Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА
на тему:
«СИСТЕМА ВИЯВЛЕННЯ КІБЕРЗАГРОЗ, ЩО СТВОРЮЮТЬСЯ СПІЛЬНОТАМИ В СОЦІАЛЬНИХ МЕРЕЖАХ. ЧАСТИНА 1. МОДУЛЬ ВИЗНАЧЕННЯ СПІЛЬНОТ З ВИКОРИСТАННЯМ КОМПОНОВКИ В ГРАФАХ»

Виконав: студент 2 курсу групи 2БС-22м
спеціальності 125 Кібербезпека
_____ Андрій НІКОЛАЙЧУК

Керівник: к. т. н., професор каф. ЗІ
_____ Наталія КОНДРАТЕНКО
«11» 12 2023 р.

Опонент: к. т. н., доцент каф. ПЗ
_____ Олександр ТКАЧЕНКО
«13» 12 2023 р.

Допущено до захисту
Завідувач кафедри ЗІ
д. т. н., проф.
_____ Володимир ЛУЖЕЦЬКИЙ
«14» 12 2023 р.

Вінниця ВНТУ – 2023 року

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації
Рівень вищої освіти II (магістерський)
Галузь знань – 12 «Інформаційні технології»
Спеціальність – 125 «Кібербезпека»
Освітньо-професійна програма – «Безпека інформаційних і комунікаційних систем»

ЗАТВЕРДЖУЮ
Зав. кафедри ЗІ, д. т. н., проф.
Володимир ЛУЖЕЦЬКИЙ
19 09 2023 року

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Ніколайчуку Андрію Вадимовичу

1. Тема роботи: «Система виявлення кіберзагроз, що створюються спільнотами в соціальних мережах. Частина 1. Модуль визначення спільнот з використанням компоновки в графах» керівник роботи: Кондратенко Наталія Романівна, к. т. н., професор, затверджені наказом ВНТУ №247 від 18.09.2023.
2. Строк подання студентом роботи – 13 грудня 2023 року
3. Вихідні дані до роботи:
   - засіб повинен запускатись в операційних системах Windows 10 та Linux;
   - засіб повинен надавати достовірність визначення спільнот у графі;
   - засіб повинен надавати перелік пов'язаних із визначеною спільнотою загроз.
4. Зміст текстової частини: Вступ. 1. Аналіз кіберзагроз в соціальних мережах за допомогою виявлення спільнот. 2. Моделі визначення спільнот в соціальних мережах за допомогою теорії графів. 3. Програмна реалізація 4. Економічне обґрунтування. Висновки. Перелік використаних джерел. Додатки.
5. Перелік графічного матеріалу.
Актуальність, мета та задачі МКР (плакат, А4). Методи визначення та аналізу спільнот в соціальних мережах (плакат, А4). Система виявлення кіберзагроз, що утворюються спільнотами соціальних мереж на основі теорії графів (плакат, А4). Сучасні підходи до виявлення аномалій в графових структурах (плакат, А4). Інтерфейс програмного засобу (плакат, А4). Алгоритм роботи методу k-середніх (плакат, А4). Алгоритм OPTICS (плакат, А4). Розподіл ступенів вузлів

(плакат, А4). Архітектура алгоритму GN (плакат, А4). Алгоритм роботи методу k-means++ (плакат, А4).

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Наталія КОНДРАТЕНКО, к.т.н., професор кафедри ЗІ	19.09	28.09
2	Наталія КОНДРАТЕНКО, к.т.н., професор кафедри ЗІ	19.09	10.10
3	Наталія КОНДРАТЕНКО, к.т.н., професор кафедри ЗІ	19.09	26.10
4	Ольга РАТУШНЯК, к. т. н., доц. каф. ЕВПМ	19.09	27.11

7. Дата видачі завдання – 1 вересня 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз завдання. Вступ	01.09.2023 – 10.09.2023	
2	Аналіз літературних джерел за напрямком магістерської кваліфікаційної роботи	10.09.2023 – 15.09.2023	
3	Науково-технічне обґрунтування	16.09.2023 – 22.09.2023	
4	Розробка технічного завдання	23.09.2023 – 29.09.2023	
5	Розробка рішень	30.09.2023 – 12.10.2023	
6	Практична реалізація, моделювання, експериментування, результати	14.10.2023 – 10.11.2023	
7	Розробка економічного обґрунтування доцільності розробки	11.11.2023 – 17.11.2023	
8	Аналіз виконання ТЗ, висновки	18.11.2023 – 24.11.2023	
9	Оформлення пояснювальної записки	25.11.2023 – 30.11.2023	
10	Попередній захист МКР та доопрацювання МКР	28.11.2023 – 01.12.2023	
11	Перевірка магістерської роботи на наявність плагіату	02.12.2023 – 10.12.2023	
12	Представлення МКР до захисту, рецензування	11.12.2023 – 14.12.2023	
13	Захист МКР	14.12.2023 – 21.12.2023	

Студент_____Андрій НІКОЛАЙЧУК

Керівник роботи_____Наталія КОНДРАТЕНКО

Магістерськ
на яких є 14 рису
найменування.

Магістерсь
виявлення кіберз
включає розроб
ідентифікації а
підвищення кіб
кіберзагроз, що
новизну та в
соціальних м
демонстрацієн
розділі оцінен

Ключ
кластериза

АНОТАЦІЯ

Магістерська кваліфікаційна робота складається з 105 сторінок формату А4, на яких є 14 рисунка, 9 таблиць, 29 формул, список використаних джерел містить 22 найменування.

Магістерська кваліфікаційна робота присвячена присвячена розробці системи виявлення кіберзагроз, що створюються спільнотами в соціальних мережах. Робота включає розробку модулю моніторингу та аналізу на основі теорії графів для ідентифікації аномалій та потенційних загроз в мережах. Головною метою є підвищення кібербезпеки в соціальних мережах і захист користувачів від можливих кіберзагроз, що виникають в цьому віртуальному середовищі. Робота має наукову новизну та велику практичну цінність, сприяючи забезпеченню безпеки в соціальних мережах. Ілюстративна частина складається з 10 плакатів з демонстрацією результатів моделювання і проведених досліджень. В економічному розділі оцінено витрати на розробку технології та програмного засобу.

Ключові слова: соціальний граф, визначення спільнот у графі, кластеризація, компонування графа, кластерний коефіцієнт.

ABSTRACT

The master's qualification work consists of 105 A4 pages, which include 14 figures, 9 tables, 29 formulas, and a list of references containing 22 titles.

The master's qualification work is devoted to the development of a system for detecting cyber threats created by communities in social networks. The work includes the development of a monitoring and analysis module based on graph theory to identify anomalies and potential threats in networks. The main goal is to improve cybersecurity in social networks and protect users from possible cyber threats arising in this virtual environment. The work has a scientific novelty and great practical value, contributing to ensuring security in social networks. The illustrative part consists of 10 posters demonstrating the results of modeling and research. The economic section estimates the costs of developing the technology and the software tool.

Keywords: social graph, definition of communities in a graph, clustering, graph layout, cluster coefficient.

ЗМІСТ

ВСТУП

В сучасному цифровому світі інтернет-спільноти соціальних мереж відіграють важливу роль у взаємодії і комунікації мільйонів користувачів по всьому світу. Однак разом з цим зростає загроза від кіберзлочинців, які використовують ці спільноти для поширення дезінформації, кібератак і інших шкідливих дій. Відсутність ефективних засобів виявлення та запобігання кіберзагрозам в соціальних мережах становить серйозну загрозу як для користувачів, так і для суспільства в цілому.

Сучасні загрози в галузі кібербезпеки надзвичайно різноманітні та динамічно розвиваються. Вони можуть приймати форму дезінформації, кібербулінгу, фішингу, атак на ідентичність та численних інших видів атак, які спрямовані на завдання шкоди індивідуальним користувачам, підприємствам та суспільству в цілому. Враховуючи серйозність цих загроз, важливо розробляти та впроваджувати ефективні заходи для їх виявлення та запобігання. Усунення цих загроз вимагає комплексного підходу, де використання теорії графів та аналізу активності спільнот відіграє ключову роль у виявленні аномалій та ефективних заходах захисту.

У даній роботі ми розглядаємо систему виявлення кіберзагроз, які виникають внаслідок діяльності самих спільнот у соціальних мережах. Ми розробляємо модуль моніторингу та аналізу, який використовує теорію графів для ідентифікації аномалій та потенційних загроз.

Актуальність. Сучасне суспільство знаходиться в стані постійного зростання залежності від інформаційних технологій та соціальних мереж. З одного боку, це принесло багато переваг і можливостей для спілкування, співпраці та обміну інформацією. З іншого боку, це також сталося джерелом серйозних кіберзагроз, які можуть вплинути на індивідів, організації та суспільство в цілому.

Злочинці, використовуючи соціальні мережі, здатні поширювати дезінформацію, проводити кібератаки, зламувати особисту інформацію та

використовувати ці можливості для своїх цілей. Актуальність даної роботи полягає в необхідності розробки системи, яка здатна виявляти та запобігати цим кіберзагрозам, зокрема тим, які формуються в межах спільнот соціальних мереж.

Підвищення кібербезпеки в соціальних мережах є актуальним завданням, оскільки це стосується безпеки мільйонів користувачів та може мати значущий вплив на суспільну діяльність, політику, економіку і інші аспекти життя. Розвиток методів виявлення кіберзагроз в спільнотах соціальних мереж має на меті забезпечити ефективний захист від небезпеки та сприяти створенню безпечного інформаційного середовища. Дана робота вирішує актуальну проблему, що вимагає ретельного наукового підходу та практичних застосувань у сфері кібербезпеки.

Об’єктом дослідження є система виявлення кіберзагроз, що створюються спільнотами в соціальних мережах, зокрема модуль визначення спільнот з використанням компоновки в графах.

Предметом дослідження є виявлення та аналіз кіберзагроз, які створюються спільнотами в соціальних мережах з використанням модулю визначення спільнот.

Метою даної магістерської кваліфікаційної роботи є розробка та впровадження системи виявлення кіберзагроз, створених спільнотами в соціальних мережах. Конкретно, метою є розробка модулю моніторингу та аналізу, який використовує теорію графів для ідентифікації аномалій та потенційних кіберзагроз в мережах. Дослідження спрямоване на створення ефективного інструменту, що дозволить виявляти небезпечні активності в соціальних мережах, а також розробку методів захисту користувачів від потенційних кіберзагроз. Мета цієї роботи полягає в сприянні забезпеченню кібербезпеки в соціальних мережах та захисті від кіберзагроз, що можуть виникати.

Для досягнення мети необхідно виконати наступні завдання:

– розробити модуль визначення спільнот;

– розробити програмний засіб;

– виконати тестування програмного засобу;

– проаналізувати результат, зробити висновки;

Методи дослідження. Для реалізації поставлених задач були використані огляд літератури, практичні експерименти, аналіз графових структур, розробку алгоритмів виявлення аномалій, моделювання, розробку програмного забезпечення, експертні опитування та інтерв'ю, статистичний аналіз даних.

Новизна одержаних результатів цієї роботи полягає в розробці модулю моніторингу та аналізу на основі теорії графів для виявлення кіберзагроз, які формуються в спільнотах соціальних мереж. Робота впроваджує підхід до аналізу активності спільнот і виявлення потенційно небезпечних ситуацій у цьому контексті, сприяючи підвищенню рівня кібербезпеки в соціальних мережах.

Практична цінність полягає в створенні ефективного інструменту виявлення та запобігання кіберзагрозам в соціальних мережах, забезпечуючи безпеку користувачів та суспільства від потенційних загроз.

1 АНАЛІЗ КІБЕРЗАГРОЗ В СОЦІАЛЬНИХ МЕРЕЖАХ ЗА ДОПОМОГОЮ ВИЯВЛЕННЯ СПІЛЬНОТ

1.1 Моделі генерації структур соціальних мереж

Сучасні соціальні мережі є важливою складовою суспільства та інформаційного простору. Вони не тільки сприяють спілкуванню між користувачами, але й відображають складні взаємодії, що відбуваються в цифровому світі. Для розуміння та аналізу цих мереж, а також для розробки ефективних систем виявлення кіберзагроз, які можуть виникати в соціальних мережах, необхідно мати глибоке уявлення про їхню структуру та механізми генерації.

Соціальні мережі відрізняються від інших типів мереж, таких як веб-мережі або транспортні мережі, через їхню складну природу та основні характеристики. Основні характеристики структур соціальних мереж включають в себе вузли, зв'язки, структуру та топологію мережі, групи та спільноти, а також розмір та масштаб мережі. Розуміння цих характеристик є ключовим для аналізу та подальшого розвитку систем виявлення кіберзагроз у соціальних мережах [1].

