout()
plt.savefig("maps.png", bbox_inches='tight')
plt.show()
#%%
print(gdf.geometry.x.describe())
print(gdf.geometry.y.describe())
print(firm1_gdf.geometry.x.describe())
print(firm1_gdf.geometry.y.describe())
#%%
# Calculate minimum distance to the closest post office for each firm
gdf['NP_Min_Dist'] = gdf.geometry.progress_apply(lambda x: firm1_gdf.distance(x).min())
gdf['UP_Min_Dist'] = gdf.geometry.progress_apply(lambda x: firm2_gdf.distance(x).min())
#%%
df_new = pd.DataFrame(gdf)
df_new
#%%
df = pd.read_csv("/kaggle/input/diploma-dataset/dataset_graphed_final_edition_blocks_with_mob_coverage.csv")
columns_to_include = ["NP_Min_Dist", "UP_Min_Dist"]
merged_df = df.join(df_new[columns_to_include])
merged_df.to_csv("/kaggle/working/dataset_graphed_final_edition_blocks_with_mob_coverage_and_post_new.csv",
↪  index=False)
merged_df
```

A.2 Програма 2 (Аналіз даних)

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.colors as mcolors
from matplotlib.colors import LinearSegmentedColormap
import matplotlib.font_manager as font_manager
from matplotlib.ticker import ScalarFormatter
import json
import warnings
import scipy.stats as stats
from tqdm import tqdm

font_path = '/kaggle/input/diploma-dataset/LibertinusSerif-Regular.ttf'
libertinus_font_prop = font_manager.FontProperties(fname=font_path)

print(f"Font name loaded: {libertinus_font_prop.get_name()}")  # It should print the font name

plt.rcParams['font.family'] = libertinus_font_prop.get_name()
pd.set_option('display.max_columns', None)
warnings.simplefilter(action='ignore', category=FutureWarning)
#%%
df = pd.read_csv("/kaggle/input/diploma-dataset/villages_categorized_mob_post.csv")
df_blocks = pd.read_csv("/kaggle/input/diploma-dataset"
                        "/dataset_graphed_final_edition_blocks_with_mob_coverage_and_post.csv")
#%%
df_blocks.iloc[16555]['graph_shop']
#%%
df.describe()
#%%
df.columns.to_list()
#%%
columns_to_plot = [
 'RD_m1_NEAR',
 'RD_m2_NEAR',
```

```
 'RD_m3_NEAR',
 'CITY2_NEAR',
 'elevators_',
 'Kiyv_NEAR_',
 'LokPark_NE',
 'NatPark_NE',
 'regPark_NE',
 'kinder_NEA',
 'bank_NEAR_',
 'cerkva_NEA',
 'education_',
 'hotels_NEA',
 'library_NE',
 'likarni_NE',
 'magaz_NEAR',
 'NP_Min_Dist',
 'UP_Min_Dist'
]

nrows = 4
ncols = 5

fig, axes = plt.subplots(nrows=nrows, ncols=ncols, figsize=(30, 20))
fig.subplots_adjust(hspace=0.8, wspace=0.7)

axes = axes.flatten()

for i, column in enumerate(columns_to_plot):
    sns.histplot(df[column], kde=True, ax=axes[i])
    title = f'Distribution of\n{column}'

    axes[i].set_title(title, fontproperties=libertinus_font_prop, fontsize=38)
    axes[i].set_xlabel(column, fontproperties=libertinus_font_prop, fontsize=33)
    axes[i].set_ylabel('Frequency', fontproperties=libertinus_font_prop, fontsize=33)
    axes[i].tick_params(axis='both', which='major', labelsize=22)


for j in range(i + 1, nrows * ncols):
    fig.delaxes(axes[j])
plt.savefig("Distributions", bbox_inches='tight')
plt.show()
#%%
nrows = 4
ncols = 5

fig, axes = plt.subplots(nrows=nrows, ncols=ncols, figsize=(30, 20))
fig.subplots_adjust(hspace=0.7, wspace=0.5)

axes = axes.flatten()

for i, column in enumerate(columns_to_plot):
    sns.boxplot(y=df[column], ax=axes[i])
    axes[i].set_title(f'Boxplot of\n{column}', fontproperties=libertinus_font_prop, fontsize=38)
    axes[i].set_ylabel(column, fontproperties=libertinus_font_prop, fontsize=28)
    axes[i].tick_params(axis='y', labelsize=22)

for j in range(i + 1, nrows * ncols):
    fig.delaxes(axes[j])
plt.savefig("boxplots", bbox_inches='tight')
plt.show()
#%%
corr_matrix = df[columns_to_plot].corr()

plt.figure(figsize=(12, 10))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.05)

plt.title('Correlation Matrix', fontproperties=libertinus_font_prop, fontsize=20)

tick_label_size = 15
libertinus_font_prop.set_size(tick_label_size)

for label in plt.gca().get_xticklabels():
    label.set_fontproperties(libertinus_font_prop)
for label in plt.gca().get_yticklabels():
    label.set_fontproperties(libertinus_font_prop)
```

```
plt.savefig("corr_matrix",  bbox_inches='tight')
plt.show()
#%%
columns_descriptor = {
  "columns" : [
    {
      "name" : "graph_city",
      "objects" : ["distance"]
    },
    {
      "name" : "graph_local_park",
      "objects" : ["distance"]
    },
    {
      "name" : "graph_regional_park",
      "objects" : ["distance"]
    },
    {
      "name" : "graph_national_park",
      "objects" : ["distance"]
    },
    {
      "name" : "graph_bank",
      "objects" : ["distance"]
    },
    {
      "name" : "graph_church",
      "objects" : ["distance"]
    },
    {
      "name" : "graph_edu",
      "objects" : ["distance"]
    },
    {
      "name" : "graph_elevator",
      "objects" : ["distance"]
    },
    {
      "name" : "graph_hotel",
      "objects" : ["distance"]
    },
    {
      "name" : "graph_kindergarten",
