

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І. І. МЕЧНІКОВА

(повне найменування закладу вищої освіти)

Факультет математики, фізики та інформаційних технологій

(повне найменування факультету)

Кафедра інформаційних технологій

(повна назва кафедри)

Кваліфікаційна робота

на здобуття ступеня вищої освіти «Магістр»

«Моделювання та аналіз соціальних мереж з використанням
графових структур»

(тема кваліфікаційної роботи українською мовою)

«**Modeling and analysis of social networks using
graph structures**»

(тема кваліфікаційної роботи англійською мовою)

Виконала: здобувачка заочної форми навчання
спеціальності 122 Комп'ютерні науки

(код, назва спеціальності)

Освітня програма Комп'ютерні науки

(назва)

Байлюк Вероніка Олегівна

(прізвище, ім'я, по-батькові здобувача)

Керівник к.т.н., доцент Фразе-Фразенко О.О.

(науковий ступінь, вчене звання, прізвище, ініціали)

(підпис)

Рецензент к.т.н., доцент, доцент кафедри інформаційних
технологій НУ ОЮА, Щербина Ю.В.

(науковий ступінь, вчене звання, прізвище, ініціали)

Рекомендовано до захисту:

Протокол засідання кафедри

Інформаційних технологій

№ від 2024 р.

Завідувачка кафедри

(підпис)

(прізвище, ім'я)

Захищено на засіданні ЕК №

протокол № від 2024 р.

Оцінка / /

(за національною шкалою/шкалою ECTS/ бали)

Голова ЕК

(підпис)

(прізвище, ім'я)

Одеса 2024

АНОТАЦІЯ

У кваліфікаційній роботі розробляється тема «Моделювання та аналіз соціальних мереж з використанням графових структур».

Мета роботи – створення методології для аналізу соціальних мереж шляхом побудови та дослідження соціальних графів. Робота зосереджена на застосуванні сучасних графових алгоритмів, таких як визначення центральності вузлів, кластеризація спільнот, аналіз впливу та поширення інформації у мережах.

У результаті виконаного дослідження було розроблено програмний засіб, який дозволяє моделювати взаємозв'язки у соціальних мережах, виявляти ключових користувачів та аналізувати структуру мережі. Також розроблено методи для візуалізації графів, що дозволяють ефективно презентувати результати аналізу для подальшого використання у сфері маркетингу, соціології та інформаційної безпеки.

ABSTRACT

The qualification work explores the topic "Modeling and analysis of social networks using graph structures".

The aim of the work is to develop a methodology for analyzing social networks through the construction and exploration of social graphs. The work focuses on applying modern graph algorithms such as node centrality detection, community clustering, and the analysis of influence and information dissemination in networks.

As a result of the research, a software tool was developed to model relationships in social networks, identify key users, and analyze network structures. Additionally, methods for graph visualization were implemented, enabling effective presentation of analysis results for further use in marketing, sociology, and information security.

ЗМІСТ

ТЕРМІНИ, СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	6
ВСТУП	8
1 АНАЛІЗ АКТУАЛЬНОСТІ ВИКОРИСТАННЯ ГРАФОВИХ МОДЕЛЕЙ	10
1.1 Графові моделі як універсальний інструмент аналізу	10
1.2 Використання графів для аналізу соціальних мереж	13
1.3 Мета та завдання дослідження	15
2 ТЕОРЕТИЧНІ АСПЕКТИ ГРАФОВИХ СТРУКТУР ТА ЇХ ЗАСТОСУВАННЯ	19
2.1 Визначення графів та їх класифікація	19
2.2 Основні поняття теорії графів	22
2.3 Метрики графів	24
3 АЛГОРИТМИ ТА ЗАСОБИ АНАЛІЗУ ТА МОДЕЛЮВАННЯ ГРАФІВ СОЦІАЛЬНИХ МЕРЕЖ	26
3.1 Порівняльна характеристика програмних засобів для аналізу соціальних мереж	26
3.2 Зведене порівняння розглянутих програмних засобів	33
3.3 Проектування програмного засобу аналізу графової структури	38
4 ПОБУДОВА ТА ВІЗУАЛІЗАЦІЯ ГРАФА СОЦІАЛЬНОЇ МЕРЕЖІ	46
4.1 Опис соціальної мережі	46
4.2 Проблематика створення соціального графа	48
4.3 Створення соціального графа на базі аудиторій	49

	5
4.4 Удосконалена методика створення соціального графа на базі аудиторій.....	53
4.5 Реалізація алгоритму парсингу.....	55
4.6 Налаштування та оптимізація методів побудови	57
ВИСНОВКИ.....	62
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	63

ТЕРМІНИ, СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

API (Application Programming Interface) – інтерфейс програмування додатків; набір інструментів і протоколів для взаємодії програмного забезпечення із зовнішніми сервісами.

GNN (Graph Neural Network) – графова нейронна мережа; метод машинного навчання для обробки графових структур.

NLP (Natural Language Processing) – обробка природної мови; технології для аналізу текстової інформації.

AI (Artificial Intelligence) – штучний інтелект; галузь інформатики, яка займається створенням інтелектуальних систем.

CSV (Comma-Separated Values) – формат файлів для зберігання табличних даних.

Gephi – програмне забезпечення для візуалізації та аналізу графів.

IGraph – бібліотека для аналізу та візуалізації графів.

Neo4j – графова база даних для моделювання взаємозв'язків.

PCA (Principal Component Analysis) – метод головних компонент; використовується для зменшення розмірності даних.

PageRank – алгоритм для оцінки важливості вузлів у графі.

Degree Centrality – ступенева центральність; метрика для визначення впливу вузлів у графі.

Clustering Coefficient – коефіцієнт кластеризації; показує, наскільки вузли в графі схильні утворювати групи.

Betweenness Centrality – міжстороння центральність; оцінює важливість вузла як посередника в мережі.

Closeness Centrality – центральність за близькістю; метрика, що відображає "відстань" вузла до інших вузлів у графі.

GraphML – формат для зберігання графових даних.

REST API (Representational State Transfer API) – інтерфейс для взаємодії з веб-сервісами, що використовує HTTP-запити.

Соціальний граф – математична модель, яка представляє структуру взаємодій у соціальній мережі у вигляді вузлів (користувачів, сторінок) та ребер (зв'язків між ними).

Кластер – група вузлів у графі, які мають високу щільність зв'язків між собою, що свідчить про спільність інтересів або взаємодій.

Вузол (Node) – базовий елемент графа, який представляє окремий об'єкт, наприклад, користувача, публічну сторінку чи організацію.

Ребро (Edge) – зв'язок між двома вузлами графа, який може позначати взаємодію, спільну аудиторію або будь-яку іншу форму зв'язку.

Калібрування графа – процес налаштування метрик і параметрів графа для забезпечення його відповідності реальним даним.

ВСТУП

У сучасному світі соціальні мережі стали невід'ємною частиною життя мільйонів людей, виступаючи важливим інструментом комунікації, обміну інформацією та взаємодії. Вони створюють складні мережі зв'язків, які відображають соціальну динаміку, інтереси та взаємодії між людьми. Аналіз цих мереж дозволяє отримати цінні інсайти щодо поведінкових патернів, структур впливу, формування спільнот та шляхів поширення інформації.

Дана робота присвячена моделюванню та аналізу соціальних мереж за допомогою графових структур. Соціальні графи дозволяють формалізувати взаємозв'язки між користувачами або організаціями у вигляді вузлів (користувачів, груп, сторінок) та ребер (зв'язків між ними). Використовуючи спеціалізовані алгоритми та програмні інструменти, можна виявити ключові вузли, оцінити ефективність комунікацій та проаналізувати структуру спільнот.

Метою роботи є розробка методології моделювання та аналізу соціальних мереж з використанням графових структур для дослідження взаємодій, впливів та закономірностей у динаміці соціальних зв'язків.

Об'єктом дослідження є процеси взаємодії користувачів у соціальних мережах.

Предметом дослідження є графові методи та алгоритми моделювання і аналізу соціальних мереж.

У ході роботи будуть досліджені різні алгоритми аналізу, такі як визначення центральності, модульності та діаметра мережі, а також виконано порівняння їх ефективності в різних програмних засобах.

Практична цінність роботи полягає у можливості застосування отриманих результатів для оцінки впливу користувачів, виявлення активних спільнот та оптимізації комунікацій у соціальних мережах. Це має важливе значення для

компаній, які використовують соціальні медіа для маркетингу, аналітики та розуміння потреб аудиторії.

Таким чином, у роботі детально досліджено методи аналізу соціальних мереж за допомогою графів, що дозволяє розкрити структуру зв'язків та динаміку взаємодій у цифровому просторі.

Дана кваліфікаційна робота магістра, складається з 63 сторінок, 4 таблиць та 11 рисунків.

1 АНАЛІЗ АКТУАЛЬНОСТІ ВИКОРИСТАННЯ ГРАФОВИХ МОДЕЛЕЙ

1.1 Графові моделі як універсальний інструмент аналізу

Графові моделі є одним із найбільш універсальних інструментів для аналізу складних систем і взаємозв'язків. Їхній підхід заснований на уявленні об'єктів як вузлів, а їхніх взаємозв'язків — як ребер, що дає змогу аналізувати структури та динаміку різних систем.

Графові моделі[1] використовуються у дуже широкому спектрі галузей.

У соціології граfi використовуються для виявлення впливових осіб, аналізу спільнот і дослідження інформаційних потоків. Основними метриками в соціальних мережах є:

- 1) центральність вузлів (наприклад, Degree Centrality, Betweenness Centrality), які допомагають визначати ключових учасників мережі;
- 2) аналіз спільнот за допомогою алгоритмів, таких як Louvain Method або Girvan-Newman, дозволяє ідентифікувати групи користувачів зі схожими інтересами.

У біоінформатиці граfi застосовуються для аналізу зв'язків між білками, генами та іншими молекулярними структурами. Графові моделі дозволяють:

- 1) аналізувати протеїнові взаємодії (Protein-Protein Interaction Networks), де вузли — це білки, а ребра — їхні взаємодії;
- 2) створювати граfi метаболічних шляхів для вивчення взаємодії хімічних речовин в організмі;
- 3) моделювати генетичні зв'язки для виявлення успадкованих захворювань.

Графові моделі є незамінним інструментом у логістиці, де вони використовуються для оптимізації маршрутів і управління транспортними мережами. Основні задачі:

- 1) знаходження найкоротших шляхів за допомогою алгоритмів, таких як Dijkstra або A*;
- 2) аналіз транспортних потоків для зменшення заторів і покращення ефективності маршрутів;
- 3) оптимізація складських мереж і доставки.

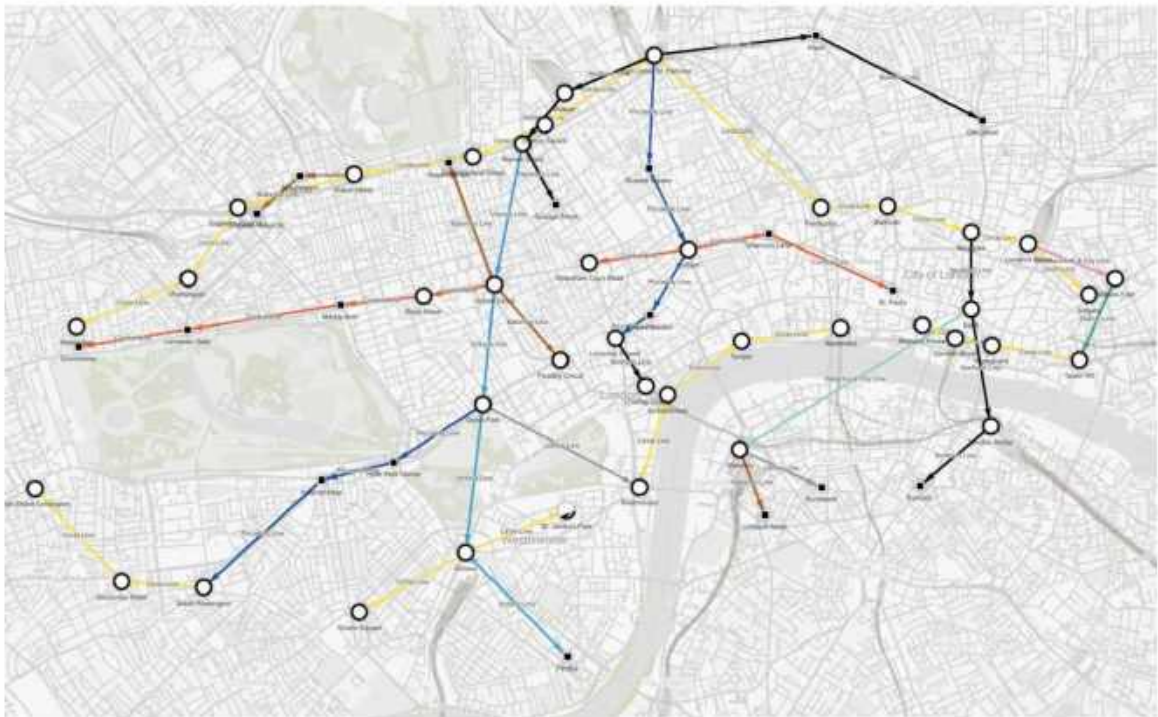


Рисунок 1.1 Приклад графу транспортної мережі міста, на якому вузли - це транспортні вузли (станції, склади), а ребра - маршрути між ними. (Джерело - <https://memgraph.com/>)

В економіці графи використовуються для аналізу взаємозв'язків між ринковими суб'єктами, потоків капіталу, а також для вивчення динаміки ринкових взаємодій. Основні задачі:

- 1) моделювання торгових мереж: компанії — вузли, контракти — ребра;
- 2) аналіз соціального капіталу: визначення ролі суб'єктів ринку в економічній екосистемі;
- 3) прогнозування поведінки ринків на основі графових метрик.