Вузлами в соціальних мережах є користувачі, які взаємодіють один з одним через різноманітні зв'язки. Ці зв'язки можуть бути дружбою, підписками, спільними інтересами, професійними контактами тощо. Зрозуміти природу та динаміку цих зв'язків є критично важливим для виявлення можливих загроз.

Структура та топологія мережі визначають, як взаємодіють користувачі та як інформація поширюється. Мережі можуть мати різні структури, такі як кліки, дерева, або багатошарові структури. Топологія мережі впливає на можливості передачі інформації та реакцію на різні події.

Однією з важливих характеристик соціальних мереж є наявність груп та спільнот. Групи можуть об'єднувати користувачів зі спільними інтересами або зв'язками. Спільноти відображають групи користувачів, які взаємодіють один з

одним частіше, утворюючи внутрішні підмережі в основній структурі мережі. Розуміння цих груп та спільнот може допомогти виявляти потенційно небезпечні активності та загрози в мережі.

Розмір та масштаб мережі визначаються кількістю користувачів та зв'язками між ними. Соціальні мережі можуть бути як невеликими групами користувачів, так і великими глобальними спільнотами зі складною структурою та великою кількістю взаємозв'язків. Розмір та масштаб мережі впливають на швидкість передачі інформації, можливості виявлення загроз, а також загальну стійкість мережі.

Для аналізу та моделювання структур соціальних мереж було розроблено різноманітні моделі. Соціальні мережі можна представити як графи, де вершини відповідають учасникам мережі, а ребра - їхнім взаємозв’язкам. Моделі генерації структур соціальних мереж - це алгоритми, які створюють такі графи з певними властивостями. Існує багато різних моделей, які враховують різні аспекти соціальних мереж, такі як ступеневий розподіл, кластеризація, транзитивність, гомофілія тощо. Деякі приклади моделей генерації структур соціальних мереж:

Модель Ердеша-Реньї (англ. Erdős–Rényi model) - це найпростіша модель, яка створює випадковий граф з заданою кількістю вершин і ребер. В цій моделі кожне ребро має однакову ймовірність існування, незалежно від інших ребер. Ця модель не добре відображає реальні соціальні мережі, оскільки вона не має кластеризації та схильна до утворення гігантської компоненти зв’язності.

Модель Барабаша-Альберта (англ. Barabási–Albert model) - це модель, яка створює безмасштабний граф з потужним законом розподілу ступеня вершин. В цьому графі багато вершин мають малу кількість сусідів, а декілька вершин мають дуже багато сусідів. Ця модель базується на принципах зростання та преференційного зчеплення: нова вершина додається до графа і з’єднується з існуючими вершинами з ймовірністю, пропорційною їхньому ступеню. Ця модель краще за модель Ердеша-Реньї враховує нерегулярну топологію реальних соціальних мереж [2].

Модель Ваттса-Строгатца (англ. Watts–Strogatz model) - це модель, яка створює граф з високим коефіцієнтом кластеризації та малою середньою довжиною шляху. В цьому графі вершини утворюють щільні підгрупи, але також мають короткого “маленького світу” ефект: будь-як два вершини можуть бути з’єднано шляхом з невеликою кількістю кроків. Ця модель базується на процесі перепробковки: починаючи з регулярного графа, деякі ребра випадково перенаправляються на інші вершини. Ця модель добре відображає властивості реальних соціальних мереж.

Модель Гірвана-Ньюмана (англ. Girvan–Newman model) - це модель, яка створює граф з явною спільнотною структурою. В цьому графі вершини поділяються на групи, які мають багато внутрішніх зв’язків і мало зовнішніх зв’язків. Ця модель базується на процесі додавання та видалення ребер: починаючи з набору окремих вершин, додаються ребра між вершинами однієї групи з високою ймовірністю і між вершинами різних груп з низькою ймовірністю. Потім деякі ребра видаляються з графа, щоб зменшити щільність зв’язків. Ця модель дозволяє симулювати розпад або появу спільнот в соціальних мережах.

Модель Клейна-Берга (англ. Kleinberg model) - це модель, яка створює граф з географічною структурою. В цьому графі вершини мають просторові координати, а ребра мають довжину, пропорційну відстані між вершинами. Ця модель базується на принципах локальності та довгих стрибків: кожна вершина з’єднується з певною кількістю найближчих сусідів і з декількома далекими вершинами, якими випадково обираються за експоненційним законом. Ця модель добре враховує просторовий аспект соціальних мереж.

Розуміння та використання цих моделей є важливим для розвитку систем виявлення кіберзагроз у соціальних мережах. Для успішного виявлення загроз важливо мати чітке уявлення про те, як формуються та еволюціонують структури соціальних мереж, а також як вони впливають на розподіл інформації та можливості зловживання в цих мережах.

В загальному понятті соціальна мережа – це певна конструкція яка

дозволяє нам відслідковувати відносини між індивідуумами, групами, організаціями або іншими соціальними одиницями. Соціальна мережа створюється при наявності між окремими її одиницями взаємодії, і тільки в випадку наявності цієї взаємодії можливо проводити грунтовне дослідження. Особливості соціального графа характеризується такими метриками, як: метрики взаємин, метрики зв'язків та сегментації. Для вирішення завдань з соціальним графом використовуються спеціальні моделі, за допомогою яких можна замінити «реальні» графи. За допомогою соціальних графів вирішують такі завдання, як: ідентифікація користувачів; соціальний пошук; генерація рекомендацій з вибору «друзів», медіа-контенту, новин, тощо; виявлення «реальних» зв'язків або збір відкритої інформації для моделювання графа. Обробка даних соціальних графів пов'язана з низкою проблем, як наприклад відмінності соціальних мереж, закритість соціальних даних [3]. Приклад соціального графа наведено на рис. 1.1.

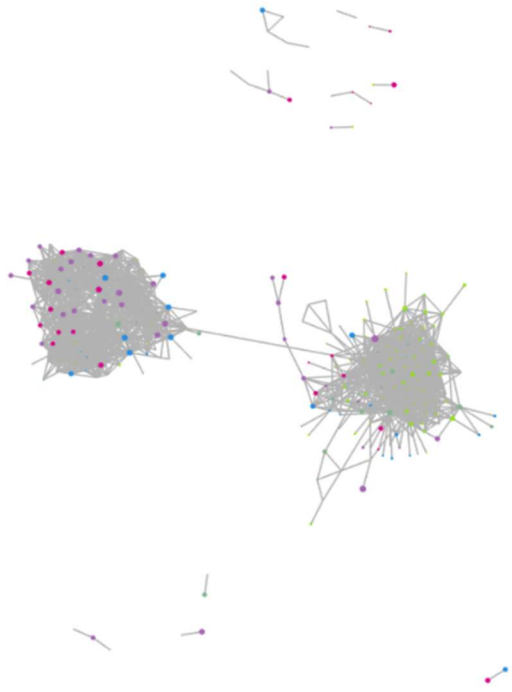


Рисунок 1.1 – Приклад соціального графу. Користувачі facebook.

Під соціальною мережею розуміється соціальна структура, яка складається з множини агентів (суб’єктів – індивідуальних чи колективних,

наприклад, індивідів, сімей, груп організацій) і визначеній на цій множині відношень (сукупності зв’язків між агентами, наприклад, знайомства, дружби, співпраці, комунікації). Формально соціальна мережа представляється як граф G(N, E), в якому N = {1,2,…,n} – скінченна множина вершин (агентів) та E – множина ребер, що відображає взаємодію агентів [4].

Соціальні мережі сприяють організації соціальних комунікацій між людьми та реалізації базових соціальних потреб. Можна виділити два трактування, які перетинаються між собою, це соціальна структури та її специфічна інернет реалізація. Техніка соціометрії (описання соціальних груп в термінах соціальних графів) була вперше запропонована в роботах Дж. Морено. Термін “соціальна мережа” був введений в 1954 році соціологом Дж. Барнсом, але масового розповсюдження термін набув починаючи з 2000 року, в зв’язку з надзвичайно швидким розвитком інтернет технологій. В наш час, багато вчених підтримують думку про існування гострого дефіциту систематичного представлення методів та алгоритмів мережевого аналізу, які були б придатні для сучасних прикладних досліджень.

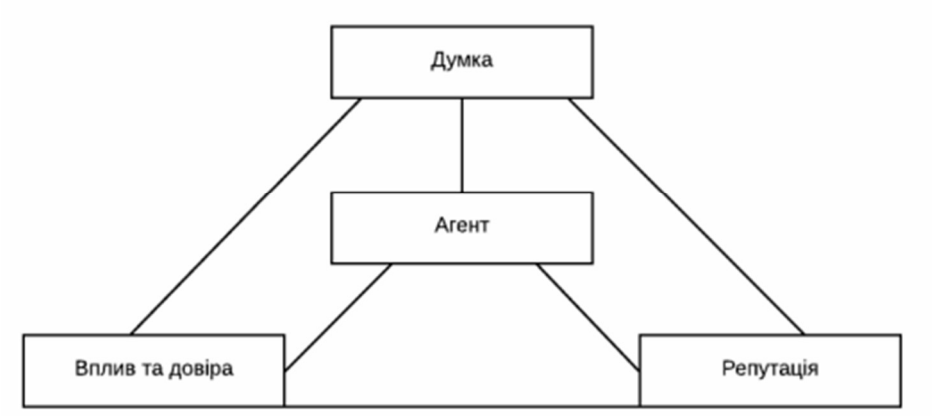


Рисунок 1.2 – Ключові слова соціальної мережі.

Можна виділити наступні переваги від користування соціальними мережами: отримання інформації (в тому числі знаходження ресурсів) від інших членів соціальної мережі, верифікація ідей через участь у взаємодіях в соціальній мережі, соціальна користь від контактів (співучасть,

самоідентифікація, соціальна рівність, соціальне сприйняття та інші); - рекреація (відпочинок, та проведення часу). Вважається, що ключовими словами будь-якої соціальної мережі є: агент, думка агента, вплив та довіра, репутація (рис. 1.2).

Facebook це найбільша в світі соціальна мережа, що почала працювати з 2004 року як мережа для студентів деяких американських університетів, її засновником є Марк Цукерберг. Користування соціальною мережею є безкоштовним, і дозволене практично усім людям. Лише деякі країни в світі обмежують доступ до соціальної мережі на законодавчому рівні, проте навіть у такій ситуації люди можуть використовувати VPN сервіси для уникнення таких обмежень [5].

На 2019 рік facebook.com налічувала більше ніж два мільярди користувачів по усьому світу. Виходячи з цього, виникає розуміння принципу отримання коштів в цій соціальній мережі. Соціальна мережа отримує кошти від реалізації та поширенні інформації у вигляді реклами або інших сторонніх додатків. Ця інформація не завжди є об’єктивною чи коректною, дуже часто навіть розробники соціальної мережі про це наголошують [6].

1.2 Національні центри кібербезпеки та їх вплив на інформаційні ресурси країни

Національні центри кібербезпеки (National Cyber Security Centers, NCSCs) в сучасному цифровому світі стали ключовими установами для забезпечення кібербезпеки національних інформаційних ресурсів та кіберінфраструктури країни. Ці центри відіграють важливу роль у протидії кіберзагрозам, координації дій, розробці та впровадженні кіберзаходів, а також у співпраці з іншими суб'єктами кібербезпеки [7].

Україна створила Національний координаційний центр кібербезпеки (НКЦК) у 2021 році як робочий орган Ради національної безпеки і оборони України (РНБО). Центр має забезпечити координацію діяльності суб’єктів

національної безпеки і оборони України під час реалізації Стратегії кібербезпеки України, підвищити ефективність системи державного управління у формування та реалізації державної політики у сфері кібербезпеки. Серед основних завдань НКЦК є: аналіз стану кібербезпеки; результатів проведення огляду національної системи кібербезпеки; стану готовності суб'єктів забезпечення кібербезпеки до виконання завдань з питань протидії кіберзагрозам; стану виконання вимог законодавства щодо кіберзахисту державних електронних інформаційних ресурсів, інформації, вимога щодо захисту якої встановлена законом, а також критичної інформаційної інфраструктури; даних про кібер-інциденти стосовно державних інформаційних ресурсів в інформаційно-телекомунікаційних системах тощо (рис 1.3). НКЦК також прогнозує та виявляє потенційні та реальні загрози у сфері кібербезпеки України, узагальнює міжнародний досвід у сфері забезпечення кібербезпеки; оперативне, інформаційно-аналітичне забезпечення РНБО з питань кібербезпеки.