      "objects" : ["distance"]
    },
    {
      "name" : "graph_library",
      "objects" : ["distance"]
    },
    {
      "name" : "graph_medicine",
      "objects" : ["distance"]
    },
    {
      "name" : "graph_shop",
      "objects" : ["distance"]
    }
  ]
}
#%%
def count_objects(json_str):
    try:
        data_list = json.loads(json_str)
        return len(data_list)
    except Exception as e:
        return 0

columns_to_analyze = [col["name"] for col in columns_descriptor["columns"]]

for column in columns_to_analyze:
    actual_column_name = f'graph_{column}' if not column.startswith('graph_') else column
    if actual_column_name in df.columns:
        df[actual_column_name + '_count'] = df[actual_column_name].apply(count_objects)
```

```
count_columns = [col + '_count' for col in columns_to_analyze if f'graph_{col}_count' in df.columns or
↪  f'{col}_count' in df.columns]
descriptive_stats = df[count_columns].describe()
print(descriptive_stats)
#%%
corr_matrix = df[count_columns].corr()
plt.figure(figsize=(12, 10))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', linewidths=.5, fmt=".2f")
plt.title('Correlation Matrix of POI Counts', fontproperties=libertinus_font_prop, fontsize=20)

tick_label_size = 15
libertinus_font_prop.set_size(tick_label_size)

for label in plt.gca().get_xticklabels():
    label.set_fontproperties(libertinus_font_prop)
for label in plt.gca().get_yticklabels():
    label.set_fontproperties(libertinus_font_prop)

plt.savefig("corr_matrix_pois", bbox_inches='tight')
plt.show()
#%%
def plot_sum_of_counts(df, count_columns):
    df['total_count'] = df[count_columns].sum(axis=1)

    cmap = LinearSegmentedColormap.from_list(name='custom', colors=['red', 'yellow', 'green'])

    scatter = plt.scatter(df['pos_x'], df['pos_y'], c=df['total_count'], cmap=cmap, s=0.1)

    cbar = plt.colorbar(scatter)

    cbar.ax.tick_params(labelsize=8)

    y_formatter = ScalarFormatter(useOffset=False)
    y_formatter.set_scientific(False)
    plt.gca().yaxis.set_major_formatter(y_formatter)
    plt.gca().xaxis.set_major_formatter(y_formatter)

    plt.tick_params(labelsize=8)

    plt.xlabel('pos_x', fontproperties=libertinus_font_prop, fontsize=10)
    plt.ylabel('pos_y', fontproperties=libertinus_font_prop, fontsize=10)
    plt.title('Scatter plot of total count of POIs', fontproperties=libertinus_font_prop, fontsize=10)

    plt.savefig("total_count", bbox_inches='tight')
    plt.show()

plot_sum_of_counts(df, count_columns)
#%%
def plot_distance_distributions(df, columns_descriptor):
    for column_info in columns_descriptor['columns']:
        column_name = column_info['name']
        attributes = column_info['objects']

        total_columns = len(attributes) * 5
        fig, axes = plt.subplots(5, total_columns, figsize=(3 * total_columns, 3 * 5),
        ↪  constrained_layout=True)
        fig.suptitle(f'Distributions for {column_name}',fontproperties=libertinus_font_prop, fontsize=28)

        fig.subplots_adjust(hspace=0.6, wspace=0.6)

        for num_pois in range(1, 6):  # From 1 to 5 POIs
            filtered_df = df[df[column_name].apply(lambda x: len(json.loads(x)) == num_pois if pd.notnull(x)
            ↪  else False)]

            for i, attr in enumerate(attributes):
                for poi_index in range(num_pois):
                    distances = [
                        json.loads(entry)[poi_index][attr] for entry in filtered_df[column_name]
                        if entry and len(json.loads(entry)) > poi_index and attr in
                        ↪  json.loads(entry)[poi_index]
                    ]

                    subplot_index = (num_pois - 1) * total_columns + i * num_pois + poi_index
                    ax = axes[subplot_index // total_columns, subplot_index % total_columns]
```

```
                    sns.histplot(distances, kde=True, ax=ax)
                    ax.set_title(f'{attr.capitalize()} (POI {poi_index + 1}) for \n{num_pois} POIs',
                    ↪  fontproperties=libertinus_font_prop, fontsize=24)
                    ax.set_xlabel(attr.capitalize(), fontproperties=libertinus_font_prop, fontsize=24)
                    ax.set_ylabel('Frequency', fontproperties=libertinus_font_prop, fontsize=24)

        plt.savefig(f"{column_name}_distribution_table", bbox_inches='tight')
        plt.show()
#%%
plot_distance_distributions(df, columns_descriptor)
#%%
clustered_columns = ['roads', 'cities', 'loc_parks', 'nat_parks', 'reg_parks', 'parks', 'banks', 'churches',
↪  'edu', 'elevs', 'hotels', 'kindergartens', 'libraries', 'medicine', 'shops', 'mobile', 'post', 'cluster',
↪  'quality']

corr_matrix_clust = df[clustered_columns].corr()

plt.figure(figsize=(12, 10))
sns.heatmap(corr_matrix_clust, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.05)
plt.title('Correlation Matrix')
plt.show()
```