Сучасний розвиток цифрового суспільства супроводжується стрімким зростанням обсягів даних. У цьому контексті графові моделі відіграють ключову роль:

- 1) обробка великих даних: за допомогою графових баз даних, таких як Neo4j, можна аналізувати мільярди вузлів і ребер;
- 2) інтеграція з машинним навчанням: графові нейронні мережі (Graph Neural Networks) дозволяють автоматизувати аналіз складних графових структур;
- 3) виявлення аномалій: графові алгоритми використовуються для пошуку нетипових взаємозв'язків у даних (наприклад, шахрайських транзакцій у фінансових мережах).

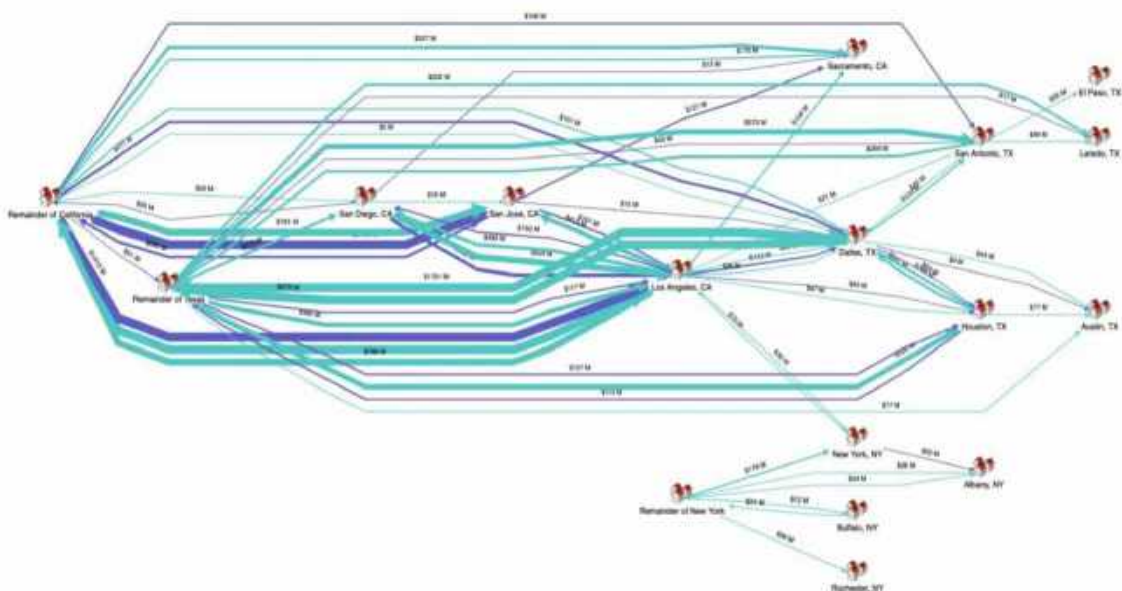


Рисунок 1.2 - Приклад графу, побудованого на великих даних (Зважений графік потоку товарів у різних географічних точках)

Графові моделі є важливим інструментом для вирішення задач у різних сферах:

- 1) забезпечують ефективний аналіз складних систем;
- 2) дозволяють моделювати взаємозв'язки та прогнозувати поведінку систем;
- 3) сприяють прийняттю оптимальних рішень на основі даних.

Таким чином, використання графів не лише сприяє ефективному управлінню інформацією, але й відкриває нові можливості для досліджень у різних галузях.

1.2 Використання графів для аналізу соціальних мереж

Соціальні мережі[2] стали основною платформою для комунікації, обміну інформацією, розваг та бізнесу. Майже кожна людина має профіль у соціальній мережі, а компанії широко використовують їх для побудови зв'язків з клієнтами та аудиторією. Аналіз соціальних мереж дозволяє глибше зрозуміти поведінкові патерни користувачів, їхні інтереси, соціальні зв'язки та вплив окремих індивідів у суспільстві. Це знання корисне як для особистих, так і для ділових стратегій, оскільки дає можливість створити більш адаптовані та цілеспрямовані маркетингові кампанії, прогнозувати соціальні тренди та розуміти поведінку аудиторії.

Графові структури є потужним інструментом для моделювання соціальних мереж, оскільки дозволяють візуально та математично відобразити складні взаємозв'язки. У соціальних мережах ці структури дозволяють:

- 1) моделювати та відображати зв'язки між індивідами чи групами;
- 2) визначати центральні вузли, або інфлюенсерів, що відіграють значну роль у поширенні інформації;

3) досліджувати структуру та щільність мережі, що вказує на ступінь зв'язності між групами.

Використання графів відкриває доступ до різних алгоритмів для оцінки зв'язків, таких як пошук коротших шляхів, обчислення центральності, кластеризація, що допомагає більш детально вивчити структуру мережі.

У соціальних мережах формуються спільноти — групи користувачів, що взаємодіють частіше між собою, ніж із зовнішнім світом. Дослідження таких спільнот є важливим для:

- 1) розуміння, як інтереси та ідеї поширюються серед певних груп;
- 2) виявлення ключових тем або трендів у межах конкретної спільноти;
- 3) аналізу поведінки аудиторії для подальшого таргетування та оптимізації комунікацій.

Графові алгоритми для виявлення спільнот, такі як Louvain або Girvan-Newman, дозволяють визначати природні угруповання у мережі, що є корисним як для соціологів, так і для маркетологів.

У соціальних мережах певні індивіди можуть мати значний вплив на аудиторію. Вони, як правило, є центральними вузлами у графі та мають великий вплив на поширення інформації. Визначення таких інфлюенсерів є корисним для маркетингових кампаній, соціологічних досліджень та прогнозування соціальної динаміки. Метрики центральності, такі як ступенева центральність, міжстороння центральність, вектор власних значень, дозволяють ідентифікувати цих ключових гравців і їхній рівень впливу на інших користувачів у мережі.

Соціальні мережі, як і інші системи, мають свою внутрішню стійкість та вразливість. Аналіз структурної стійкості мережі дозволяє зрозуміти:

- 1) як мережа реагує на втрату ключових вузлів (наприклад, блокування або відхід інфлюенсерів);
- 2) які вузли є критично важливими для зв'язності мережі;

3) як відбувається поширення інформації та як можна запобігти її небажаному поширенню (наприклад, у випадку фейкових новин).

Графові методи дозволяють проаналізувати стійкість мережі та виявити вузли, чия відсутність могла б призвести до розпаду мережі або суттєвих змін у ній.

Обсяг даних, який генерується у соціальних мережах, є величезним, і його аналіз потребує ефективних методів обробки та виявлення закономірностей. Використання графових структур у поєднанні з методами машинного навчання та штучного інтелекту дозволяє:

- 1) знаходити схожі патерни поведінки серед користувачів;
- 2) прогнозувати популярні тренди та теми, які можуть швидко набирати популярність;
- 3) здійснювати розподілену обробку даних для швидкого аналізу великих графових структур.

Соціальні мережі є потужним інструментом для бізнесу, а графовий аналіз дозволяє визначати цільові аудиторії, аналізувати вплив інфлюенсерів, розуміти потреби клієнтів і оптимізувати маркетингові стратегії. Використання графів для виявлення взаємозв'язків і сегментації користувачів допомагає компаніям точніше спрямовувати свої зусилля та досягати кращих результатів у спілкуванні з клієнтами.

1.3 Мета та завдання дослідження

Метою роботи є аналіз взаємодії суб'єктів у соціальних мережах за допомогою графових структур, визначення закономірностей у комунікації, формування кластерів взаємодії та оцінка впливовості окремих вузлів (користувачів або груп).

Для досягнення поставленої мети передбачено вирішення наступних завдань:

- 1) дослідження соціальних мереж як системи:
 - a) визначити основні характеристики соціальних мереж, їх структуру та типи взаємодії між елементами;
 - b) описати роль графових структур у моделюванні соціальних зв'язків;
- 2) збір та підготовка даних:
 - a) зібрати дані з соціальних мереж (публічні профілі, взаємозв'язки, тематичні групи, взаємодії);
 - b) підготувати дані для аналізу у вигляді графів: вузли (користувачі чи сторінки) та ребра (зв'язки між ними);
- 3) побудова графової моделі мережі:
 - a) визначити вузли мережі (користувачів, сторінки, групи);
 - b) встановити зв'язки між вузлами (підписки, згадки, коментарі, лайки);
 - c) побудувати орієнтований або неорієнтований граф залежно від характеру взаємодій;
- 4) аналіз взаємодій у мережі:
 - a) провести аналіз ступеня центральності вузлів (визначити найвпливовіші елементи мережі);
 - b) оцінити кластеризацію та виявити спільноти на основі графових алгоритмів;
 - c) дослідити закономірності у взаємодії вузлів залежно від часу, тематики або інших характеристик;
- 5) візуалізація структури соціальної мережі:
 - a) побудувати графічне представлення мережі для наочного аналізу її структури;

b) використати спеціалізовані інструменти (Gephi, NetworkX) для створення візуалізацій.

Результатом дослідження стане:

- 1) глибше розуміння структури та функціонування соціальних мереж;
- 2) виявлення впливових вузлів (інфлюенсерів) та їх ролі у поширенні інформації.

Для аналізу соціальних мереж використовуються як кількісні, так і якісні методи, які дозволяють дослідити структуру взаємодій, вплив окремих елементів і загальні закономірності функціонування мережі. Основні методи, які застосовуються в роботі:

1) методи збору даних:

a) веб-скрапінг: використовується для автоматичного збору даних із соціальних мереж (профілів, сторінок, груп) за допомогою спеціалізованих інструментів, таких як Python-бібліотеки (requests, BeautifulSoup, Selenium) або API, надані платформами;

b) API-сервіси: дозволяють отримати структуровані дані (наприклад, список друзів, підписки, реакції на публікації) легально і без зайвих витрат часу;

c) ручний збір даних: У разі обмеженого доступу до API чи неможливості автоматизації, дані збираються вручну, наприклад, через копіювання потрібної інформації зі сторінок;

2) методи обробки даних:

a) підготовка графової моделі: зібрані дані перетворюються у граф, де вузли відповідають користувачам або сторінкам, а ребра — зв'язкам між ними (підписки, лайки, згадки);

b) форматування даних: використовується для приведення даних у зручний для аналізу формат (наприклад, CSV, GML, GraphML). Для цього застосовуються інструменти, як-от Pandas у Python;

с) очищення даних: видалення дублікатів, помилкових записів або нерелевантних зв'язків для підвищення точності аналізу;

3) методи аналізу:

а) алгоритми графового аналізу:

– центральність вузлів: використовується для визначення найвпливовіших елементів у мережі, наприклад, через ступеневу центральність, міжсторонню центральність або PageRank;

– кластеризація: застосовуються алгоритми для виявлення спільнот у мережі, такі як Louvain або Girvan-Newman;

б) статистичний аналіз: визначення частоти взаємодій, середньої кількості зв'язків на одного вузла, аналіз впливу географії чи тематики на активність;

4) методи візуалізації:

а) Gephi: використовується для побудови та аналізу графів. Інструмент дозволяє створювати інтерактивні візуалізації, виявляти кластери та оцінювати основні метрики;

б) NetworkX: Python-бібліотека для створення графів і обчислення їхніх властивостей. Використовується для математичного аналізу мереж;

с) інші інструменти: наприклад, Tableau або Matplotlib для побудови діаграм і графіків;

5) експериментальні методи:

а) моделювання поширення інформації: використовується для вивчення того, як інформація чи вплив поширюється через мережу;

б) сценарний аналіз: створення моделей з видаленням вузлів або ребер для оцінки стійкості мережі до змін.