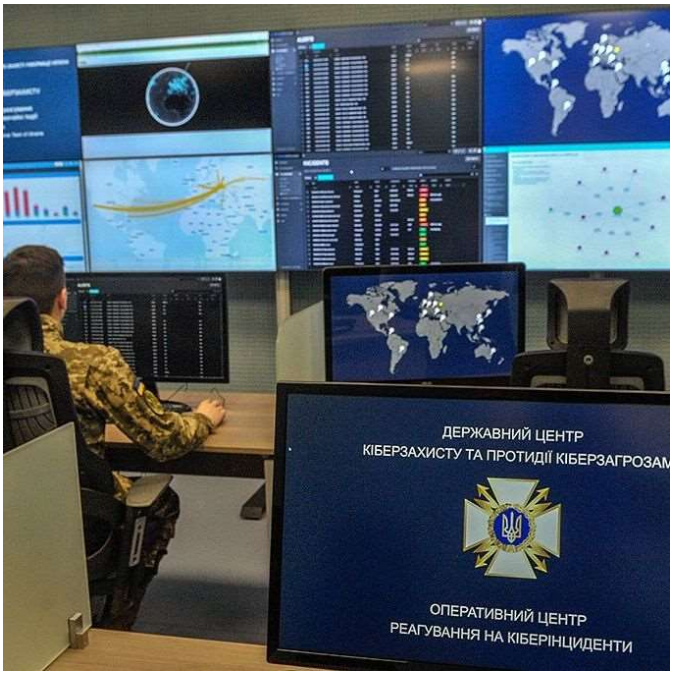


Рисунок 1.3 – Оперативний центр

НКЦК бере участь в організації і проведенні міжнаціональних і міжвідомчих кібернавчань та тренінгів, розробляє відповідні методичні

документи і рекомендації. НКЦК має право запитувати та одержувати від органів виконавчої влади, органів місцевого самоврядування, підприємств, установ і організацій статистичні дані, інформацію, довідкові та інші матеріали, необхідні для вирішення питань, що належать до його компетенції; користуватися інформаційними базами даних державних органів, державними, в тому числі урядовими, системами зв'язку і комунікацій, мережами спеціального зв'язку та іншими технічними засобами тощо.

НКЦК є важливим елементом національної системи кібербезпеки України, який сприяє захисту інформаційних ресурсів країни в умовах постійної кіберагресії з боку Російської Федерації та інших акторів. НКЦК сприяє покращенню координації та взаємодії між різними суб'єктами кібербезпеки, зокрема між цивільними та військовими структурами. НКЦК також забезпечує аналітичне та прогностичне підґрунтя для прийняття рішень РНБО щодо стратегічних напрямків розвитку країни у сфері кібербезпеки. НКЦК сприяє підвищенню свідомості громадськості та підготовці фахівців у сфері кібербезпеки. НКЦК також сприяє розширенню міжнародної співпраці України у сфері кібербезпеки з іншими державами та міжнародними організаціями [8].

Національні центри кібербезпеки виконують низку ключових функцій для забезпечення кібербезпеки в межах країни:

1. Координація відповіді на кіберзагрози: Однією з основних функцій NCSCs є координація відповіді на кіберзагрози та інциденти. Вони визначаються як центральний пункт координації дій у випадку кібератаки чи іншої кіберзагрози. Це включає співпрацю з галузевими міністерствами, правоохоронними органами, провайдерами інтернету, приватним сектором та іншими зацікавленими сторонами для впровадження відповідних заходів і відновлення інформаційних систем після інциденту.
2. Аналіз та спостереження: NCSCs відстежують кіберзагрози та виявляють нові тренди у світі кібербезпеки. Вони аналізують інциденти, виявляють

схожість між ними та визначають вектори атак. Ця інформація допомагає виявляти вразливості та розробляти відповідні стратегії захисту.

3. Співпраця та партнерство: NCSCs співпрацюють з міжнародними партнерами, іншими національними центрами кібербезпеки, а також з приватним сектором. Це сприяє обміну інформацією, розробці спільних проектів та стратегій захисту. Спільно з іншими країнами вони також можуть розробляти міжнародні стандарти та норми щодо кібербезпеки.

Вплив на інформаційні ресурси країни:

1. Захист інформаційних ресурсів: NCSCs відіграють важливу роль у захисті інформаційних ресурсів країни. Вони розробляють та впроваджують кіберзаходи, які спрямовані на забезпечення безпеки інформаційних систем та мереж. Це включає в себе захист критично важливої інфраструктури, державних інформаційних ресурсів та основних галузей економіки.
2. Підвищення обізнаності та освіти: NCSCs проводять освітні та інформаційні кампанії, спрямовані на підвищення обізнаності населення та підприємств щодо кібербезпеки. Це допомагає запобігати соціальному інженерингу, фішингу та іншим кіберзагрозам, які можуть впливати на інформаційні ресурси країни.
3. Реагування на інциденти: NCSCs забезпечують швидке та ефективне реагування на кіберінциденти. Це допомагає зменшити можливі збитки та шкоду внаслідок кібератаки на інформаційні ресурси країни.
4. Сприяння інноваціям та розвитку кібербезпеки: NCSCs сприяють розвитку та інноваціям у галузі кібербезпеки. Вони можуть фінансувати дослідження та розробки нових технологій для захисту інформаційних ресурсів країни.

1.2 Типи загроз в соціальних мережах та профілактичні заходи

Соціальні мережі, як важлива частина сучасного цифрового світу,

створюють унікальну платформу для спілкування, обміну інформацією та спільностей. Проте разом зі своєю популярністю, вони стали об'єктом різних кіберзагроз, які можуть впливати на користувачів і навіть на суспільство в цілому.

Соціальні мережі - це популярні та зручні платформи для спілкування, обміну інформацією, розваги та навчання. Однак, разом з перевагами, вони несуть і ризики для безпеки та приватності своїх користувачів. Зловмисники можуть використовувати різні методи та технології для атаки на особисті дані, фінансові ресурси, репутацію або психологічний стан користувачів соцмереж. Тому важливо бути свідомим та обережним у використанні соцмереж, а також застосовувати належні заходи захисту.

Загрози в соціальних мережах можна класифікувати за різними критеріями, наприклад, за джерелом, за ціллю, за способом реалізації тощо. Одним з можливих підходів є поділ загроз на безпосередні, опосередковані та відкриті [9]. Безпосередні загрози - це такі, які спрямовані на компрометацію або пошкодження обладнання, програмного забезпечення або даних користувача. До таких загроз належать хакерські атаки, шкідливе програмне забезпечення (віруси, трояни, черв’яки, шпигунське ПЗ тощо), атаки викрадення сесії (session hijacking), атаки викрадення ідентифікаційних даних (credential stuffing) та інші. Опосередковані загрози - це такі, які не призводять до безпосереднього пошкодження або крадіжки даних, але створюють передумови для подальших атак або порушують приватність користувача. До таких загроз належать спостереження (surveillance), спам (небажана рекламна пошта), фармінг (pharming) (перенаправлення на фальшивий сайт), фрауд (fraud) (обман для отримання вигоди) та інші. Відкриті загрози - це такі, які пов’язані з небезпечною поведінкою користувачів соцмереж або з їх невмілим використанням. До таких загроз належать формування залежності (Ігрової, комп’ютерної, Інтернет) спілкування з небезпечними людьми (збоченці, шахраї, грифери), залучення до виконання протиправних дій (хакерство, порушення авторського прав та свобод інших) та інші.

Профілактичні заходи, які можуть допомогти уникнути або зменшити наслідки загроз в соціальних мережах, можна розділити на технічні та організаційні. Технічні заходи - це такі, які пов’язані з використанням спеціального програмного забезпечення або налаштуванням параметрів безпеки. До таких заходів належать: встановлення антивірусного ПЗ та його регулярне оновлення, використання складних та унікальних паролів для кожної соцмережі та їх збереження в безпечному місці або в зашифрованому вигляді, використання двофакторної аутентифікації, якщо це можливо, використання захищених протоколів передачі даних (HTTPS), використання VPN (віртуальної приватної мережі) для підключення до публічних Wi-Fi точок, налаштування приватності свого профілю та обмеження доступу до своєї інформації для сторонніх осіб, перевірка джерела та достовірності отриманих повідомлень або запитів перед їх виконанням тощо [10]. Організаційні заходи - це такі, які пов’язані з формуванням свідомості та культури безпеки серед користувачів соцмереж. До таких заходів належать: освіта та підвищення рівня грамотності у сфері Інтернет-безпеки, розвиток критичного мислення та медіаграмотності, дотримання етики та правил поведінки в соцмережах, уникнення публікації конфіденційної або компрометуючої інформації про себе чи інших, уникнення залежності від соцмереж або перевантаження Їх інформацією, пошук та звертання до допомоги у разі потрапляння в складну ситуацію тощо.

Типи загроз в соціальних мережах:

1. Фішинг та соціальний інженеринг: Ці види атак включають у себе обман користувачів з метою витягнути конфіденційну інформацію. Зловмисники можуть виглядати як друзі або надійні джерела і намагатися отримати паролі, номери кредитних карток та інші особисті дані.
2. Кібербулінг та онлайн-гонитва: Кібербулінг в соціальних мережах може призвести до психологічних проблем і депресії. Кібербулінг включає у себе образи, залякування та інші форми насильства в мережі.
3. Поширення дезінформації та фейкових новин: Соціальні мережі часто використовуються для поширення дезінформації та фейкових новин, що

можуть впливати на громадську думку і призводити до негативних наслідків для суспільства.

4. Зловживання особистою інформацією: Зловмисники можуть використовувати особисту інформацію, яку користувачі розміщають у своїх профілях, для шахрайства, викрадення особистості чи інших злочинів.

Профілактичні заходи:

1. Особисті паролі та двофакторна аутентифікація: Використовуйте складні та унікальні паролі для своїх облікових записів на соціальних мережах. Активуйте двофакторну аутентифікацію для додаткового рівня захисту.
2. Налаштування приватності: Ретельно налаштуйте налаштування приватності на своїх облікових записах, обмежте доступ інших користувачів до вашої особистої інформації.
3. Розуміння схем фішингу та соціального інженерингу: Освоюйте навички виявлення підозрілих повідомлень та профілів, які можуть використовуватися для фішингу та соціального інженерингу.
4. Постійне моніторинг та негайна реакція: Постійно слідкуйте за активністю у своєму обліковому записі та реагуйте на будь-які підозрілі ситуації.
5. Посилення медіаграмотності: Навчайтеся розпізнавати дезінформацію та фейкові новин, перевіряйте джерела інформації перед поширенням.
6. Публічне освіту та навчання: Проводьте навчальні заходи та інформаційні кампанії щодо безпеки в соціальних мережах серед користувачів.

У заключенні, можна сказати, що соціальні мережі - це не тільки засіб комунікації і розваги, але і потенційне джерело загроз для безпеки і приватності своїх користувачів [11]. Тому необхідно бути уважним до можливих ризиків та застосовувати належні заходи профілактики та захисту. Також важливо слідкувати за новинами та оновленнями у сфері кібербезпеки, адже загрози

постійно змінюються та ускладнюються. Соціальні мережи можуть бути корисними та цікавими, якщо використовувати їх розумно та відповідально.

1.4 Методи виявлення спільнот зловмисників та спостереження за їх діяльністю

Виявлення спільнот зловмисників та спостереження за їх діяльністю в соціальних мережах є важливою складовою кібербезпеки та боротьби з кіберзлочинністю. Здатність ідентифікувати потенційних загрози та небажаних діяльностей дозволяє вчасно реагувати та запобігати можливим кібератакам, обману чи іншим кіберзлочинам. У цьому розділі розглянемо різні методи та підходи до виявлення спільнот зловмисників та спостереження за їх діяльністю в соціальних мережах.

Аналіз соціальних мереж - це процес дослідження різних систем з використанням теорії мереж [12]. Він почав широко застосовуватися саме тоді, коли стало зрозуміло, що величезна кількість наявних мереж (соціальних, економічних, біологічних) володіють універсальними властивостями: вивчивши один тип, можна зрозуміти структуру і будь-яких інших мереж та навчитися робити передбачення щодо них.

Будь-які мережі складаються з окремих учасників (людей або речей у мережі) і відносин між ними. Мережі дуже часто візуалізуються за допомогою графів - структур, що складаються з безлічі точок і ліній, які відображають зв'язки між цими точками. Учасники представлені у вигляді вузлів мережі, а їхні відносини представлені у вигляді ліній, що їх пов'язують. Така візуалізація допомагає отримати якісну та кількісну оцінку мереж:

Аналіз графів - це метод вивчення структури та властивостей графів, які є математичними моделями, що складаються з вершин та ребер [13]. Графи можуть представляти різні об’єкти та зв’язки між ними, наприклад, соціальні мережі, транспортні системи, комунікаційні мережі тощо. Аналіз графів дозволяє виявляти характеристики графів, такі як ступені вершин,

кластеризація, центральність, довжина шляху, планарність тощо. Це може бути корисно для знаходження найкоротших маршрутів, визначення ключових учасників мережі, виявлення спільнот або аномалій тощо. Для прикладу, аналіз графів може бути використаний для виявлення злочинних організацій за даними про їхню комунікацію або фінансову активність (рис 1.4).

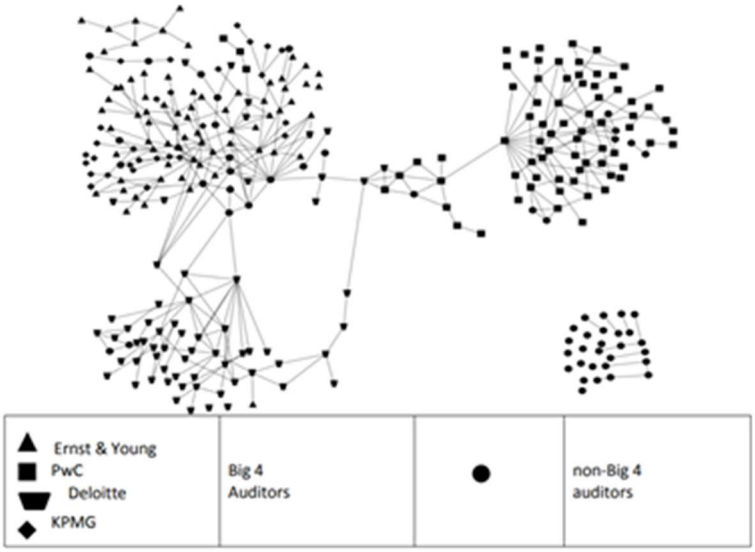


Рис. 1.4. Граф, що відображає співпрацю партнерів з аудиту в Данії у 2010-2014

Один із ефективних методів виявлення спільнот зловмисників полягає у використанні аналізу графів [6]. Соціальні мережі можна подати у вигляді графів, де користувачі є вузлами, а зв'язки між ними - ребрами. Алгоритми аналізу графів дозволяють виявити групи користувачів, які взаємодіють між собою чи мають спільні характеристики. Це може вказувати на можливі спільноти зловмисників.

Постійний моніторинг активності користувачів у соціальних мережах дозволяє виявити незвичайні патерни та аномалії. Системи аналітики великих даних можуть використовувати машинне навчання для виявлення непередбачуваних змін у поведінці користувачів, таких як надмірний обсяг повідомлень, незвичайні запити, спам чи підозрілі з'єднання.

Текстовий аналіз - це метод обробки та інтерпретації текстових даних за допомогою комп’ютерних алгоритмів. Текстовий аналіз може включати такі

завдання, як класифікація тексту за темою або тональністю, виявлення ключових слів та фраз, витягання іменованих сутностей та їхнього типу (особи, організації, місця тощо), аналогія семантичних зв’язків між словами та реченнями, генерація тексту на основі певного запиту або контексту тощо. Текстовий аналіз може бути корисний для аналогія сенсу та інформативності тексту, видобутку фактів та доказів з тексту, створення резюме або перекладу тексту тощо. Для прикладу, текстовий аналіз може бути використаний для пошуку та порівняння інформації з різних джерел про певну подію або особу.

Аналіз текстового контенту, який користувачі публікують у соціальних мережах, може допомогти виявити загрози та наміри. Алгоритми обробки природної мови можуть автоматично аналізувати текстові повідомлення та визначати ключові слова, теми та емоції, що можуть вказувати на можливі загрози або наміри зловмисників.

Підозрілі патерни поведінки - це метод виявлення незвичайних або підозрілих зразків поведінки за даними про дії та інтеракції осіб або об’єктів. Підозрілі патерни поведінки можуть свідчити про потенційне порушення правил, норм або законів, а також про наявність прихованих мотивів або інтересів. Підозрілі патерни поведінки можуть бути виявлені за допомогою статистичних або машинного навчання, що дозволяє визначити відхилення від нормальної або очікуваної поведінки за певними критеріями. Для прикладу, підозрілі патерни поведінки можуть бути використані для виявлення шахрайства, корупції, тероризму або кіберзлочинності за даними про фінансові транзакції, телефонні дзвінки, електронну пошту, соціальні мережі тощо [14].

Виявлення підозрілих патернів поведінки в соціальних мережах включає в себе аналіз взаємодій між користувачами, швидкість та обсяг публікацій, а також інші фактори, які можуть вказувати на небажану діяльність. Прикладами можуть бути автоматизовані акаунти, боти, спамери та інші форми некоректної діяльності.

Спостереження за адресою чи локацією - це метод відстеження та фіксації руху або перебування особи, речі або місця за допомогою візуальних

або технічних засобів. Спостереження за адресою чи локацією може бути проведене з метою збору інформації про діяльність, контакти, наміри або характеристики об’єкта спостереження. Спостереження за адресою чи локацією може використовувати такі методи, як фотографування, відеозапис, слідкування, GPS-трекери, супутникове зондування тощо. Для прикладу, спостереження за адресою чи локацією може бути використане для встановлення місцезнаходження злочинця, забезпечення безпеки свідка, контролю за дотриманням домашнього арешту тощо.

Використання інформації про IP-адреси та геолокацію користувачів може допомогти виявити недозволені дії. Це важливо для виявлення зловмисників, які намагаються приховати свою справжню ідентичність чи місцезнаходження.

Використання штучного інтелекту - це метод застосування комп’ютерних систем, які можуть імітувати або покращувати людські здібності до мислення, навчання, сприйняття та вирішення проблем. Використання штучного інтелекту може допомогти оптимізувати та автоматизувати різні процеси, покращити якість та швидкість прийняття рішень, розширити можливості аналогія та обробки даних, створити новий контент або продукти тощо. Використання штучного інтелекту може мати різні форми, такі як експертні системи, нейронні мережі, генетичні алгоритми, глибоке навчання, машинне навчання тощо. Для прикладу, використання штучного інтелекту може бути використане для розпізнавання обличчя, голосу, емоцій, мови, тексту, зображень тощо [15] .

Це деякі з методів аналізу даних, які можуть бути корисними для правоохоронної діяльності. Однак, це не повний перелік, а тільки приклади. Існують інші методи, які можуть бути застосовані в залежності від конкретної ситуації, цілей та джерел даних. Також важливо враховувати, що кожен метод має свої переваги та недоліки, свої можливості та обмеження. Тому необхідно бути обережним та критичним при використанні будь-якого методу, а також дотримуватися етичних та правових норм.

2 МОДЕЛІ ВИЗНАЧЕННЯ СПІЛЬНОТ В СОЦІАЛЬНИХ МЕРЕЖАХ ЗА ДОПОМОГОЮ ТЕОРІЇ ГРАФІВ

2.1 Представлення соціальних мереж графовими моделями

Соціальні мережі - це спосіб представлення соціальних взаємодій між людьми, групами, організаціями тощо. Графові моделі - це математичні структури, що складаються з вершин (вузлів) та ребер (зв'язків) між ними. Представлення соціальних мереж графовими моделями дозволяє використовувати різні методи аналізу, вимірювання, візуалізації та моделювання соціальних процесів. Це складні системи, які складаються з великої кількості суб'єктів (людей, організацій, ресурсів тощо) і їхніх взаємозв'язків. Для аналізу та дослідження таких систем потрібно мати ефективні методи представлення та обробки інформації про них. Одним з таких методів є графове моделювання, яке дозволяє використовувати математичний апарат теорії графів для опису та вивчення структури та динаміки соціальних мереж.

Графове представлення соціальних мереж базується на ідентифікації основних елементів мережі - суб'єктів і їхніх взаємодій. Суб'єкти мережі можуть бути розглянуті як вершини графа, а їхні взаємодії - як ребра, які з'єднують вершини. Залежно від типу та характеру взаємодій, ребра можуть бути орієнтованими або неорієнтованими, зваженими або незваженими, одно- або багатозначними. Також можна розглядати певні підмножини вершин або ребер як окремі сутності, наприклад, спільноти, класи, кластери тощо [16].

- Графове представлення дозволяє застосовувати ряд методик та алгоритмів для аналізу соціальних мереж, таких як:
- Визначення основних характеристик мережі, таких як щільність, довжина шляху, коефіцієнт кластеризації, ступеневий розподіл тощо.
- Виявлення ключових суб'єктів мережі за допомогою метрик центральності, таких як ступенева, посередницька, близькостна або

векторна центральність.

- Виявлення структури мережі за допомогою методів детекції спільнот, таких як модулярність, спектральна кластеризація, стохастична блокова модель тощо.
- Моделювання процесів у мережі за допомогою методів симуляції та дифузії, таких як модель СІР (англ. SIR), модель Баса (англ. Bass), модель Клейна-Берга (англ. Kleinberg) тощо.

Графове представлення соціальних мереж має ряд переваг перед іншими методами:

- Воно універсальне та гнучке: можна використовувати різні типи графів для різних типів мереж, а також комбінувати їх за потребою.
- Воно наочне та інтуітивне: можна візуалізувати графи за допомогою різних програмних засобів та аналізувати їх за допомогою геометричних або топологічних властивостей.
- Воно ефективне та обчислювальне: можна використовувати ряд оптимізованих алгоритмів для розв’язання складних задач на графах, таких як пошук найкоротшого шляху, максимального потоку, мінімального розрізу тощо.

Однак, графове представлення соціальних мереж також має деякі недоліки:

- Воно може бути неповним або неточним: не завжди можливо отримати всю інформацію про суб’єктів та їхні взаємодії, а також врахувати всі фактори, які впливають на них.
- Воно може бути складним або непередбачуваним: деякі графи можуть мати велику розмірність або складну структуру, яка утруднює їх аналіз та інтерпретацію.
- Воно може бути чутливим або нестабільним: деякі графи можуть сильно змінюватися в залежності від динаміки мережі або параметрів моделювання.

Таким чином, графове представлення соціальних мереж є потужним

інструментом для аналізу та дослідження складних соціальних систем. Однак, для його ефективного застосування потрібно враховувати його переваги та недоліки, а також обирати відповідні типи графів, методи та алгоритми для конкретних задач.

Графова модель популярна для представлення соцмереж, тому що вона дозволяє ефективно кодувати ймовірнісні залежності між великою кількістю випадкових змінних, що відображають акторів та взаємодії у соцмережах [9]. Графова модель також спрощує аналіз, вимірювання, візуалізацію та моделювання соціальних процесів, таких як поширення інформації, інфекції, інновацій, соціального впливу тощо. Графова модель може бути орієнтованою або неорієнтованою, зваженою або незваженою, одно- або багатозначною залежно від типу соцмережі та цілей дослідження.

Основні кроки представлення соціальних мереж графовими моделями такі:

- Визначення множини акторів (вершин), що складають соціальну мережу. Актори можуть бути індивідами, групами, організаціями або будь-якими іншими сутностями, що мають соціальну взаємодію.
- Визначення множини взаємодій (ребер), що з’єднують акторів. Взаємодії можуть бути односпрямованими або двоспрямованими, одноразовими або тривалими, бінарними або ваговими, позитивними або негативними тощо.
- Визначення додаткових атрибутів акторів та взаємодій, що характеризують їх властивості, статус, роль, поведінку тощо. Атрибути можуть бути категоричними або числовими, статичними або динамічними, об’єктивними або суб’єктивними тощо.
- Побудова графа G(N, E), де N - множина вершин (акторів), E - множина ребер (взаємодій), а кожна вершина та ребро мають певний набір атрибутів. Граф може бути орієнтованим або неорієнтованим, зваженим або незваженим, одно- або багатозначним тощо.

  Прикладом представлення соціальних мереж графовими моделями може

бути наступний:

- Мережа друзів у Facebook. Акторами є користувачі Facebook, взаємодіями є дружба, лайки, коментарі, повідомлення тощо. Атрибутами акторів можуть бути ім’я, вік, стать, освіта, інтереси тощо. Атрибутами взаємодій можуть бути час, тип, зміст, настрій тощо. Граф буде орієнтованим і зваженим.
- Мережа спецслужб у світовому спостереженню. Акторами є країни, що мають спецслужби, взаємодіями є співпраця, конфлікт, шпигунство, дезінформація тощо. Атрибутами акторів можуть бути регіон, рівень розвитку, політична система, військова потуга тощо. Атрибутами взаємодій можуть бути мета, засоби, результат, вплив тощо. Граф буде орієнтованим і зваженим.