Таким чином, комбінація цих методів дозволяє повноцінно дослідити соціальні мережі, їхню структуру, функціонування та вплив окремих елементів.

2 ТЕОРЕТИЧНІ АСПЕКТИ ГРАФОВИХ СТРУКТУР ТА ЇХ ЗАСТОСУВАННЯ

2.1 Визначення графів та їх класифікація

Граф[3] — це абстрактна математична структура, що моделює систему взаємозв'язків між об'єктами. Він є фундаментальним інструментом для аналізу складних систем, які мають властивість структурованості. Графи широко використовуються в теоретичній інформатиці, соціології, біоінформатиці, логістиці, нейронауці та багатьох інших галузях, оскільки дозволяють зручним чином описати та вивчати взаємодії між компонентами системи.

Граф складається з двох основних множин:

- 1) Множина вузлів (вершин), які представляють об'єкти чи сутності.
- 2) Множина ребер, які описують зв'язки між цими вузлами.

Графи можна класифікувати за різними параметрами, що характеризують специфіку їхньої структури та поведінки в системі (рис. 2.1). Розглянемо основні типи графів, які використовуються в аналізі.

1) орієнтовані графи: орієнтований граф визначається множиною вузлів і напрямлених ребер. У такому графі кожне ребро має напрямок, що вказує, з якого вузла зв'язок виходить і в який вузол входить. Це дозволяє моделювати асиметричні взаємодії, такі як:

- a) взаємодія в соціальних мережах (користувач А підписаний на користувача В, але В не обов'язково підписаний на А);
- b) транзакції (грошові перекази між об'єктами);

2) неорієнтовані графи: неорієнтований граф характеризується тим, що його ребра не мають напрямку. Це означає, що зв'язки між вузлами є симетричними, тобто вони однаково діють в обидва боки. Такий підхід актуальний у випадках, коли зв'язки є взаємними, наприклад:

- a) дружба (користувачі А і В однаково "дружать");
 б) спільна участь у проектах (два вузли працюють над одним завданням);

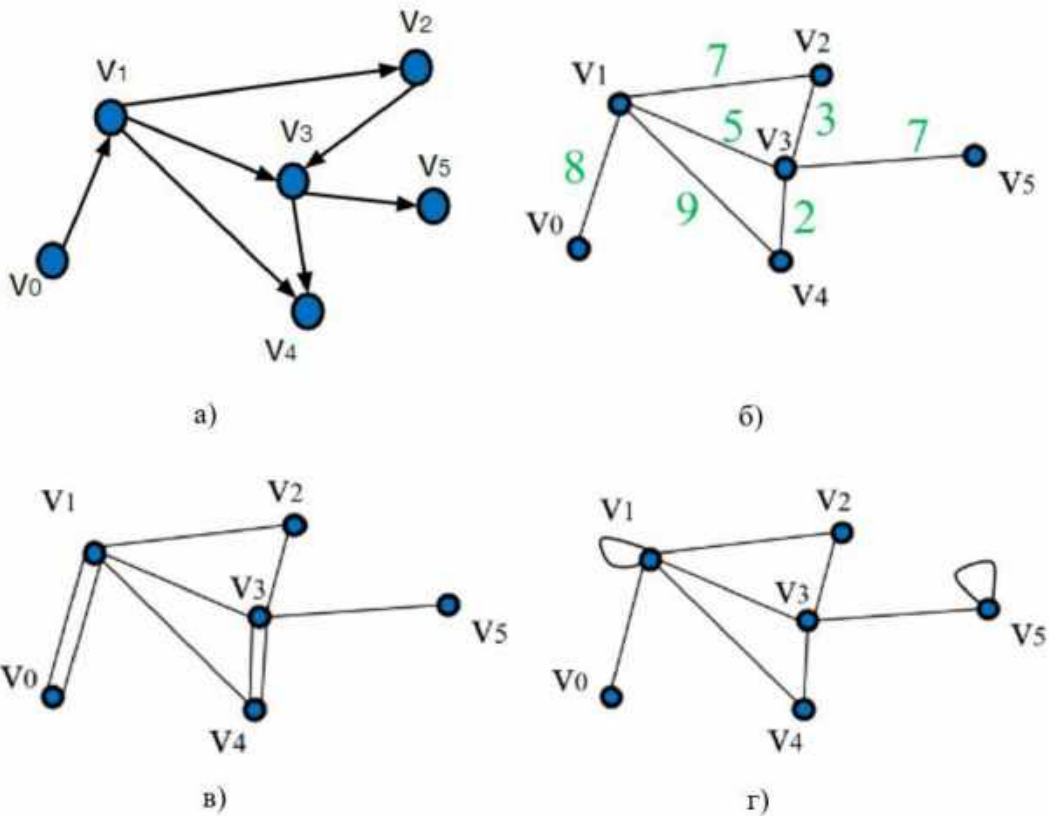


Рисунок 2.1 Види графів

(а - орієнтований, б - зважений, в - мультиграф, г - псевдограф)

3) зважені графи: зважений граф має ребра, кожне з яких має числову характеристику — вагу. Вага може відобразити інтенсивність, силу або частоту взаємодії між вузлами. Наприклад:

- а) у мережах телефонних дзвінків вага ребра може відобразити кількість дзвінків між двома абонентами;

b) у логістичних системах вага ребра може відповідати вартості транспортування;

4) незважені графи: у незважених графах усі зв'язки розглядаються як рівнозначні. Незважаючи на простоту, такі графи можуть бути корисними для базового аналізу структур мереж, де вага зв'язків не має критичного значення;

5) динамічні графи: динамічний граф враховує зміни у структурі мережі з часом. Динаміка може включати додавання нових вузлів, видалення старих або зміни у зв'язках між вузлами. Такий підхід дозволяє моделювати:

- a) еволюцію соціальних мереж;
- b) зміни у транспортних маршрутах;
- c) зростання чи зменшення впливу окремих елементів;

6) багаторівневі графи: багаторівневий граф дозволяє одночасно відображати кілька типів взаємодій між вузлами. Наприклад:

a) у соціальній мережі вузол може мати одночасно зв'язок "дружба" і "професійна взаємодія";

b) у біологічних мережах один вузол може представляти як фізичну взаємодію, так і функціональну залежність;

7) плоскі графи: ці графи можуть бути намальовані на площині так, що їхні ребра не перетинаються. Плоскі графи використовуються у географічних і транспортних моделях для відображення топологічно простих систем;

8) щільні та розріджені графи:

a) щільні графи мають велику кількість ребер порівняно з кількістю вузлів;

b) розріджені графи характеризуються відносно невеликою кількістю зв'язків.

Класифікація графів дає змогу обрати найбільш підходящу модель для вирішення конкретного завдання. Наприклад:

- 1) орієнтовані графи підходять для моделювання потоків даних чи інформації;
- 2) неорієнтовані графи використовуються для аналізу взаємних зв'язків у соціальних мережах;
- 3) зважені графи актуальні для оцінки сили зв'язків між вузлами, наприклад, у маркетингу чи логістиці;
- 4) динамічні графи необхідні для відстеження змін у реальному часі, таких як зростання мережі.

Різноманітність типів графів дозволяє адаптувати математичну модель до характеристик конкретної системи, що є критично важливим для ефективного аналізу складних мереж.

2.2 Основні поняття теорії графів

Графова теорія[4] є важливою математичною дисципліною, яка досліджує абстрактні структури, що моделюють взаємозв'язки між об'єктами. Вона базується на вивченні вузлів (вершин) і ребер, які можуть мати різні властивості та характеристики. Завдяки своїй універсальності, теорія графів знаходить широке застосування в багатьох галузях, таких як соціологія, біоінформатика, логістика, інформатика та навіть економіка.

Визначення базових елементів графа

- 1) вузли (вершини):
 - а) визначають основні об'єкти, які входять до складу системи. Наприклад, у соціальних мережах вузли можуть представляти користувачів або сторінки;
 - б) вершина може мати атрибути (імена, категорії, рейтинги), що дозволяють більш детально моделювати властивості об'єктів;
- 2) ребра:

а) це з'єднання між вузлами, що відображає певну взаємодію. Наприклад, у транспортних мережах ребра можуть позначати маршрути, а у соціальних мережах — дружбу або підписки;

б) ребра можуть бути спрямованими (орієнтованими) або неспрямованими;

3) зваженість ребер:

а) вага ребра визначає інтенсивність або значущість взаємодії. У зважених графах вага може відображати частоту взаємодій, часові затрати чи витрати ресурсів;

4) ступінь вузла (degree):

а) кількість зв'язків вузла, яка може бути поділена на вхідні (in-degree) та вихідні (out-degree) для орієнтованих графів;

б) високий ступінь вузла зазвичай вказує на його центральну роль у мережі.

Сучасний аналіз соціальних мереж активно використовує методи машинного навчання та штучного інтелекту. Одним із найбільш перспективних напрямів є застосування графових нейронних мереж (Graph Neural Networks, GNNs). Ці моделі дозволяють навчати представлення вузлів і ребер графа з урахуванням їхньої топології та властивостей, що значно підвищує точність таких задач, як:

1) класифікація вузлів (наприклад, визначення типу користувача в соціальній мережі);

2) прогнозування зв'язків (наприклад, визначення потенційних друзів);

3) виявлення аномалій (наприклад, підозрілих взаємодій у графі).

Застосування GNNs також відкриває нові можливості для автоматичного аналізу динамічних графів, дозволяючи відстежувати зміни у соціальній мережі в реальному часі. Використання таких моделей спрощує виявлення ключових вузлів, оптимізацію мережі та оцінку її стабільності.

2.3 Метрики графів

Для аналізу структури та властивостей графів використовується цілий набір метрик і характеристик:

1) центральність вузлів:

a) ступенева центральність: Простий підрахунок кількості зв'язків вузла;

b) міжстороння центральність: Визначає, наскільки часто вузол перебуває на найкоротших шляхах між іншими вузлами;

c) ейгенвекторна центральність: Враховує вплив вузла в контексті впливовості його сусідів;

2) діаметр графа: максимальна відстань між будь-якими двома вузлами.

Ця характеристика дозволяє оцінити розмір мережі та швидкість поширення інформації в ній;

3) щільність графа: співвідношення кількості фактичних зв'язків до максимально можливої кількості зв'язків. Висока щільність свідчить про тісну взаємодію між вузлами;

4) коефіцієнт кластеризації: відображає ймовірність того, що вузли, з'єднані з певним вузлом, також з'єднані між собою. Ця метрика використовується для аналізу локальних груп у мережах;

5) модульність: мірило здатності графа бути поділеним на чітко визначені кластери або спільноти. Використовується для виявлення груп вузлів із сильними внутрішніми зв'язками;

6) середня довжина шляху: визначає середню відстань між усіма парами вузлів. Ця характеристика важлива для оцінки ефективності мережі.

Графи є основним інструментом для моделювання та аналізу складних систем у різних сферах:

- 1) соціальні мережі: вивчення взаємодій між користувачами, виявлення інфлюенсерів, аналіз поширення інформації;
- 2) транспорт і логістика: оптимізація маршрутів, моделювання дорожнього трафіку;
- 3) біоінформатика: дослідження взаємодій між білками чи генами;
- 4) економіка та фінанси: моделювання ринкових зв'язків, аналіз транзакцій;
- 5) комп'ютерні науки: побудова мереж зв'язків у базах даних, аналіз інтернет-графа.

Графова теорія є важливим інструментом для розв'язання задач із великою кількістю зв'язків між об'єктами, що дозволяє не лише аналізувати структуру системи, а й прогнозувати її поведінку.

3 АЛГОРИТМИ ТА ЗАСОБИ АНАЛІЗУ ТА МОДЕЛЮВАННЯ ГРАФІВ СОЦІАЛЬНИХ МЕРЕЖ

3.1 Порівняльна характеристика програмних засобів для аналізу соціальних мереж

Аналіз соціальних мереж є важливим інструментом для дослідження, зокрема для розуміння поведінки користувачів, виявлення трендів та здійснення прийняття рішень. Існує багато програмних засобів, призначених для аналізу соціальних мереж, і у кожного з них є свої унікальні властивості і переваги.