Представлення соціальних мереж графовими моделями дозволяє застосовувати різні методи аналізу, такі як:

- Вимірювання характеристик акторів та взаємодій, таких як ступінь центральності, престиж, авторитет, сила зв’язку, еквівалентність тощо.
- Виявлення структурних патернів у мережі, таких як кластери, спільноти, сегменти, ядра, зв’язні компоненти тощо.
- Моделювання процесів у мережі, таких як поширення інформації, інфекції, інновацій, соціального впливу тощо.
- Візуалізація мережі за допомогою графічних зображень, що демонструють розташування та зв’язки акторів та взаємодій.

Графові моделі дозволяють використовувати математичну теорію графів для аналізу структури та динаміки соцмереж. Графи складаються з вершин, які представляють суб’єктів мережі, і ребер, які представляють їхні взаємодії [12]. Залежно від типу та характеру взаємодій, ребра можуть мати різні властивості, такі як напрямок, вагу, кольори тощо. Ось деякі приклади графових моделей соцмереж:

- Граф спільнот: цей граф показує, як суб’єкти мережі утворюють групи або спільноти, які мають багато внутрішніх зв’язків і мало зовнішніх

зв’язків. Кожна спільнота може мати свій колір або мітку, щоб позначити її ідентичність або характеристики.

- Граф центральності: цей граф показує, як суб’єкти мережі мають різний рівень впливу або значимості в мережі. Це можна виміряти за допомогою різних метрик центральності, таких як ступенева, посередницька, близькостна або векторна центральність.
- Граф дифузії: цей граф показує, як інформація або інновації поширюються у мережі за допомогою процесу дифузії. Дифузія - це процес, коли суб’єкти мережі приймають нову ідею або поведінку в залежності від їхнього стану та стану їхнього оточення.

Отже, графове представлення є ефективним інструментом для аналізу соцмереж, тому що воно дозволяє:

- Використовувати математичний апарат теорії графів для опису та вивчення структури та динаміки соцмереж. Теорія графів надає ряд концепцій, метрик, алгоритмів та теорем, які можуть бути застосовані до різних аспектів соцмереж, таких як щільність, центральність, спільноти, дифузія тощо.
- Використовувати різні типи графів для різних типів мереж, а також комбінувати їх за потребою. Графи можуть мати різні властивості, такі як орієнтованість, зваженість, кольорованість тощо, які відображають різні характеристики суб’єктів та їхніх взаємодій. Наприклад, можна використовувати орієнтовані графи для показу напрямку впливу або інформації, зважені графи для показу сили або частоти взаємодій, кольоровані графи для показу належності до спільнот або категорій тощо.
- Використовувати візуалізацію графів для наочного та інтуітивного аналізу соцмереж. Візуалізація дозволяє показати граф у просторовій формі, де можна спостерігати розташування, розмір, форму, колір та інші атрибути вершин і ребер. Візуалізація також дозволяє застосовувати різні методи геометричного або топологічного аналізу, такі як масштабування, проекція, фільтрація, кластеризація тощо. Наприклад, можна

використовувати Gephi - це програмний засіб для візуалізації і аналізу графів.

2.2 Компоновка графів на підграфи та суграфи

Компоновка графів на підграфи та суграфи - це процес розбиття графа на менші частини, які мають певні властивості або функції. Цей процес може бути корисним для аналізу структури та динаміки графа, а також для спрощення алгоритмів та обчислень на графах [15].

Підграф графа G - це граф, який складається з деякої підмножини вершин та ребер G. Суграф графа G - це підграф, який містить усі ребра G, що з'єднують вершини підграфа. Породжений підграф - це суграф, який утворений з певної підмножини вершин G разом з усіма ребрами, що з'єднують пари вершин з цієї підмножини.

Суграф - це спеціальний випадок підграфа, який утворений з частини вершин існуючого графа і всіх ребер, які з'єднують ці вершини. Тобто, суграф є повним підграфом початкового графа.

Індукований підграф - це підграф, який утворений з частини вершин існуючого графа і всіх ребер, які мають обидва кінці в цих вершинах. Тобто, індукований підграф є максимальним суграфом початкового графа.

Спануючий підграф - це підграф, який містить всі вершини початкового графа, але може мати менше ребер. Тобто, спануючий підграф є мінімальним підграфом, який зберігає зв'язність початкового графа.

Спануюче дерево - це спануючий підграф, який є деревом, тобто не містить циклів. Спануюче дерево можна знайти для будь-якого зв'язного неорієнтованого графа. Спануючих дерев може бути багато для одного і того ж графа.

Компоновка графів на підграфи та суграфи може мати різні застосування в теорії та практиці. Ось деякі приклади:

- Виявлення спільнот у соціальних мережах. Спільнотою називається суграф, в якому вершини мають багато внутрішніх зв’язків і мало зовнішніх зв’язків. Виявлення спільнот дозволяє аналізувати структуру та динаміку соціальних мереж, а також виявляти ключових учасників, лідерів, опінійних лідерів тощо.
- Побудова маршрутів у транспортних мережах. Маршрутом називається підграф, який складається з послідовності ребер, якими можна проїхати від однієї вершини до іншої. Побудова маршрутів дозволяє оптимізувати час, витрати, енергію тощо при пересуванні по транспортних мережах.
- Розфарбування графів у теорії кодування. Розфарбуванням графа називається призначення кольорів вершинам графа таким чином, що жодні дві суміжні вершини не мають одного кольору. Розфарбування графів дозволяє кодувати інформацію за допомогою мінімальної кількості символів, а також запобігати помилкам при передачі інформації.

Компоновка графів на підграфи та суграфи - це процес розбиття графа на менші частини, які мають певні властивості або функції. Для виявлення спільнот у соціальних мережах, ми можемо використовувати такі види компоновки:

- Компоновка на суграфи: цей вид компоновки базується на ідеї, що вершини, які належать одній спільноті, мають багато внутрішніх зв’язків і мало зовнішніх зв’язків. Тобто, ми шукаємо суграфи, які є щільними підмножинами вершин і ребер початкового графа. Це можна зробити за допомогою різних методів, таких як модулярність, спектральна кластеризація, стохастична блокова модель тощо.
- Компоновка на підграфи: цей вид компоновки базується на ідеї, що вершини, які належать одній спільноті, мають схожі характеристики або поведінку. Тобто, ми шукаємо підграфи, які є підмножинами вершин початкового графа, але не обов’язково всіх ребер, які з’єднують ці вершини. Це можна зробити за допомогою різних методів, таких як атрибутна кластеризація, тематичне моделювання, нейронні мережі тощо.

Компоновка графів на підграфи та суграфи може мати різні застосування та цілі. Наприклад:

- Компоновка може допомогти спростити аналіз складних графів, виділяючи їх основні структурні елементи, такі як кластери, спільноти, сегменти, ядра, зв’язні компоненти тощо.
- Компоновка може допомогти виявити важливих акторів та взаємодії в соціальних мережах, вимірюючи їх характеристики, такі як ступінь центральності, престиж, авторитет, сила зв’язку, еквівалентність тощо.
- Компоновка може допомогти моделювати процеси в графах, такі як поширення інформації, інфекції, інновацій, соціального впливу тощо.
- Компоновка може допомогти візуалізувати графи за допомогою графічних зображень, що демонструють розташування та зв’язки акторів та взаємодій.

Для компоновки графів на підграфи та суграфи існують різні методи та алгоритми. Наприклад:

- Метод рекурсивного подання (recursive decomposition) полягає в розбитті графа на два або більше попарно неперетинаючихся підграфи за допомогою розриву (cut) або сепаратору (separator), і повторенню цього процесу для кожного отриманого підграфа до досягнення базового випадку.
- Метод спектральної кластеризації (spectral clustering) полягає в застосуванню методу кластеризації до спектральних характеристик матриць суміжностей або лапласян графу для видобуття його кластерної структури.
- Метод модулярностей (modularity) полягає в максимізації функції модулярності, яка вимірює різницю між кількістю ребер в підграфах та очікуваною кількістю ребер в випадкових графах з такою ж ступеневою послідовністю.
- Метод мінімального розрізу (minimum cut) полягає в знаходженні такого розриву (cut) графа, який мінімізує сумарну вагу ребер, що перетинають

розрив. Цей метод може застосовуватися для вирішення задач сегментації зображень, кластеризації даних, максимального потоку тощо [10].

- Метод кернігана-ліна (Kernighan-Lin) полягає в розбитті графа на два попарно неперетинаючихся підграфи однакового розміру за допомогою обміну пар вершин, що максимізує сумарну вагу ребер, що з'єднують підграфи. Цей метод може застосовуватися для оптимізації розташування елементів на електронних схемах.
- Метод графового спектру (graph spectrum) полягає в використанні власних значень та власних векторів матриць суміжностей або лапласян графу для характеризації його структурних та динамічних властивостей. Цей метод може застосовуватися для аналізу спектральної щільності, спектральної кластеризації, спектрального кольорування тощо.

Опишемо один з можливих алгоритмів послідовного типу, який приводить до отримання локального максимуму у відповідності з критерієм для випадку, коли існують деякі обмеження. Часто при компонуванні модулів в комірці, яка є прототипом задачі розрізу графу на куски, ставиться вимога отримання рівних за кількісно вершин кусків. Крім того, як правило, деякі вершини графу жорстко закріплюються за визначеними кусками, тобто є забороненими для перестановок.

Нехай задано граф $G(X, LL)$ та, множина заборонених елементів $Q \in X, |Q| = q$. Потрібно знайти такий розріз $B(G_i)$, на l однакових кусків, щоб

$$(\forall x, y \in Q)[x \in X_i \Rightarrow y \notin X_i V y \in X_i \Rightarrow x \notin X_i], \qquad (2.1)$$

де X_i- вершини i -го куска.

Інакше передбачається, що будь-які дві заборонені вершини розміщуються в різних шматках. Ця вимога не порушує загальності міркувань.

Побудова першого шматка починається з вершини $x_\xi \in Q$ яку апріорно вважаємо належною множині вершин X_1 першого шматка $G_1 = (X_1, U_1)$ та такою що створює перший рівень. На першому рівні множина $X_1 = \{x_\xi\}$.

Для визначення вершини наступного рівня, тобто другої вершини, яку необхідно розмістити в кусок G_1 будується множина вершин, суміжних з

вершиною $x_\xi (\xi \in E = \{1,2,\dots,q\})$. Позначимо цю множину Γx_ξ.

Введемо поняття відносної ваги $\sigma(x_i)$ для будь-якої вершини графу x_i :

$$\sigma(x_i) = \rho(x_i) - a_{ik} \tag{2.2}$$

Де a_{ik} - кількість ребер, що з'єднують вершину x_i з вершинами X_1.У відповідності з обраним критерієм для отримання необхідного шматка з множини Γx_ξ необхідно вибрати вершину x_i з мінімальною величиною $\sigma(x_i)$ тобто вибрати x_i для якої:

$$\sigma(x_i) = min\, \sigma\,(x_i) \tag{2.3}$$

де $i = \{1,2,\dots,t\}, t = \left|\Gamma x_\xi\right|$

Вершина x_i є вершиною другого рівня. На другому рівні

$$X_1 = \{x_\xi, x_i\}. \tag{2.4}$$

Далі розглянемо множину $\Gamma x_\xi \cup \Gamma x_i$ та для кожної вершини $x_k \in (\Gamma x_\xi \cup \Gamma x_i)$ визначимо відносну вагу у відповідності з виразом. Вибираючи вершину x_k з мінімальною вагою, отримаємо: $X_1 = \{x_\xi, x_i, x_k\}$.

Вказаний процес виконується до тих пір, поки множина X_1не буде містити n/l елементів. Отриманий кусок G_1 видаляється з графу G . Вибирається наступна заборонена вершина $x_\gamma \in Q$ та будується наступний кусок за описаною процедурою.

Сформулюємо описану процедуру у вигляді алгоритму.

1. Видаляємо першу заборонену вершину x_ξта за матрицею суміжності або її кодовою реалізацією будуємо множину Γx_ξ .
2. Для вершин з множини Γx_ξ визначаємо за формулою відносні ваги та вибираємо з них мінімальну.
3. З підмножини вершин з рівною відносною вагою вибираємо вершину з більшим локальним ступенем, розміщуємо її в шматок.
4. Підраховуємо кількість Sвершин в шматку. При $S < R$ переходимо до п.7.; якщо $S = R$ то кусок отримано. Видаляємо його з графу G.
5. Перевіряємо існування заборонених вершин.
6. Будуємо множину вершин, суміжних раніш вибраним в даний шматок.