Ражек[5] — це програмне забезпечення, спеціально розроблене для аналізу великих мереж. Назва походить від слова "павук" на словенській мові, що символізує роботу з багатокомпонентними структурами. Ражек орієнтований на аналіз великих графів із десятками чи навіть сотнями тисяч вузлів і зв'язків.

Основні можливості Ражек:

- 1) аналіз великих мереж: Ражек дозволяє працювати з графами, які важко обробляти іншими програмами через обмеження оперативної пам'яті. Завдяки оптимізованим алгоритмам, він справляється із завданнями аналізу великих мереж без значного навантаження на ресурси комп'ютера;
- 2) візуалізація графів: Інструмент підтримує різні типи візуалізації, включаючи 2D і 3D моделі. Це забезпечує зручний огляд структури мережі та її взаємозв'язків;
- 3) аналіз центральності: Ражек обчислює метрики центральності, включаючи ступеневу центральність, міжсторонню центральність, центральність за близькістю та ейгенвекторну центральність;
- 4) кластеризація: Підтримує виявлення спільнот у мережах, включаючи алгоритми поділу графів на підграфи;

5) динамічний аналіз: Можливість аналізувати мережі, що змінюються з часом. Це особливо актуально для соціальних мереж, де зв'язки між вузлами можуть постійно змінюватися;

6) аналіз взаємодій: Рајек дозволяє вивчати взаємодії між окремими вузлами або групами вузлів, що корисно для ідентифікації ключових елементів у мережі.

Gephi[6] — це універсальна програма з відкритим вихідним кодом для візуалізації та аналізу мереж. Вона є одним із найбільш популярних інструментів у сфері дослідження соціальних, економічних, наукових і навіть біологічних мереж. Gephi забезпечує інтуїтивно зрозумілий інтерфейс, потужні алгоритми обробки та широкий набір можливостей для виявлення закономірностей у складних графових структурах.

Основні характеристики Gephi

1) візуалізація мереж:

a) Gephi дозволяє створювати графічне представлення мереж із вузлів і зв'язків;

b) підтримує інтерактивний перегляд графів у режимі реального часу;

c) працює як із двовимірними (2D), так і з тривимірними (3D) мережами;

d) використовує різні алгоритми для розташування вузлів (ForceAtlas, Fruchterman-Reingold, Yifan Hu тощо), що забезпечують візуальну зрозумілість навіть для складних графів;

2) аналіз центральності:

a) Gephi дозволяє обчислювати ключові метрики центральності:

– ступенева центральність: аналізує кількість зв'язків кожного вузла;

- міжстороння центральність: оцінює, наскільки вузол є важливим посередником у мережі;
 - ейгенвекторна центральність: враховує вплив сусідніх вузлів;
 - центральність за близькістю: визначає, наскільки вузол близький до всіх інших у мережі;
- 3) виявлення кластерів та спільнот:
- a) Gephi підтримує алгоритми кластеризації для виявлення спільнот у мережі, зокрема алгоритм Louvain, який є одним із найпоширеніших методів для аналізу модульності;
 - b) дозволяє позначати кластери різними кольорами, що допомагає у візуальному аналізі спільнот;
- 4) аналіз динамічних мереж:
- a) Gephi дозволяє аналізувати мережі, що змінюються з часом:
 - додаються нові вузли чи зв'язки;
 - змінюються властивості існуючих вузлів або зв'язків;
 - b) це забезпечує розуміння еволюції мережі;
- 5) обробка великих мереж:
- a) Gephi ефективно працює із середніми та великими мережами (до 100 тисяч вузлів та ребер), що робить його придатним для дослідження складних структур;
- 6) розширюваність:
- a) Gephi підтримує плагіни, що розширюють функціонал програми:
 - наприклад, плагіни для інтеграції з базами даних або обробки тексту;
 - b) користувачі можуть розробляти власні плагіни завдяки відкритому вихідному коду;
- 7) експорт даних;

- a) графи можуть експортуватися у популярних форматах (CSV, GML, GraphML);
- b) можливість експорту візуалізацій у форматах PNG або PDF для презентацій чи публікацій.

IGraph[7] — це бібліотека для аналізу графів і мереж, розроблена для ефективної роботи з великими даними. Вона підтримує широкий спектр алгоритмів для вивчення властивостей мереж і забезпечує інтеграцію з популярними мовами програмування, такими як Python, R і C. IGraph є одним із найкращих інструментів для аналізу складних графових структур завдяки високій продуктивності, масштабованості та гнучкості.

Основні характеристики IGraph:

- 1) підтримка великих мереж:
 - a) IGraph розроблений для роботи з дуже великими графами (до мільйонів вузлів і ребер), забезпечуючи високу продуктивність;
 - b) алгоритми бібліотеки оптимізовані для обробки даних, що дозволяє використовувати її навіть на звичайних комп'ютерах;
- 2) масштабованість:
 - a) підходить для обробки як малих, так і надвеликих мереж, забезпечуючи однаково ефективну роботу;
 - b) завдяки оптимізації, бібліотека швидко обчислює ключові метрики навіть для графів із високою щільністю зв'язків;
- 3) широкий набір алгоритмів:
 - a) алгоритми кластеризації:
 - виявлення спільнот у мережі (Louvain, Walktrap);
 - розбиття графа на підграфи за різними критеріями;
 - b) центральність вузлів:

- обчислення метрик, таких як ступенева, міжстороння та ейгенвекторна центральність;
- c) аналіз шляху:
 - пошук найкоротших шляхів між вузлами;
 - аналіз діаметра графа та середньої довжини шляху;
- d) генерація графів:
 - створення випадкових графів (Erdős–Rényi), графів із безмасштабною структурою (Barabási–Albert) або графів "малого світу" (Watts–Strogatz);
- 4) гнучкість:
 - a) підтримує налаштування графів із додаванням атрибутів до вузлів і ребер (наприклад, ваги, назви або категорії);
 - b) забезпечує можливість кастомізації алгоритмів для спеціальних завдань;
- 5) інтеграція з мовами програмування
 - a) Python:
 - використовується через модуль `python-igraph`, який забезпечує простий доступ до всіх функцій бібліотеки;
 - b) C:
 - пряма робота з бібліотекою для найвищої продуктивності;
- 6) візуалізація
 - a) хоча `IGraph` не спеціалізується на візуалізації, вона підтримує базові функції для створення графіків;
 - b) для розширених можливостей візуалізації рекомендується інтеграція з `Matplotlib` (Python);
- 7) відкритий код:
 - a) бібліотека є безкоштовною і має відкритий вихідний код, що дозволяє розробникам розширювати її функціонал.

NetMiner[8] — це потужне програмне забезпечення для візуального аналізу та моделювання соціальних мереж. Відмінною рисою цього інструменту є інтеграція функціональності для глибокого аналізу мережевих структур, що включає візуалізацію, обчислення ключових метрик, кластеризацію та прогнозування. NetMiner широко використовується у сфері соціології, бізнес-аналітики, дослідження ринків та інших галузях.

Основні характеристики NetMiner:

- 1) візуальна аналітика:
 - a) інтерактивна візуалізація:
 - інструмент підтримує гнучке налаштування вигляду графів із можливістю змінювати розташування вузлів, кольори та стиль ребер;
 - користувачі можуть досліджувати мережу в реальному часі, маніпулюючи окремими вузлами та ребрами;
 - b) 3D-візуалізація:
 - забезпечує глибше розуміння складних мереж за допомогою тривимірного перегляду;
- 2) аналіз великих мереж
 - a) NetMiner підтримує роботу з великими наборами даних, включаючи мережі з десятками тисяч вузлів;
 - b) алгоритми оптимізовані для швидкого виконання розрахунків навіть на звичайних комп'ютерах;
- 3) кластеризація та модульність:
 - a) підтримка алгоритмів виявлення спільнот, таких як:
 - Louvain: для пошуку кластерів із високою модульністю;
 - Edge Betweenness: для виявлення зв'язків, що відіграють роль мостів між спільнотами;

- b) інструмент дозволяє відображати кластери різними кольорами, спрощуючи їхнє дослідження;
- 4) обчислення мережевих метрик:
 - a) центральність вузлів:
 - ступенева, міжстороння, ейгенвекторна та інші види центральності;
 - b) глобальні характеристики:
 - аналіз щільності графа, середньої довжини шляху та діаметра мережі;
 - c) локальні характеристики:
 - коефіцієнт кластеризації вузлів для оцінки локальної злагоженості;
- 5) аналіз динамічних мереж:
 - a) підтримка роботи з мережами, які змінюються з часом:
 - додавання чи видалення вузлів і ребер;
 - відстеження змін у структурі спільнот;
- 6) прогнозування:
 - a) NetMiner дозволяє моделювати поширення інформації в мережі:
 - аналіз впливу ключових вузлів;
 - оцінка впливу видалення або додавання певних зв'язків;
- 7) інтеграція з іншими інструментами:
 - a) підтримує імпорт і експорт даних у різних форматах (CSV, GraphML, Pajek NET, UCINET DL тощо);
 - b) може інтегруватися з базами даних для автоматичного оновлення даних мережі;
- 8) скриптовий інтерфейс:

а) крім графічного інтерфейсу, NetMiner підтримує створення скриптів для автоматизації задач. Це дає змогу досвідченим користувачам налаштовувати аналіз під конкретні потреби.

3.2 Зведене порівняння розглянутих програмних засобів

Таблиця 3.1 – Порівняння програмних засобів

Програмне забезпечення	Pajek	Gephi	IGraph	NetMiner
1	2	3	4	5
Тип	автономна програма	автономна програма	бібліотека для розробників	автономна програма
Мова програмування	.NET	Java, OpenGL	Python	Python
Платформа	Windows	Windows	C/Python	Windows
Швидкість обчислення	висока	висока	висока	висока
Ліцензія	безкоштовна для некомерційного використання	відкритий вихідний код	відкритий вихідний код	відкритий вихідний код

У табл. 3.1 наведене порівняння показників мережі та інструментів для аналізу графів[**Помилка! Джерело посилання не знайдено.**]:

1) ступінь центральності (Degree Centrality) – це характеристика вузла, яка визначається кількістю зв'язків, що під'єднуються до цього вузла. Вузли з вищим ступенем вважаються більш центральними. У спрямованих мережах розрізняють два типи ступеня центральності:

- a) вихідний ступінь (Out-degree) – кількість зв'язків, що виходять із вузла до інших вузлів;
 - b) вхідний ступінь (In-degree) – кількість зв'язків, що надходять до вузла від інших вузлів;
- 2) коефіцієнт кластеризації (Clustering Coefficient) – це міра, яка визначає рівень злагоженості сусідів вузла. Вона показує, наскільки вузли у графі схильні формувати трикутники, тобто взаємозв'язки між трьома вузлами;
- 3) міжстороння центральність (Betweenness Centrality) – показник центральності, який базується на комунікаційних потоках у графі. Вона оцінює кількість найкоротших шляхів у мережі, що проходять через вузол. Вузли з високою міжсторонньою центральністю є важливими "мостами", які з'єднують різні частини графа. Відсутній єдиний стандартний спосіб обчислення цього показника, але всі методи враховують структуру шляхів у графі;
- 4) центральність близькості (Closeness Centrality) – характеристика, що показує, наскільки вузол розташований близько до всіх інших вузлів у графі. Вузли з високою центральністю близькості зазвичай мають кращий доступ до всієї мережі;
- 5) діаметр мережі (Network Diameter) – геодезична характеристика графа, яка визначає найбільшу відстань між будь-якими двома вузлами в мережі. Зазвичай діаметр мережі знаходиться в діапазоні від 1 до $G-1G-1G-1$, де G – кількість вузлів у графі;
- 6) рейтинг сторінки (Page Rank) – це алгоритм ранжування вузлів, який спочатку був розроблений для пошукової системи Google. Цей алгоритм оцінює важливість вузлів у графі, враховуючи кількість і вагу зв'язків, що ведуть до них;
- 7) розбиття графа (Partition) – процес розділення графа G на менші компоненти, які відповідають певним властивостям. Це дозволяє детальніше аналізувати окремі частини мережі;

8) HITS (Hyperlink-Induced Topic Search) – алгоритм для аналізу посилань, який оцінює вузли на основі тематики. HITS враховує вихідні посилання вузлів і ранжує веб-сторінки за їхньою важливістю. Найчастіше цей алгоритм використовується для аналізу та оцінки веб-сайтів шляхом обробки всіх гіперпосилань.