2.2 Моделі визначення зв’язних підграфів

Моделі визначення зв’язних підграфів - це методи, які дозволяють виявити та виділити частини графа, які мають високий рівень зв’язності між вершинами. Зв’язність графа означає, що між будь-якою парою вершин існує шлях, який поєднує їх. Зв’язні підграфи можуть мати різне значення та застосування в різних областях, таких як соціальні мережі, транспортні мережі, теорія кодування тощо.

Зв’язні підграфи - це підграфи, в яких між будь-якою парою вершин існує шлях, що з’єднує їх (рис 2.1). Моделі визначення зв’язних підграфів - це методи та алгоритми, що дозволяють знаходити, виділяти, класифікувати та аналізувати зв’язні підграфи в заданому графі. Зв’язні підграфи можуть мати різну структуру, розмір, кількість та властивості, залежно від типу графа та цілей дослідження.

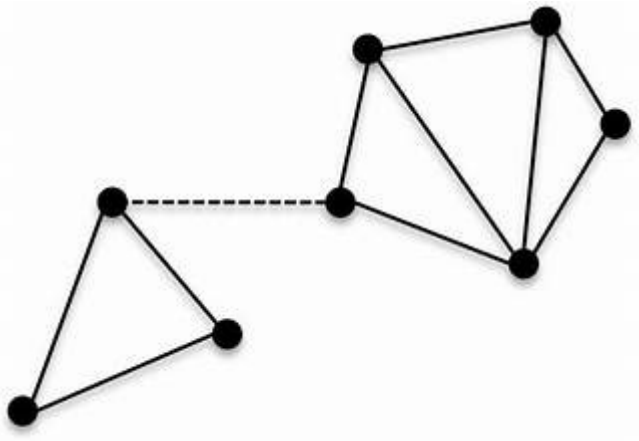


Рисунок 2.1 – Зв’язний граф

Ось деякі приклади моделей визначення зв’язних підграфів:

Модель компонент зв’язності (connected components) полягає в розбитті графа на максимальні зв’язні підграфи, так що між будь-якими двома вершинами кожного підграфа існує шлях, а між вершинами різних підграфів - немає. Ця модель застосовується для виявлення структурних патернів у графах, таких як кластери, спільноти, сегменти тощо. Ця модель базується на ідеї, що

зв’язний граф може бути розбитий на декілька неперетинаючих частин, які називаються компонентами зв’язності. Компонента зв’язності - це максимальний зв’язний підграф, який не може бути розширений додаванням інших вершин або ребер. Для неорієнтованих графів компоненти зв’язності можна знайти за допомогою алгоритму пошуку в глибину або пошуку в ширину. Для орієнтованих графів існують різні види компонент зв’язності, такі як сильно-зв’язні, односторонньо-зв’язні та слабко-зв’язні компоненти.

Модель кластеризації: ця модель базується на ідеї, що зв’язний граф може бути розбитий на декілька груп вершин, які мають схожість або близькість за певними критеріями. Кластеризація - це процес розподілу вершин на кластери або суграфи, так що вершини в одному кластері мають багато внутрішніх зв’язків і мало зовнішніх зв’язків. Для кластеризації графів можна використовувати різні методи, такі як модулярність, спектральна кластеризація, стохастична блокова модель тощо.

Модель спануючих дерев: ця модель базується на ідеї, що зв’язний граф може бути розбитий на декілька підграфів, які є деревами, тобто не містять циклів. Спануюче дерево - це підграф, який містить всі вершини початкового графа, але має мінімальну кількість ребер. Спануюче дерево можна знайти для будь-якого зв’язного неорієнтованого графа. Спануючих дерев може бути багато для одного і того ж графа. Для знаходження спануючих дерев можна використовувати різні алгоритми, такі як алгоритм Прима, алгоритм Краскала, алгоритм Борувки тощо.

Модель блокового графа (block graph) полягає в розбитті графа на двозв’язні компоненти (блоки), які є кліками (повними підграфами), і таких, що будь-яке ребро належить рівно одному блоку [3]. Ця модель застосовується для характеризації жорсткості та стабільності графа.

Модель ожин (biconnected components) полягає в розбитті графа на суграфи, яким можна приписати дерево ожин (biconnected component tree), таке що кожна вершина дерева - це ожина (biconnected component), а кожне ребро дерева - це точка розриву (cut vertex), яка належить двом ожинам. Ця модель

застосовується для аналізу структурної складності та надмірності графа.

Існують ще інші моделі визначення зв’язних підграфів, такі як:

Модель максимальних зв’язних підграфів (maximal connected subgraphs) полягає в знаходженні всіх таких зв’язних підграфів графа, які не є підграфами жодного іншого зв’язного підграфа. Ця модель застосовується для виявлення найбільших зв’язних компонент графа [5].

Модель максимальної зв’язності (maximum connectivity) полягає в знаходженні такого зв’язного підграфа графа, який має найбільшу кількість ребер. Ця модель застосовується для оптимізації розподілу ресурсів та навантаження в мережах.

Модель к-зв’язності (k-connectivity) полягає в перевірці, чи є граф к-зв’язним, тобто чи можна вилучити менше ніж k вершин або ребер, щоб граф став незв’язним. Ця модель застосовується для оцінки надійності та стійкості графа до порушень.

2.3 Критерії визначення спільнот в сильнозв’язних компонентах графа

Граф - це математична структура, яка складається з множини вершин і множини ребер, які з’єднують пари вершин. Граф може бути орієнтованим або неорієнтованим, залежно від того, чи мають ребра напрямок. Граф може моделювати різні види мереж, таких як соціальні, біологічні, транспортні тощо.

Сильнозв’язна компонента графа - це підграф, в якому існує шлях з будь-якої вершини до кожної з інших вершин. Це означає, що вершини в сильнозв’язній компоненті мають високий рівень взаємодії і співпраці між собою. Сильнозв’язна компонента може містити одну або більше вершин, а також один або більше орієнтованих циклів. Якщо кожну сильнозв’язну компоненту стягнути до однієї вершини, отримаємо орієнтований ациклічний граф, ущільнення графа.

Спільнота - це група вершин, які мають багато зв’язків всередині групи і

мало зв’язків з іншими групами. Спільноти можуть бути визначені за допомогою різних критеріїв, таких як модулярність, що вимірює розбиття графа на групи таким чином, щоб максимізувати кількість ребер всередині груп і мінімізувати кількість ребер між групами. Існують різні алгоритми для пошуку спільнот у графах, такі як метод Лувена, метод Гірвана-Ньюмана, метод Клейна-Берга та інші.

Визначення спільнот у сильно зв’язних компонентах графа - це складна задача, яка потребує аналізу структури і динаміки графа. Один з можливих підходів - це застосування методу модулярностей до ущільнення графа, щоб отримати розбиття на спільноти на рівні сильно зв’язних компонент. Інший підхід - це застосування методу Лувена до оригінального графа, а потім об’єднуючи вершини, які належать до одної сильно зв’язної компоненти в одне ребро. Обидва підходи мають свої переваги і недоліки, і вибір найкращого залежить від мети дослідження і властивостей графа.

Критерії визначення спільнот у сильнозв’язних компонентах графа можуть бути різними, залежно від того, що ми хочемо виявити про структуру і динаміку графа. Наприклад, ми можемо зацікавлені в тому, як сильнозв’язні компоненти формуються і розвиваються в часі, як вони впливають на поведінку вершин і ребер, як вони корелюють з іншими характеристиками графа, такими як ступінь, кластеризація, центральність тощо. Для кожного з цих аспектів ми можемо мати різні метрики і методи для визначення спільнот у сильнозв’язних компонентах графа.

Один з можливих критеріїв - це сила зв’язку між вершинами в сильно зв’язних компонентах. Сила зв’язку - це міра того, наскільки часто вершини спілкуються або обмінюються інформацією. Сила зв’язку може бути оцінена за допомогою різних параметрів, таких як частота, тривалість, інтенсивність або якість комунікації. Вершини з високою силою зв’язку можуть утворювати спільноту, яка характеризується стабільністю і надійністю. Вершини з низькою силою зв’язку можуть утворювати спільноту, яка характеризується гнучкістю і адаптацією.

Інший можливий критерій - це схожість між вершинами в сильно зв’язних компонентах. Схожість - це міра того, наскільки вершини мають спільні атрибути або інтереси. Схожість може бути оцінена за допомогою різних параметрів, таких як загальна кількість спільних сусідів, загальна кількість спільних ребер, загальна кількість спільних характеристик тощо

Сильнозв’язна компонента графа - це підграф, в якому існує шлях з будь-якої вершини до кожної з інших вершин. Це означає, що вершини в сильнозв’язній компоненті мають високий рівень взаємодії і співпраці між собою. Сильнозв’язна компонента може містити одну або більше вершин, а також один або більше орієнтованих циклів. Якщо кожну сильнозв’язну компоненту стягнути до однієї вершини, отримаємо орієнтований ациклічний граф, ущільнення графа.

Спільнота - це група вершин, які мають багато зв’язків всередині групи і мало зв’язків з іншими групами. Спільноти можуть бути визначені за допомогою різних критеріїв, таких як модулярність, що вимірює розбиття графа на групи таким чином, щоб максимізувати кількість ребер всередині груп і мінімізувати кількість ребер між групами. Існують різні алгоритми для пошуку спільнот у графах, такі як метод Лувена, метод Гирвана-Ньюмана, метод Клейна-Берга та інші.

Визначення спільнот у сильнозв’язних компонентах графа - це складна задача, яка потребує аналізу структури і динаміки графа. Один з можливих підходів - це застосування методу модулярностей до ущільнення графа, щоб отримати розбиття на спільноти на рівні сильно зв’язних компонент. Інший підхід - це застосування методу Лувена до оригінального графа, а потім об’єднуючи вершини, які належать до одної сильно зв’язної компоненти в одне ребро. Обидва підходи мають свої переваги і недоліки, і вибір найкращого залежить від мети дослідження і властивостей графа.

Спільноти в сильнозв’язних компонентах графа - це підмножини вершин, які мають високу щільність зв’язків всередині та низьку щільність зв’язків з іншими підмножинами. Визначення спільнот в сильнозв’язних компонентах

графа - це завдання, яке полягає в розбитті сильнозв’язної компоненти на такі підмножини, що максимізують певний критерій якості.

Існують різні критерії визначення спільнот в сильнозв’язних компонентах графа, такі як:

- Модулярність (modularity) - це функція, яка вимірює різницю між кількістю ребер в підмножинах та очікуваною кількістю ребер в випадкових графах з такою ж ступеневою послідовністю. Максимізація модулярності дозволяє видобути спільноти, які мають багато внутрішніх зв’язків та мало зовнішніх.
- Кондуктанс (conductance) - це функція, яка вимірює відношення кількості ребер, що перетинають певну підмножину вершин, до суми ступеней вершин у цій підмножині. Мінімум кондуктансу дозволяє видобути спільноти, яким потребується найменше ребер для їх роз’єднання.
- Нормалований зріз (normalized cut) - це функція, яка вимірює відношення кількості ребер, що перетинають певну підмножину вершин, до суми ступенів вершин у цій підмножині та у доповнення до неї. Мінімум нормалованого зрізу дозволяє видобути спільноти, які мають балансоване розподілення ступенів вершин.

Критерій модулярності (modularity) полягає в максимізації функції модулярності, яка вимірює різницю між кількістю ребер в спільнотах та очікуваною кількістю ребер в випадкових графах з такою ж ступеневою послідовністю. Цей критерій застосовується для виявлення структурних патернів у графах, таких як кластери, спільноти, сегменти тощо.

- Максимальний потік між двома вершинами графа - це найбільша кількість одиниць даних, які можна передати від однієї вершини до іншої за допомогою ребер, що мають певну пропускну здатність. Цей критерій застосовується для оптимізації розподілу ресурсів та навантаження в мережах.

- Критерій максимальної кластеризації (maximum clustering) полягає в знаходженні такого розбиття графа на попарно неперетинаючихся підграфи,

Методи визначення та аналізу спільнот в соціальних мережах наведені у таблиці 1.