Таблиця 3.2 – порівняння показників мережі та інструментів для аналізу графів

Мережеві метрики	Pajek	Gephi	IGraph	NetMiner
1	2	3	4	5
HITS	-	+	-	+
Partition	+	+	-	-
Page Rank	-	+	+	+
Network Diameter	+	+	+	+
Closeness Centrality	+	+	+	+
Betweenness Centrality	+	+	+	+
Clustering Coefficient	+	+	+	+
Degree Centrality	+	+	+	+

Однією з ключових характеристик соціальних мереж є наявність і структура спільнот. Для ідентифікації спільнот у графах було розроблено ряд спеціалізованих алгоритмів, які дозволяють ефективно визначати їхню присутність і взаємозв'язки. У таблиці 3.2 наводиться порівняння кількох таких алгоритмів, зокрема:

1) статистичні методи (Statistical Method): До цієї категорії належать методи моделювання відпалу та алгоритми на основі spin-glass. Вони використовуються для аналізу структур спільнот у графі та виявлення різних груп вузлів;

2) метод Лувена (Louvain Method): Відомий алгоритм, що базується на модульності графа. Цей підхід передбачає об'єднання вершин у спільноти з метою максимізації показника модульності;

3) жадібний алгоритм (Greedy Method): Один із класичних підходів до аналізу спільнот, який використовує жадібну оптимізацію для побудови структури графа. Цей алгоритм поступово покращує структуру спільнот шляхом послідовного виконання локально оптимальних рішень.

Таблиця 3.3 – порівняльний аналіз алгоритмів для виявлення спільнот

Вид алгоритма	Pajek	Gephi	IGraph	NetMiner
Louvain Method	+	+	+	+
Greedy Method	-	-	+	-
Statistical Method	-	-	-	-

У таблиці 3.4 наведено порівняння ключових функцій алгоритмів, таких як час завантаження, модульність, рейтинг сторінки та діаметр графа. Для цього порівняння було використано дані із набору egonets, що стосуються Facebook:

- 1) Degree Centrality: визначає центральність вузлів у мережі (кількість їхніх зв'язків);
- 2) Load Time: час завантаження даних для графа;
- 3) Network Diameter: максимальна відстань між вузлами у графі;
- 4) PageRank: алгоритм ранжування вузлів (популярний для пошуку в Google);
- 5) Modularity: оцінює якість розбиття графа на спільноти.

Таблиця 3.4 – тривалість виконання для різних інструментів

Вид алгоритма	Pajek	Gephi	IGraph	NetMiner
1	2	3	4	5
Degree Centrality	2с	4с	6,2с	15с
Load Time	3с	29с	3,7с	40с
Network Diameter	Не використовується	120с	3,5с	15с
PageRank	Не використовується	10с	9,8с	Не використовується
Modularity	6с	30с	9с	14с

Розглядаючи різноманітні інструменти для роботи зі складними мережами, зокрема Gephi, Pajek, IGraph та NetMiner, можна дійти висновку, що для аналізу соціальних мереж існує широкий вибір програмного забезпечення з різноманітними функціональними можливостями. Вибір конкретного інструмента визначається потребами користувача, його рівнем досвіду, типом даних, а також вимогами до аналізу.

Виявлення та аналіз структури спільнот є одним із ключових завдань під час роботи з соціальними мережами. Це завдання зазвичай спрямоване на вивчення функціональної організації мережі. Аналіз даних мережі дозволяє краще зрозуміти її структуру, а результати дослідження можна ефективно представити для подальшого використання. У межах цього дослідження основна увага приділялася алгоритмам, які використовуються для виявлення спільнот, та їх порівнянню. Дослідження охоплювало також аналіз мережевих метрик, таких як складність алгоритмів і ключові показники мережі, що були проаналізовані й порівняні між собою.

Інтеграція алгоритмів штучного інтелекту в програмні засоби для аналізу соціальних мереж стала ключовим напрямом розвитку. Наприклад:

— Gephi підтримує інтеграцію плагінів для використання машинного навчання, але безпосередньо GNNs не підтримуються;

— IGraph має гнучкість у застосуванні алгоритмів машинного навчання через Python API, що дозволяє користувачам поєднувати його з TensorFlow або PyTorch для навчання графових моделей;

— NetMiner пропонує вбудовані алгоритми кластеризації та аналізу, але вони обмежені класичними методами.

Подальший розвиток цих інструментів у напрямку інтеграції методів GNNs та динамічних графів відкриває перспективи для більш глибокого аналізу.

3.3 Проектування програмного засобу аналізу графової структури

У цьому підрозділі здійснена розробка ефективного алгоритму для аналізу графових структур соціальних мереж, заснованого на парсингу даних.

Функціональні вимоги до програмного рішення яке проектується наступні:

- 1) збір та парсинг даних із соціальних мереж через API або веб-скрапінг;
- 2) побудова графової моделі на основі зібраних даних;
- 3) аналіз графової структури із застосуванням алгоритмів теорії графів;
- 4) візуалізація результатів аналізу;
- 5) збереження та керування даними у базі даних для подальшого

використання.

Для проектування обрано модульну архітектуру, що забезпечує вимоги масштабованості та гнучкості. Програма складається з наступних основних компонентів[10]:

- 1) модуль збору даних (DataCollector): відповідає за отримання даних із зовнішніх джерел;
- 2) модуль парсингу (Parser): обробляє сирі дані та перетворює їх у внутрішні структури;

- 3) модуль побудови графа (GraphBuilder): створює графову модель на основі структурованих даних;
- 4) модуль аналізу (Analyzer): виконує аналіз графової структури з використанням алгоритмів теорії графів;
- 5) модуль візуалізації (Visualizer): забезпечує візуальне представлення результатів аналізу;
- 6) модуль збереження даних (DataStorage): керує збереженням та доступом до даних у базі даних.

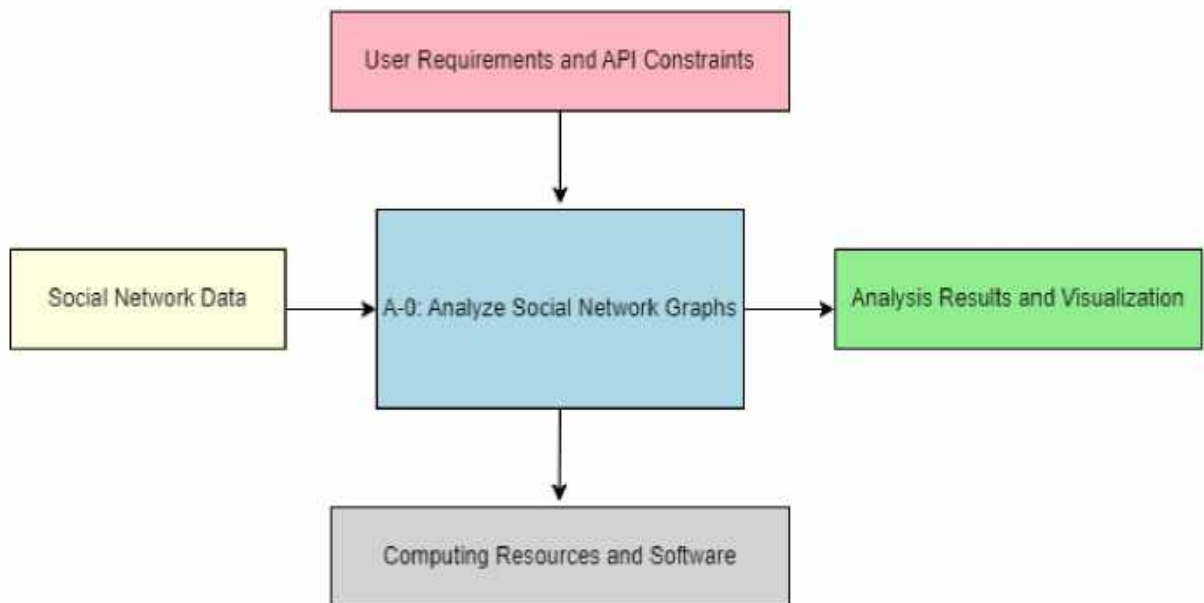


Рисунок 3.1 Контекстна діаграма

Технології та інструменти для реалізації обрано наступні: мова програмування Python з бібліотеками NetworkX (робота з графами), Pandas (обробка даних), Matplotlib/Plotly (візуалізація).

У якості бази даних доцільно використовувати SQLite (для «легкого» зберігання, при роботі з невеликими обсягами, т.з. «Proof of concept») або PostgreSQL (для масштабних даних).

Для скрапінгу використовуються модуль Requests (HTTP-запити), та BeautifulSoup (парсинг HTML), а також API клієнти для роботи з API соцмереж.

Для обробки великих обсягів даних пропонується використовувати паралельні обчислення з використанням бібліотек:

- 1) Multiprocessing: для розподілу задач на декілька процесів;
- 2) AsyncIO: асинхронне виконання завдань;
- 3) Dask: розподілені обчислення для роботи з великими даними.

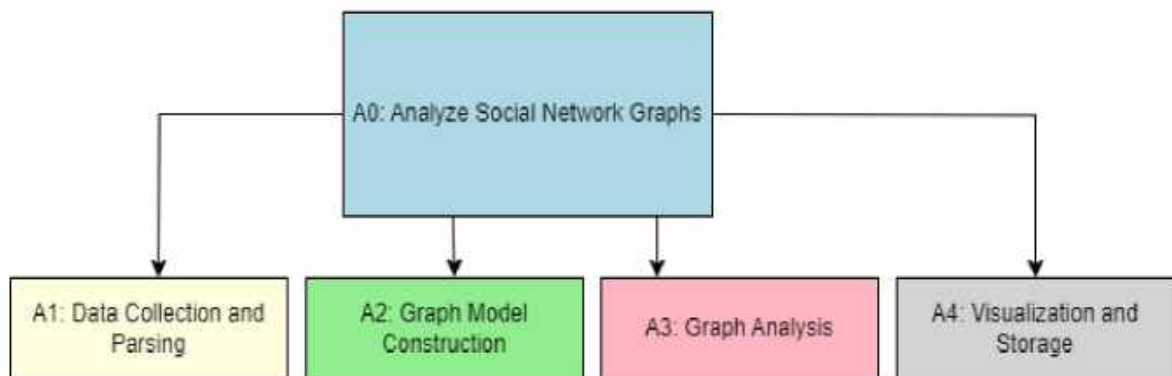


Рисунок 3.2 Декомпозиція верхнього рівня

Опис модулів програми[11]

Модуль DataCollector

- Призначення: Збір даних із різних джерел.
- Методи:
 - collect_api_data(): отримання даних через API;
 - collect_web_data(): веб-скрапінг сторінок.
- Особливості:
 - Обробка помилок з'єднання;
 - Паралельне виконання запитів для підвищення швидкості.

Модуль Parser

- Призначення: Розбір сирих даних та перетворення їх у структурований формат.

- **Методи:**

- `parse_json()`: обробка даних у форматі JSON;
- `parse_html()`: обробка HTML-контенту.

- **Особливості:**

- використання регулярних виразів для вилучення інформації;
- валідація даних.

Модуль `DataStorage`

- Призначення: Збереження та доступ до даних.

- **Методи:**

- `save_to_db()`: збереження даних у базу;
- `load_from_db()`: завантаження даних.

- **Особливості:**

- підтримка різних СУБД (SQLite, PostgreSQL);
- кешування для швидкого доступу.

Модуль `GraphBuilder`

- Призначення: Побудова графової моделі.

- **Методи:**

- `build_graph()`: створення графа з даних;
- `add_node(node)`, `add_edge(edge)`: додавання вузлів та ребер.

- **Особливості:**

- підтримка великих графів;
- оптимізація структури даних для ефективної роботи.

Модуль `Analyzer`

- Призначення: Аналіз графа з використанням алгоритмів.

- **Методи:**

- `calculate_centrality()`: обчислення центральності вузлів;
- `detect_communities()`: виявлення спільнот.
- Особливості:
 - використання бібліотеки NetworkX;
 - можливість розширення алгоритмів.

Модуль Visualizer

- Призначення: Візуалізація графів та результатів аналізу.
- Методи:
 - `draw_graph()`: побудова графічного представлення;
 - `export_visualization(format)`: експорт у різні формати.
- Особливості:
 - налаштування стилів візуалізації.

Інтерактивні графіки з використанням Plotly або Bokeh:

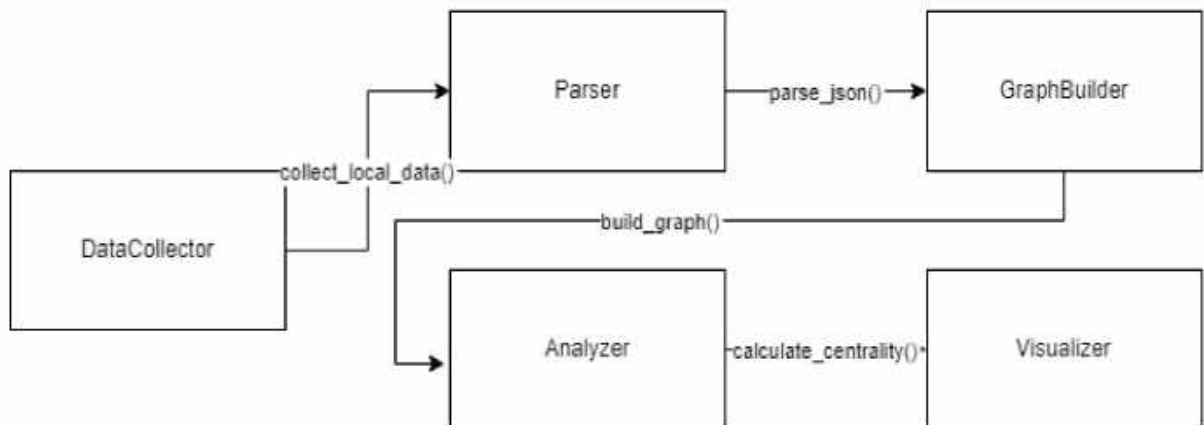


Рисунок 3.3 Сценарій процесу

Код програми має наступний вигляд:

```

import networkx as nx
import matplotlib.pyplot as plt

# Sample local data (mimicking API response)
local_data = [

```

```
    {"id": 1, "name": "Alice", "connections": [2, 3]},
    {"id": 2, "name": "Bob", "connections": [1, 3]},
    {"id": 3, "name": "Charlie", "connections": [1, 2]},
]

# Module 1: DataCollector
class DataCollector:
    def __init__(self, data=None):
        self.data = data

    def collect_local_data(self):
        if not self.data:
            raise ValueError("No local data provided")
        return self.data

# Module 2: Parser
class Parser:
    def __init__(self):
        self.data = None

    def parse_json(self, json_data):
        if not json_data:
            raise ValueError("No JSON data provided")
        return json_data

# Module 3: GraphBuilder
class GraphBuilder:
    def __init__(self):
        self.graph = nx.Graph()

    def build_graph(self, nodes, edges):
        self.graph.add_nodes_from(nodes)
        self.graph.add_edges_from(edges)

# Module 4: Analyzer
class Analyzer:
    def __init__(self, graph):
        self.graph = graph

    def calculate_centrality(self):
        return nx.degree_centrality(self.graph)

    def detect_communities(self):
```

```

        return
list(nx.community.greedy_modularity_communities(self.graph))

# Module 5: Visualizer
class Visualizer:
    def __init__(self, graph):
        self.graph = graph

    def draw_graph(self):
        plt.figure(figsize=(10, 8))
        nx.draw(self.graph,
node_color='lightblue',      edge_color='gray',      with_labels=True,
font_size=10)                node_size=500,
        plt.show()

# Main Program
if __name__ == "__main__":
    # Step 1: Collect data locally
    collector = DataCollector(data=local_data)
    collected_data = collector.collect_local_data()

    # Step 2: Parse data
    parser = Parser()
    parsed_data = parser.parse_json(collected_data)

    # Step 3: Build a graph
    builder = GraphBuilder()
    nodes = [entry['id'] for entry in parsed_data]
    edges = [(entry['id'], connection) for entry in parsed_data for
connection in entry['connections']]
    builder.build_graph(nodes, edges)

    # Step 4: Analyze the graph
    analyzer = Analyzer(builder.graph)
    centrality = analyzer.calculate_centrality()
    communities = analyzer.detect_communities()

    # Step 5: Visualize the graph
    visualizer = Visualizer(builder.graph)
    visualizer.draw_graph()

    # Print analysis results
    print("Centrality:", centrality)

```

```
print("Communities:", communities)
```

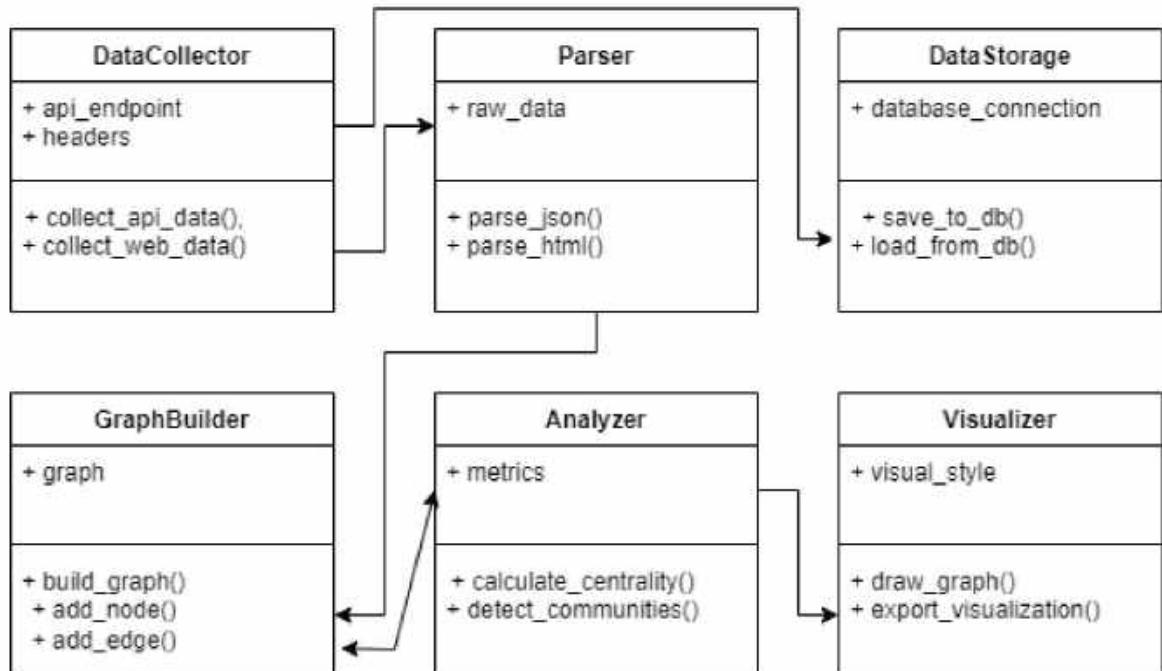


Рисунок 3.4 Діаграма класів

Розроблена програма дозволяє вирішувати завдання, пов'язані з аналізом соціальних мереж, такі як визначення впливових осіб, аналіз спільнот та моделювання взаємодій у мережах. Інтеграція всіх компонентів у єдину систему забезпечує узгодженість роботи й зручність використання. Модульна архітектура програми спрощує її вдосконалення та адаптацію до потреб користувачів. У подальшому, можна додати нові методи для роботи з іншими джерелами даних або вдосконалити алгоритми аналізу. Тестування програми показало її здатність виконувати основні завдання аналізу графових структур. Система демонструє основні принципи побудови систем для аналізу соціальних мереж та використання графових моделей.

4 ПОБУДОВА ТА ВІЗУАЛІЗАЦІЯ ГРАФА СОЦІАЛЬНОЇ МЕРЕЖІ

4.1 Опис соціальної мережі

Facebook.com — це найбільша соціальна мережа у світі, заснована у 2004 році Марком Цукербергом. Спочатку вона була створена як платформа для студентів кількох американських університетів.

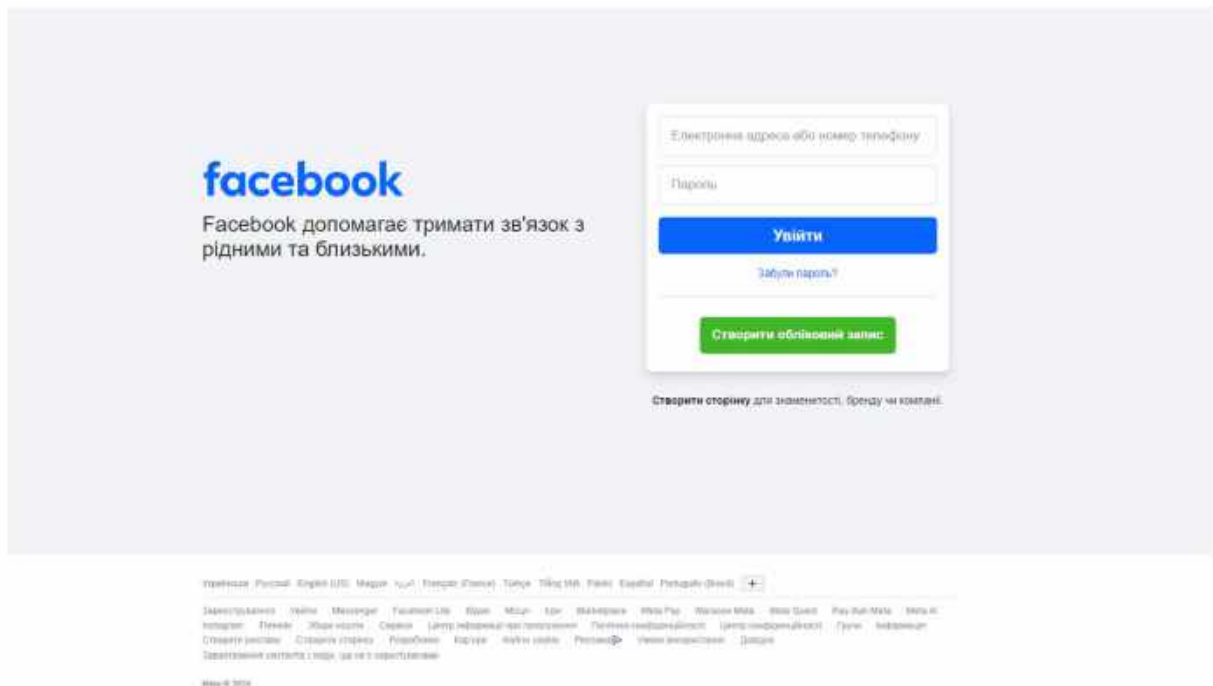


Рисунок 4.1 — Інтерфейс соціальної мережі Facebook

Використання соціальної мережі є повністю безкоштовним і доступним майже для всіх користувачів у світі. Лише в окремих країнах доступ до платформи обмежується на законодавчому рівні, однак навіть у таких умовах користувачі можуть обходити ці обмеження за допомогою VPN-сервісів.

Соціальна мережа надає широкий спектр можливостей для своїх користувачів:

- 1) створення особистого профілю, який може бути як загальнодоступним, так і приватним, залежно від налаштувань конфіденційності;
- 2) реєстрація публічних сторінок для представлення компаній, продуктів, брендів або ведення блогів;
- 3) особисті профілі та публічні сторінки мають можливість взаємодіяти між собою та демонструвати соціальну активність через такі функції, як надсилання повідомлень, створення публікацій, коментування постів, поширення контенту та висловлення емоцій (вподобання, реакції, згадування інших користувачів);
- 4) користувачам доступне створення приватних діалогів, до яких можна запрошувати інших учасників мережі;
- 5) учасники платформи можуть додавати один одного до списку друзів або підписуватися на публікації інших.

Станом на 2018 рік Facebook.com об'єднував понад два мільярди користувачів по всьому світу, що підкреслює його глобальну значущість та вплив. Така масова аудиторія створює ідеальні умови для отримання доходів, які реалізуються через монетизацію реклами та інтеграцію сторонніх додатків. При цьому розповсюджувана інформація не завжди є достовірною чи об'єктивною, що викликає питання щодо її якості та точності.

Таким чином, глибоке розуміння механізмів функціонування та аналізу соціальної мережі Facebook.com може стати важливим інструментом у сфері інформаційної безпеки та протидії дезінформації. Вивчення алгоритмів, що визначають соціальну активність користувачів, та принципів поширення інформації дозволяє більш ефективно використовувати потенціал цієї мережі для досягнення стратегічних цілей.