Таблиця 1 - Методи визначення та аналізу спільнот в соціальних мережах

Методи визначення та аналізу спільнот в соціальних мережах		
Підхід до виявлення спільнот	Математичні моделі для виявлення	Способи розрахунків показників виявлення спільнот
1. Кластеризація на основі графів. Підхід ґрунтується на розділенні графа на окремі компоненти зв'язності. Розрізання графа на компоненти зв'язності дозволяє визначити ізольовані групи вершин, які можуть утворювати потенційні спільноти в мережі.	Математична модель - виділення субграфів (кластерів) у великому графі. Це досягається шляхом виявлення розрізу в графі, який розділяє вершини на різні групи. Ці субграфи аналізуються на предмет їх структури та взаємозв'язків між вершинами.	1.Теорія множин для визначення зв'язків між вершинами у графі та їх групування. Наприклад, визначення групи вершин, які мають спільних сусідів. 2.Порівняння внутрішніх зв'язків між вершинами групи з їх зовнішніми зв'язками з рештою графа. Спільнота вважається, якщо кількість зв'язків між вершинами у групі більше, ніж кількість зв'язків цих вершин з іншими вершинами в графі. 3.Коефіцієнт кластеризації це середня ступінь кластеризації для всіх вершин у графі. Вимірює, як часто вершини утворюють з'єднання з іншими вершинами в своєму ж

		графічному околі.
2. Метод k-середніх. Розділяє вершини графа на k кластерів, де кожен кластер представлений центроїдом. Центроїд є "середнім" представником вершин у кластері та визначається як середнє арифметичне координат вершин у кластері.	Математична модель – мінімізація внутрішньокластерної дисперсії. Для кожного кластера обчислюється сума відстаней між кожною вершиною та центроїдом кластера, і ця сума мінімізується.	1.Для кожної вершини обчислюється її відстань до центроїду кожного кластера. Кожна вершина призначається до кластера з найменшою відстанню. Метод базується на кластеризації вершин у k кластерів, критерій визначення спільноти може бути виражений через принципи кластеризації. 2.Кластеризація k-середніх використовує міру інерції (inertia) або внутрішньокластерний показник розкиду. Це сума квадратів відстаней між кожною точкою та центроїдом її кластера. Кластеризація намагається мінімізувати це значення.
3. Ієрархічна кластеризація. Створює ієрархію кластерів у вигляді дерева. Кожен вузол дерева представляє кластер, а злиття або розбиття кластерів відбувається на різних рівнях ієрархії. Цей метод може бути агломеративним (починає з окремих вершин і об'єднує їх) або дізгломеративним (починає з усіх вершин і розділяє їх).	Математичні моделі для обчислення подібності між вершинами та групами вершин на різних рівнях ієрархії. Різноманітні метрики подібності, такі як відстань між кластерами чи кореляція між їх характеристиками.	1.Методи об'єднання або розбиття, які базуються на метриках відстаней або подібності між вершинами або групами вершин. Різні метрики для визначення спільнот. Одним із можливих критеріїв це зростання внутрішньокластерних зв'язків порівняно зі зв'язками між кластерами.

		2. Коефіцієнт кластеризації в ієрархічній кластеризації це середнє значення міри кластеризації для різних рівнів ієрархії, якщо існують різні рівні об'єднання кластерів.

У сильній спільноті кожна вершина має більше зв'язків всередині спільноти, ніж з рештою графа.

У слабкій спільноті сума всіх ступенів всередині V більша ніж сума всіх ступенів у напрямку до решти мережі.

2.4 Обчислення кластерного коефіцієнту для спільнот

Кластерний коефіцієнт для спільнот — це міра, яка вказує, наскільки щільно пов’язані елементи в кожному кластері. Кластер — це група об’єктів, які мають схожі характеристики. Це допомагає оцінити якість кластеризації, тобто розбиття множини даних на групи, що мають схожі характеристики. Існує декілька способів обчислення кластерного коефіцієнту для спільнот, але один з найпоширеніших — це середнє геометричне кількості трикутників та зірок у кожному кластері. Трикутник — це трійка елементів, які всі між собою пов’язані, а зірка — це чотирьохелементна структура, де один елемент пов’язаний з трьома іншими, але вони не пов’язані між собою. Чим більше трикутників та зірок у кластері, тим більше він є когезивним, тобто складається з елементів, що мають сильні зв’язки.

Ієрархічна кластеризація — це метод, який будує дерево або ієрархію кластерів, починаючи від окремих об’єктів і поступово об’єднуючи їх у більші групи. Існують два основних типи ієрархічної кластеризації: агломеративна і

дивізивна. Агломеративна кластеризація починає з того, що кожен об’єкт утворює окремий кластер, а потім на кожному кроці об’єднує найближчі кластери, поки не залишиться один кластер, що містить всі об’єкти. Дивізивна кластеризація починає з того, що всі об’єкти належать до одного кластера, а потім на кожному кроці розбиває найбільший кластер на два менших, поки кожен об’єкт не стане окремим кластером. Ієрархічна кластеризація має перевагу в тому, що не потребує заздалегідь задавати кількість кластерів, а також дозволяє візуалізувати структуру даних за допомогою дендрограми. Однак, вона має недоліки в тому, що є обчислювально складною, особливо для великих наборів даних, і не може коригувати свої рішення на попередніх кроках. Ієрархічна кластеризація може бути застосована до графів, якщо визначити метрику відстані між кластерами графів, наприклад, за допомогою матриці суміжності або спектральної теорії графів.

Кластеризація OPTICS (Ordering Points To Identify the Clustering Structure) — це метод, який виявляє кластери з різною щільністю і формою, а також виділяє шумові об’єкти. OPTICS схожий на метод DBSCAN, але не потребує задавати параметр радіусу околу для визначення щільності, а замість цього використовує порядок обходу об’єктів, який відображає їхню щільність. OPTICS створює так званий діаграму досяжності, яка показує, наскільки далеко потрібно перейти від одного об’єкта до іншого, щоб вони належали до одного кластера. На цій діаграмі кластери відображаються як довгі ділянки з низьким значенням досяжності, а шумові об’єкти — як піки з високим значенням досяжності. OPTICS має перевагу в тому, що може пристосовуватися до різних рівнів щільності кластерів, а також виявляти кластери складної форми. Однак, він має недоліки в тому, що є обчислювально складним, особливо для великих наборів даних, і не дає явного розбиття на кластери, а тільки порядок об’єктів. Кластеризація OPTICS може бути застосована до графів, якщо визначити метрику відстані між вершинами графів, наприклад, за допомогою найкоротшого шляху або геодезичної відстані.

OPTICS (Ordering Points To Identify the Clustering Structure) - це алгоритм,

який знаходить кластери у просторових даних на основі щільності. Він схожий на алгоритм DBSCAN, але не потребує задавати параметр eps, який визначає радіус околу для кожної точки. Замість цього, OPTICS використовує параметр minPts, який визначає мінімальну кількість сусідів для визначення ядрової точки, і порядок досяжності, який вимірює відстань до найближчої ядрової точки. OPTICS сортує точки за зростанням порядку досяжності і виводить дендрограму, яка показує структуру кластерів на різних рівнях щільності.

Приклад обчислення за методом OPTICS:

1. Нехай ми маємо деякий набір даних з 10 точками, які розташовані у двовимірному просторі. Ми хочемо знайти кластери за допомогою OPTICS з параметром minPts = 3.
2. Ми обираємо довільну точку, наприклад, A, і обчислюємо відстань до всіх інших точок. Ми визначаємо, що точка A має 4 сусіди на відстані менше або рівній 2, тому вона є ядровою точкою. Ми встановлюємо порядок досяжності точки A як нескінченність, тому що вона є першою точкою в порядку. Ми додаємо точку A до списку відвіданих точок і до списку порядку.
3. Ми обираємо найближчу ядрову точку до точки A, наприклад, B, і обчислюємо відстань до всіх інших точок. Ми визначаємо, що точка B має 3 сусіди на відстані менше або рівній 2, тому вона також є ядровою точкою. Ми встановлюємо порядок досяжності точки B як відстань до точки A, тобто 2. Ми додаємо точку B до списку відвіданих точок і до списку порядку.
4. Ми обираємо найближчу ядрову точку до точки B, наприклад, C, і обчислюємо відстань до всіх інших точок. Ми визначаємо, що точка C має 3 сусіди на відстані менше або рівній 2, тому вона також є ядровою точкою. Ми встановлюємо порядок досяжності точки C як відстань до точки B, тобто 1. Ми додаємо точку C до списку відвіданих точок і до списку порядку.
5. Ми обираємо найближчу ядрову точку до точки C, наприклад, D, і

обчислюємо відстань до всіх інших точок. Ми визначаємо, що точка D має 3 сусіди на відстані менше або рівній 2, тому вона також є ядровою точкою. Ми встановлюємо порядок досяжності точки D як відстань до точки C, тобто 1. Ми додаємо точку D до списку відвіданих точок і до списку порядку.

6. Ми обираємо найближчу ядрову точку до точки D, наприклад, E, і обчислюємо відстань до всіх інших точок. Ми визначаємо, що точка E має 2 сусіди на відстані менше або рівній 2, тому вона не є ядровою точкою. Ми встановлюємо порядок досяжності точки E як відстань до точки D, тобто 1. Ми додаємо точку E до списку відвіданих точок і до списку порядку.
7. Ми повертаємося до точки D і шукаємо іншу ядрову точку, яка ще не була відвідана, наприклад, F, і обчислюємо відстань до всіх інших точок. Ми визначаємо, що точка F має 3 сусіди на відстані менше або рівній 2, тому вона також є ядровою точкою. Ми встановлюємо порядок досяжності точки F як відстань до точки D, тобто 1. Ми додаємо точку F до списку відвіданих точок і до списку порядку.
8. Ми обираємо найближчу ядрову точку до точки F, наприклад, G, і обчислюємо відстань до всіх інших точок. Ми визначаємо, що точка G має 3 сусіди на відстані менше або рівній 2, тому вона також є ядровою точкою. Ми встановлюємо порядок досяжності точки G як відстань до точки F, тобто 1. Ми додаємо точку G до списку відвіданих точок і до списку порядку.
9. Ми обираємо найближчу ядрову точку до точки G, наприклад, H, і обчислюємо відстань до всіх інших точок. Ми визначаємо, що точка H має 2 сусіди на відстані менше або рівній 2, тому вона не є ядровою точкою. Ми встановлюємо порядок досяжності точки H як ві Ми повертаємося до точки G і шукаємо іншу ядрову точку, яка ще не була відвідана, наприклад, I, і обчислюємо відстань до всіх інших точок. Ми визначаємо, що точка I має 3 сусіди на відстані менше або рівній 2, тому

вона також є ядровою точкою. Ми встановлюємо порядок досяжності точки I як відстань до точки G, тобто 1. Ми додаємо точку I до списку відвіданих точок і до списку порядку.

10. Ми обираємо найближчу ядрову точку до точки I, наприклад, J, і обчислюємо відстань до всіх інших точок. Ми визначаємо, що точка J має 2 сусіди на відстані менше або рівній 2, тому вона не є ядровою точкою. Ми встановлюємо порядок досяжності точки J як відстань до точки I, тобто 1. Ми додаємо точку J до списку відвіданих точок і до списку порядку.
11. Ми повертаємося до точки I і шукаємо іншу ядрову точку, яка ще не була відвідана, але не знаходимо жодної. Ми повертаємося до точки G і шукаємо іншу ядрову точку, яка ще не була відвідана, але не знаходимо жодної. Ми повертаємося до точки F і шукаємо іншу ядрову точку, яка ще не була відвідана, але не знаходимо жодної. Ми повторюємо цей процес, поки не повернемося до точки A, яка є першою точкою в порядку.
12. Ми завершуємо алгоритм і отримуємо список порядку, який виглядає так: A, B, C, D, E, F, G, H, I, J. Ми також отримуємо значення порядку досяжності для кожної точки, які виглядають так: ∞, 2, 1, 1, 1, 1, 1, 1, 1, 1.
13. Ми виводимо дендрограму, яка показує структуру кластерів на різних рівнях щільності (рис 2.2).
14. Як видно, дендрограма має дві гілки, які відповідають двом кластерам: {A, B, C, D, E} і {F, G, H, I, J}. Якщо ми виберемо деякий поріг для порядку досяжності, наприклад, 1.5, то ми отримаємо ці два кластери як результат кластеризації. Якщо ми виберемо інший поріг, наприклад, 2.5, то ми отримаємо три кластери: {A, B}, {C, D, E} і {F, G, H, I, J}. Якщо ми виберемо ще інший поріг, наприклад, 3.5, то ми отримаємо десять кластерів: {A}, {B}, {C}, {D}, {E}, {F}, {G}, {H}, {I}, {J}. Таким чином, метод OPTICS дозволяє вибрати рівень деталізації кластеризації в залежності від відстані до точки G, тобто 1.