4.2 Проблематика створення соціального графа

На початковому етапі аналізу соціальної мережі необхідно побудувати математичну модель соціального графа, яка буде точно відображати структуру самої мережі. Теоретично це завдання виглядає досить простим: кожного користувача можна розглядати як окремий вузол графа, а зв'язки між вузлами встановлюються на основі дружніх або інших соціальних взаємодій. Проте, на практиці виникає ціла низка суттєвих труднощів, які ускладнюють реалізацію цього завдання.

Однією з основних перешкод є обмеження, накладені на взаємодію з платформою Facebook.com. Соціальна мережа дозволяє працювати з її даними лише через структурований програмний інтерфейс API, який має досить обмежений функціонал. Згідно з політикою конфіденційності Facebook, програмно неможливо отримати такі типи інформації:

- 1) особисті дані користувачів, включаючи їхню ідентифікацію чи контактні відомості;
- 2) дані про коло спілкування користувача, такі як інформація про друзів або соціальні зв'язки;
- 3) відомості про його друзів та підписників, які могли б бути основою для створення графа;
- 4) інформацію про активність користувача, наприклад, його вподобання, коментарі або публікації;
- 5) деталі про його інтереси, які допомогли б кластеризувати користувачів на основі спільних тем.

Ці обмеження створюють суттєві труднощі для побудови соціального графа, адже неможливо достовірно і повністю встановити зв'язки між користувачами. Відсутність доступу до ключової інформації призводить до того, що графи стають лише частково репрезентативними. З цієї причини завдання

побудови повноцінного соціального графа на основі даних Facebook залишається актуальною проблемою, яка досі не має конструктивного вирішення в сучасній науковій літературі.

Таким чином, обмеження доступу до даних та відсутність прозорого способу взаємодії з соціальною мережею через API ставлять серйозний виклик перед дослідниками, які намагаються розробити математичні моделі для аналізу соціальних мереж. Це підкреслює необхідність пошуку альтернативних методів, які могли б частково компенсувати нестачу інформації.

4.3 Створення соціального графа на базі аудиторій

Розглянемо реальну ситуацію, яка демонструє обмеження у доступі до даних соціальних мереж та можливі альтернативні підходи до аналізу. У зв'язку з відсутністю достатньої кількості інформації про користувачів, виникає необхідність використовувати інші методології. Оскільки дані про індивідуальних користувачів залишаються недоступними через політику конфіденційності, соціальний граф у цьому випадку буде складатися виключно з публічних сторінок.

Кожна публічна сторінка у Facebook має можливість публікувати контент, який може бути коментований користувачами. При цьому, коментарі можуть отримувати відповіді, створюючи ієрархію взаємодій. До 2018 року API Facebook дозволяло програмно отримувати дані про коментаторів публікацій у вигляді унікальних хешованих значень. Це означало, що можна було відстежити коментарі конкретного користувача, але без можливості ідентифікації його реальної особи. Таким чином, якщо один і той самий користувач коментував кілька різних публікацій, це дозволяло виявити його активність у різних дискусіях, хоча його особистість залишалася анонімною.

На основі таких даних можна побудувати аналіз перетину аудиторій між різними публікаціями. Якщо перевірити перетин аудиторій для всіх доступних публікацій, можна отримати множину зв'язків, що демонструють схожість аудиторій у межах соціальної мережі. Використовуючи ці дані, стає можливим виконати ранжування взаємозв'язків між публічними сторінками. Наприклад, якщо дві сторінки мають значний перетин аудиторії, це свідчить про їхню тематичну або цільову близькість.

Цей підхід базується на методології *data fusion*, яка дозволяє інтегрувати дані з різних джерел для отримання нового рівня аналітики. По суті, навіть за відсутності детальних даних про індивідуальних користувачів, аналіз взаємодії аудиторій публічних сторінок дозволяє виявляти ключові закономірності та будувати ефективні соціальні графи. Такий підхід є особливо цінним у контексті сучасних обмежень на доступ до даних у соціальних мережах і дає змогу отримати додаткові інсайти про структуру та взаємодію в інформаційному просторі.

Приклад взаємодії між публічними сторінками Facebook вказана на Рисунку 4.2.

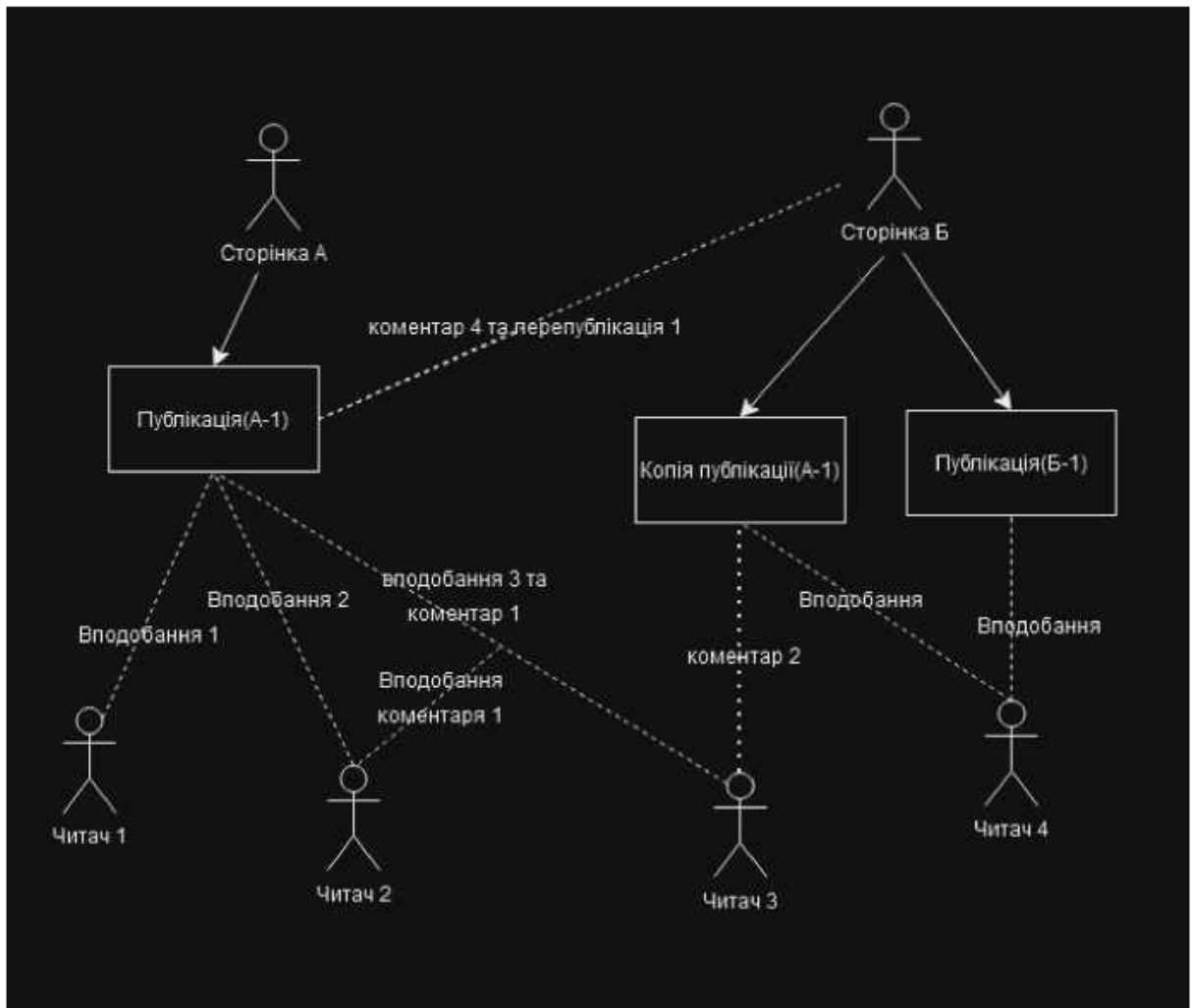


Рисунок 4.2 — Взаємодія між публічними сторінками Facebook

Опис елементів схеми:

1) Facebook сторінка А:

а) ця сторінка створює публікацію А-1;

б) публікація отримує:

— коментар 1 від читача 2, який також додає вподобання до цієї публікації;

— вподобання 1 від читача 1;

— вподобання коментаря 1 від читача 2;

- 2) поширення публікації сторінкою Б:
 - a) Facebook сторінка Б бере публікацію А-1 і створює її копію (перебирає її контент);
 - b) копія публікації також отримує:
 - коментар 2 від читача 3;
 - вподобання публікації від читача 3;
- 3) додаткові дії сторінки В:
 - a) сторінка В також створює окрему публікацію Б-1, яка:
 - отримує вподобання від читача 4;
- 4) зв'язки між сторінками та читачами:
 - a) читачі (користувачі):
 - взаємодіють із контентом сторінок через вподобання, коментарі або поширення;
 - b) коментар 4 з'являється на сторінці А, але пов'язаний із взаємодією сторінки Б;

Основні взаємодії, показані на схемі:

- 1) роль публічних сторінок:
 - a) Facebook сторінки створюють оригінальний контент, який може бути поширений іншими сторінками або користувачами;
 - b) контент може дублюватися та отримувати додаткові коментарі або вподобання;
- 2) активність користувачів:
 - a) читачі (користувачі) взаємодіють із контентом через вподобання, коментарі або відповіді на коментарі;
 - b) користувачі можуть бути активними одночасно на різних публікаціях, що створює перетини аудиторій;
- 3) перетин аудиторій:

- a) читачі сторінки А та сторінки Б можуть бути спільними (наприклад, читач З);
 - b) взаємодії на різних сторінках (через коментарі або вподобання) створюють зв'язки між цими сторінками;
- 4) поширення контенту:
- a) контент публікації А-1 розповсюджується через перебиральну активність сторінки Б.
 - b) кожна копія отримує власні коментарі та вподобання, які пов'язані із новою сторінкою.

4.4 Удосконалена методика створення соціального графа на базі аудиторій

Після 2018 року використання попереднього підходу до побудови соціального графа стало неможливим через суттєві зміни в політиці конфіденційності Facebook. Ці зміни були запроваджені після резонансного скандалу, пов'язаного з незаконним використанням даних користувачів компанією Cambridge Analytica. Основним наслідком стало те, що Facebook перестав надавати дані про осіб, які взаємодіяли з публікаціями, що виключило можливість ідентифікації коментаторів за допомогою хешованих ідентифікаторів. Таким чином, виникла необхідність розробки нового підходу, який би дозволив зберегти функціональність аналізу соціальних графів.

Для вирішення цієї проблеми можна застосувати підхід, який використовує альтернативні методи визначення схожості між коментарями та встановлення зв'язків між користувачами. Основою такого підходу є база даних, що містить усі публікації та коментарі до них. Створення зв'язків між вузлами графа базується на трьох ключових кроках:

- 1) аналіз семантичної схожості коментарів:

- a) порівнюються всі слова, що містяться в коментарях, на основі їхнього смислового значення;
 - b) запроваджується спеціальна метрика, яка оцінює рівень схожості двох різних коментарів;
 - c) використання сучасних технологій обробки природної мови (NLP) дозволяє підвищити точність аналізу, включаючи синоніми, контекст та частоту вживання слів;
- 2) аналіз гіперпосилань у коментарях:
- a) перевіряється наявність у коментарях гіперпосилань, зокрема тих, що ведуть до одного й того самого домену (внутрішнього або зовнішнього);
 - b) наявність однакових посилань може свідчити про тематичну спорідненість або взаємодію користувачів із подібним контентом;
- 3) аналіз точних збігів доменів:
- a) перевіряються коментарі на наявність однакових посилань із точним збігом доменів;
 - b) усі скорочені посилання (наприклад, bit.ly) розгортаються до їхнього кінцевого вигляду, щоб врахувати приховані збіги.

Кожен із цих кроків генерує окрему метрику, що відображає рівень схожості між коментарями. Для отримання комплексної оцінки схожості ці метрики об'єднуються з урахуванням їхньої ваги, яка визначається під час процесу калібрування параметрів. Таким чином, кожен коментар оцінюється не лише на основі його змісту, але й на основі зв'язків, які він створює через гіперпосилання.

Після визначення схожості між коментарями стає можливим повернутися до попереднього підходу аналізу. Хоча особистість коментаторів залишається невідомою, можна порівнювати аудиторії двох сторінок на основі їхніх публікацій. Це дозволяє побудувати зв'язки на соціальному графі, а також

провести мережевий аналіз для виявлення кластерів, впливових вузлів та інших ключових характеристик мережі.

Такий підхід забезпечує збереження структури соціального графа на високому рівні, навіть за умови втрати детальної інформації про окремих користувачів. Головним є те, що в мережі зберігається достатня кількість зв'язків для аналізу, хоча дотримання точної топології не є обов'язковим. Завдяки цьому методологія залишається ефективною та дозволяє адаптувати аналіз до сучасних обмежень у доступі до даних.

4.5 Реалізація алгоритму парсингу

Імпортуємо бібліотеки, які необхідні для парсингу:

— `requests` – дозволяє відправляти HTTP-запити;

— `BeautifulSoup` – дозволяє розібрати HTML-код сторінки та знайти потрібні елементи.

```
import requests
from bs4 import BeautifulSoup as bs
```

Копіюємо URL сторінки, яку потрібно парсити:

```
URL_TEMPLATE = "https://www.facebook.com/gtbank"
```

Створення списків для зберігання кількості підписників та вподобань:

```
sub_counter, like_counter = [], []
```

Створення функції для парсингу сторінки по заданому URL. Повертає два списку: підписники та вподобання:

User-Agent – заголовок для імітації запитів від реального браузеру, що сервер Facebook не блокував запити.

```
def parse(url):
    try:
        headers = {
            "User-Agent": "Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0
Safari/537.36"
        }
```

Відправляємо GET-запит на вказаний URL:

```
r = requests.get(url, headers=headers)
```

Якщо статус відповіді від серверу не дорівнює 200, виводиться повідомлення про помилку:

```
if r.status_code != 200:
    print(f"Error: Unable to access {url}, status code
{r.status_code}")
    return [], []
```

Розбираємо HTML-код сторінки за допомогою BeautifulSoup:

```
soup = bs(r.text, "html.parser")
```

Парсинг за допомогою BeautifulSoup:

- метод `find_all` шукає всі елементи сторінки з вказаним класом;
- потрібно переконавшись, що вказані класи (наприклад, `'class_name_for_subscribers'`) вірні. Вони можуть мінятися.

```
subs = soup.find_all('a',
class_='class_name_for_subscribers')
```

```

        likes = soup.find_all('a',
class_='class_name_for_likes')

        sub_counter.append(sub.text.strip())
        for like in likes:

            like_counter.append(like.text.strip())

        return sub_counter, like_counter
    except Exception as e:
        print(f"An error occurred: {e}")
        return [], []

```

Викликаємо функцію `parse` за вказаним URL:

```
sub, like = parse(URL_TEMPLATE)
```

Виводимо результат у консоль:

```
print(f"Subscribers: {sub}")
print(f"Likes: {like}")

```

4.6 Налаштування та оптимізація методів побудови

Використовуючи дані про вже існуючі соціальні графи, стає можливим провести порівняння між новоствореними графами та графами, отриманими раніше. Такий підхід є ключовим для визначення та налаштування правильних коефіцієнтів метрик, що впливають на точність моделі. Калібрування базується на використанні метрик графів, які були детально розглянуті в теоретичній частині дослідження.

Процес калібрування включає багатоетапне порівняння двох графів на основі обраних метрик. Для кожної пари графів оцінюються їхні характеристики, такі як центральність вузлів, діаметр мережі, модульність, щільність зв'язків та

інші параметри. Ці порівняння дозволяють виявити розбіжності між старими і новими графами, після чого проводиться оптимізація значень метрик.

Калібрування є ітеративним процесом. Він повторюється до тих пір, поки значення метрик архівного графа не почнуть співпадати зі значеннями метрик графа, побудованого на основі поточних даних. Це дозволяє мінімізувати відхилення та забезпечити відповідність між моделлю і реальними взаємозв'язками, представленими у графі.

Після завершення калібрування отримуємо структуровані соціальні графи, які мають високу ступінь відповідності реальним даним. Ці графи можна використовувати для проведення подальшого аналізу, такого як виявлення ключових вузлів, визначення кластерів, оцінка впливу окремих елементів мережі та моделювання динаміки їхньої взаємодії. Крім того, налаштовані графи можуть слугувати базою для розробки рекомендаційних систем, аналізу поширення інформації або оцінки ефективності маркетингових стратегій.

Таким чином, калібрування є важливим етапом у процесі побудови соціальних графів, що забезпечує їхню точність, адаптованість до реальних умов і готовність до комплексного аналізу. Завдяки цьому підходу дослідники отримують більш гнучкий інструмент для вивчення соціальних мереж та взаємозв'язків у них.

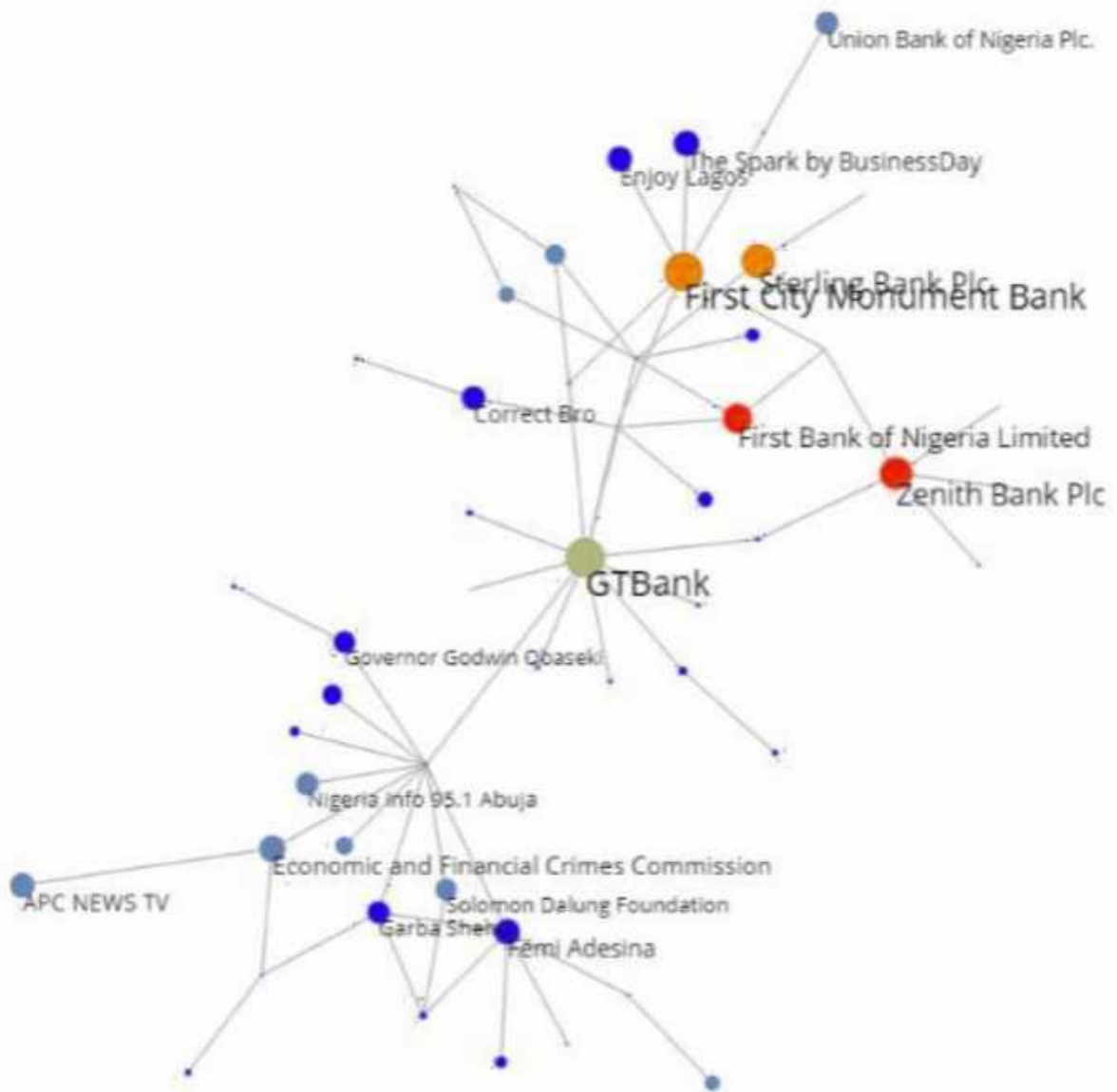


Рисунок 4.3 — Приклад створеної мережі

На зображенні представлено соціальний граф, який відображає взаємозв'язки між різними вузлами, які є компаніями. Вузли (точки) позначають об'єкти, а зв'язки між ними (лінії) демонструють певну взаємодію або спільну активність.

Основні елементи графа:

1) вузли:

а) відображають об'єкти(банки). Наприклад:

— GTBank: центральний вузол, який має найбільше зв'язків, що вказує на його високу активність або вплив;

— First City Monument Bank, Zenith Bank Plc, First Bank of Nigeria Limited та інші вузли є частиною взаємодій;

2) ребра (зв'язки):

а) лінії між вузлами показують взаємодію(спільну аудиторію, коментарі, вподобання та інші форми активності);

б) чим більше зв'язків має вузол, тим більш впливовим він є в межах цієї мережі;

3) кольори вузлів:

а) вузли мають різні кольори, які позначають належність до певних кластерів або спільнот. Наприклад:

— вузли одного кольору представляють організації чи сторінки зі спільною аудиторією або тематичною спрямованістю;

— сірий (наприклад, GTBank і Zenith Bank Plc) свідчить про банки, пов'язані спільною діяльністю;

— темно-синій та сітло-сірий вузли представляють інші типи організацій;

4) кластери:

а) групи вузлів, які мають щільні зв'язки між собою, формують окремі кластери;

б) наприклад, вузли, пов'язані з GTBank, формують основний кластер, що свідчить про його центральну роль у взаємодії;

5) впливові вузли:

ВИСНОВКИ

Аналіз соціальних графів є потужним інструментом для дослідження взаємозв'язків, динаміки та впливу в межах соціальних мереж. Застосування графових структур дозволяє виявити ключові вузли, визначити кластеризацію спільнот і оцінити ефективність комунікацій. На прикладі представлених візуалізацій можна побачити, як інформація розподіляється між різними групами, які об'єкти є центральними, а які мають периферійний вплив.

У контексті сучасних викликів, пов'язаних із доступом до даних, були запропоновані альтернативні підходи до побудови соціальних графів, засновані на аналізі аудиторій, семантичній схожості коментарів та спільних взаємодіях. Це дозволяє створювати репрезентативні моделі навіть за умови обмеженого доступу до персональних даних користувачів.

Результати таких досліджень мають широке практичне застосування: від маркетингових стратегій і політичного аналізу до моделювання інформаційного впливу та боротьби з дезінформацією. Важливим є те, що аналіз соціальних графів дозволяє зберігати баланс між приватністю даних і точністю отриманих результатів, що відповідає сучасним вимогам до етики та конфіденційності.

Таким чином, побудова та аналіз соціальних графів є перспективним напрямом досліджень, що сприяє глибшому розумінню механізмів взаємодії у цифровому середовищі та дозволяє ефективно використовувати ці знання в різних галузях.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Wasserman, Stanley; Faust, Katherine (1994). "Social Network Analysis in the Social and Behavioral Sciences". *Social Network Analysis: Methods and Applications*. Cambridge University Press. pp. 1–27. ISBN 9780521387071.
2. Боднар В.О. Соціальні мережі як інструмент комунікації у сучасному світі. — Київ: Академія соціальних наук, 2020. — 320 с.
3. Мазуренко В. В., Штовба С. Д. Огляд моделей аналізу соціальних мереж // Вісник Вінницького політехнічного інституту. 2015. № 2. С. 62–74.
4. Кравець Л.І. Теорія графів у соціальному аналізі. — Тернопіль: Економічна думка, 2018. — 270 с.
5. Pajek. URL: <http://mrvar.fdv.uni-lj.si/pajek/>.
6. Gephi. URL: <https://gephi.org/>.
7. IGraph, URL: <https://igraph.org/>.
8. NetMiner. URL: <http://www.netminer.com/main/main-read.do>.
9. Newman M.E.J. The structure and function of complex networks // *Society for Industrial and Applied Mathematics*. 2003. Vol. 45. P. 167–256.
10. Задорожний М.П. Аналіз та візуалізація даних у соціальних мережах. — Дніпро: Ліра, 2020. — 340 с.
11. Юрій Ткаченко. Алгоритми на Python — Київ: Академія, 2020. — 300 с.