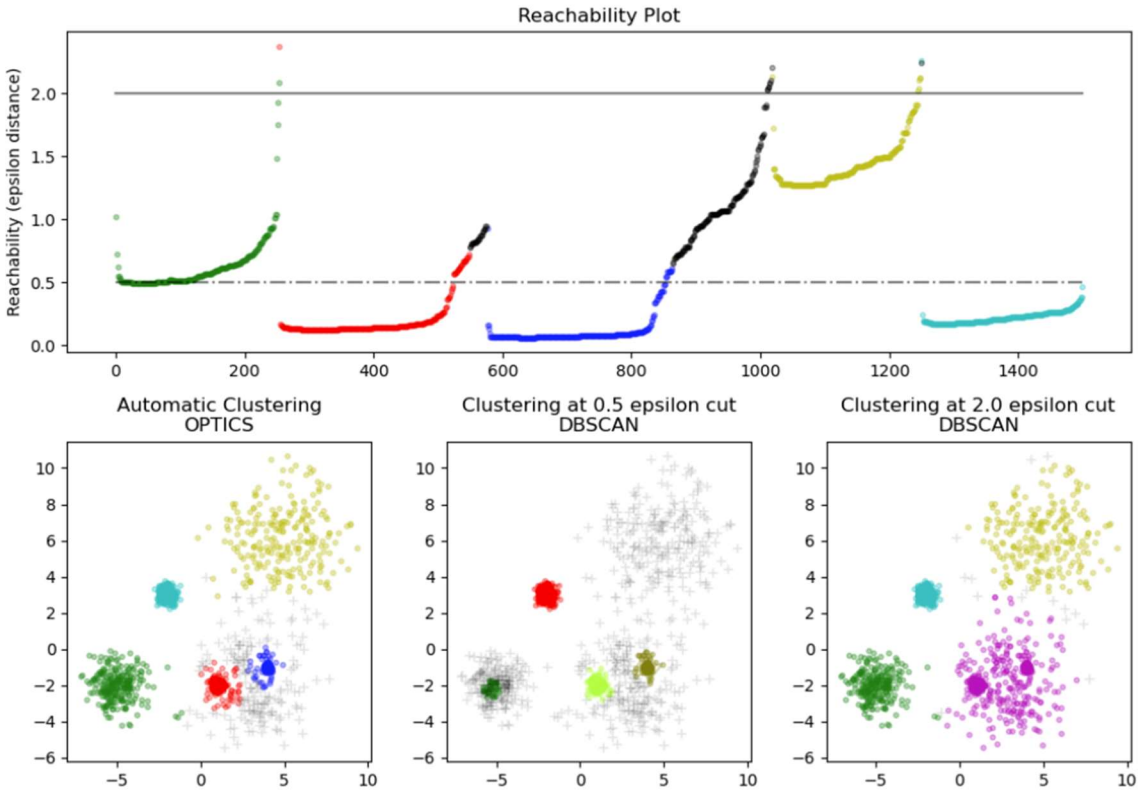


Рисунок 2.2 – Приклад алгоритму OPTICS

Нейромережа Кохонена, або самоорганізаційна карта — це метод, який використовує штучну нейронну мережу для відображення багатовимірних даних на двовимірну або одновимірну сітку. Нейромережа Кохонена складається з двох шарів: вхідного і вихідного. Вхідний шар містить нейрони, які отримують значення ознак об’єктів, а вихідний шар містить нейрони, які представляють кластери. Нейрони вихідного шару з’єднані між собою сусідніми відносинами, які визначають топологію сітки. Навчання нейромережі Кохонена полягає в тому, що для кожного об’єкта знаходиться найближчий нейрон вихідного шару, який називається переможцем, і коригуються ваги його та його сусідів, щоб зменшити відстань між ними і об’єктом. Таким чином, нейромережа Кохонена адаптується до структури даних і формує кластери, які відображають схожість і сусідство об’єкта. Нейромережа Кохонена має перевагу в тому, що може здійснювати кластеризацію без заздалегідь заданої кількості кластерів, а також відображати багатовимірні дані на низьковимірну сітку, що сприяє візуалізації та інтерпретації. Однак, вона має недоліки в тому,

що є чутливою до вибору параметрів навчання, таких як швидкість навчання, розмір сусідства, функція активації тощо, і не гарантує стабільності та оптимальності отриманих кластерів. Нейромережа Кохонена може бути застосована до графів, якщо визначити метрику відстані між вершинами графів, наприклад, за допомогою косинусної подібності або евклідової відстані.

Алгоритм K-Means групує дані, намагаючись розділити вибірки на n груп рівної дисперсії, мінімізуючи критерій, відомий як інерція або сума квадратів у межах кластера. Цей алгоритм вимагає вказати кількість кластерів.

Алгоритм k-середніх розділяє набір з N вибірок X на K непересічних кластерів, кожен з яких описується середнім значенням μ зразків у кластері. Засоби зазвичай називають «центроїдами» кластера; зауважте, що вони взагалі не є точками від X, хоча вони живуть в одному просторі.

K-means часто називають алгоритмом Лойда. В основному алгоритм складається з трьох кроків. На першому кроці вибираються початкові центроїди, а найпростішим методом є вибір зразків із набору k даних X. Після ініціалізації K-засоби складаються з циклу між двома іншими кроками. Перший крок призначає кожен зразок до найближчого центроїда. Другий крок створює нові центроїди, беручи середнє значення всіх зразків, призначених кожному попередньому центроїду. Обчислюється різниця між старим і новим центроїдами, і алгоритм повторює ці останні два кроки, поки це значення не стане меншим за порогове значення. Іншими словами, він повторюється до тих пір, поки центроїди не зрушяться суттєво.

Кластерний коефіцієнт для спільнот може приймати значення від 0 до 1. Чим ближче він до 1, тим краще кластеризація, тобто тим більше елементи кожного кластера схожі між собою. Чим ближче він до 0, тим гірше кластеризація, тобто тим менше елементи кожного кластера пов’язані між собою.

Щоб покращити кластерний коефіцієнт для спільнот, можна спробувати використати інший метод кластеризації, який краще підходить для даної множини даних, або змінити кількість кластерів, які хочете отримати. Також

можна використати іншу міру для обчислення кластерного коефіцієнту для спільнот, яка враховує не тільки кількість трикутників та зірок, але й інші аспекти структури кластерів, наприклад, їхню розмірність, густина, модулярність тощо. Ось деякі приклади таких мір:

Коефіцієнт силового співтовариства (Strength of Community Coefficient) — це міра, яка враховує не тільки кількість трикутників та зірок, але й кількість ребер, що виходять з кожного кластера. Ця міра показує, наскільки кластери ізольовані від інших кластерів, тобто наскільки вони є автономними спільнотами.

Коефіцієнт модулярності (Modularity Coefficient) — це міра, яка враховує не тільки кількість трикутників та зірок, але й відхилення кількості ребер в кожному кластері від очікуваної кількості ребер, якщо б елементи були розподілені випадково. Ця міра показує, наскільки кластери відрізняються від випадкового розбиття, тобто наскільки вони є структурованими спільнотам.

Тепер спробуємо застосувати аналіз соціальних мереж на практиці. Для цього можна використати мову програмування Python, а точніше бібліотеку networkx, призначену для роботи з графами, бібліотеку matplotlib для візуалізації та бібліотеку community для виділення спільнот усередині мережі. Давайте їх імпортуємо:

Як датасет візьмемо листування за допомогою електронної пошти від великого європейського університету, де міститься анонімна інформація про всі вхідні та вихідні електронні повідомлення між членами дослідницької установи (посилання). Датасет містить файл формату txt, де в кожному рядку перелічено пари вузлів, які пов'язані один з одним.

Кількість вершин: 1005

Кількість ребер: 25571

Середня кількість сусідів у вузлів у графі: 25.443

Датасет було імпортовано і перетворено на граф. Потім ми послідовно вивели основні параметри графа: кількість вузлів, ребер і середню кількість сусідів у вузлів у графі. Останній параметр відображає, наскільки тісно

пов'язані вузли між собою.

Для того щоб зрозуміти, як можна працювати з кожним конкретним графом, спочатку потрібно зрозуміти, як він влаштований. Давайте коротко розглянемо характеристики, за допомогою яких можна зрозуміти структуру графа.

Насамперед розглянемо поняття зв'язності та спрямованості. Граф називається зв'язним, якщо кожна пара вузлів графа пов'язана між собою, тобто з будь-якого вузла можна прийти в будь-який інший вузол. Якщо ж граф незв'язний, то його можна розбити на максимально зв'язні підграфи (звані компонентами). Також графи можуть бути спрямованими і ненаправленими. Це визначається наявністю спрямованості зв'язків між двома учасниками. Одним із прикладів спрямованої мережі є транзакції між клієнтами банку, де в кожного клієнта можуть бути як вхідні, так і вихідні платежі.

У загальному випадку в спрямованих графах зв'язки не взаємні, тому для спрямованих графів замість поняття зв'язності виділяють поняття компоненти слабкої та сильної зв'язності. Компонента вважається слабко зв'язною, якщо при ігноруванні напрямку виходить зв'язний граф. Компонента сильної зв'язності може бути такою, якщо всі її вершини взаємно досяжні. Давайте подивимося, яку структуру має граф із нашого датасету з електронним листуванням:

Граф є спрямованим і складається з кількох компонент.
Тут ми провели перевірку на спрямованість і зв'язність графа і з'ясували, що граф із датасету є спрямованим і містить кілька незв'язних компонент. Давайте спробуємо подивитися ближче на найбільші компоненти сильної та слабкої зв'язності:

Кількість вершин: 986

Кількість ребер: 25552

Середня кількість сусідів у вузла в графі: 25.9148

Кількість вершин: 803

Кількість ребер: 24729

Середня кількість сусідів у вузла в графі: 30.7958

Отже, ми отримали основні характеристики для компоненти слабкої зв'язності та для компоненти сильної зв'язності, що входить до неї. Давайте подивимося, які висновки ми можемо зробити на цьому етапі. Ми бачимо, що зі 1005 учасників один з одним спілкуються 986 осіб, при цьому 183 людини з них надсилали електронні листи іншим людям в односторонньому порядку, і тільки 803 людини підтримували двостороннє спілкування. У 823 випадках спроба налагодити комунікацію за допомогою електронної пошти виявилася провальною. Також ми бачимо, що найактивніші учасники (ті, що входять до компоненти сильної зв'язності) підтримують комунікацію в середньому з 30 людьми.

Давайте розглянемо й інші ключові характеристики графів, що визначають їхню структуру. Графи вважаються зваженими, якщо відносини між вузлами відображають не тільки саму наявність зв'язку, а й певну вагу, що відображає силу цього зв'язку. Наш датасет з електронними повідомленнями зваженим не є, оскільки в ньому враховується тільки факт наявності листування, але не кількість відправлених листів.

Крім того, вузли та зв'язки можуть створювати різні типи мереж: однодольні, дводольні або багаторівневі. Однодольні мережі складаються з одного типу учасників і зв'язків між ними. Дводольні мережі складаються з двох різних типів учасників, де один із типів вузлів пов'язаний тільки з іншим типом. Багаторівневі мережі також включають два типи учасників, проте зв'язки можуть з'єднувати як учасників різних типів, так і однотипних учасників (наприклад, стосунки між менеджерами і стосунки між проектами). Узятий нами для дослідження датасет являє собою мережу однодольного типу, оскільки складається тільки з одного типу учасників і зв'язків між ними.

А тепер давайте спробуємо візуалізувати взятий нами датасет. Для цього нам знадобиться бібліотека matplotlib, вже імпортована нами вище:

У першому рядку задається розмір майбутнього зображення, якому потім присвоюється певна назва. У третьому рядку функції draw передається граф і

вказується розмір вузлів, після чого граф виводиться на екран (рис 2.3).

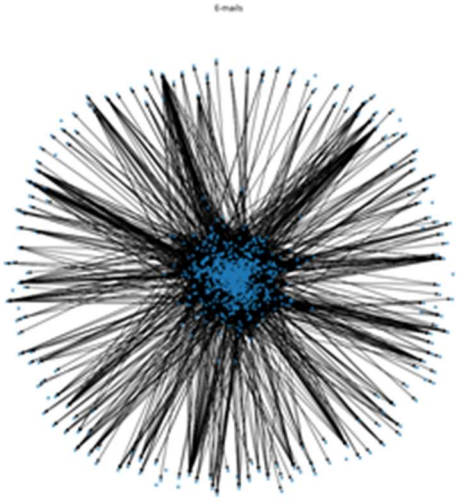


Рис. 2.3. Граф взаємодій користувачів з інформацією про всі вхідні та вихідні електронні повідомлення між членами дослідницької установи.

На отриманому графі ми бачимо, що існує низка точок, не пов'язаних з іншими учасниками комунікації. Ці люди, не пов'язані з рештою учасників, потрапили на граф, оскільки надсилали листи виключно самі собі. Також можна помітити, що на периферії розташована низка точок, які пов'язані з рештою графа нечисленними вхідними зв'язками, - це учасники, які потрапили в компоненту слабкої зв'язності для нашого графа, але не потрапили в компоненту сильної зв'язності.

Давайте також розглянемо граф, що ілюструє компоненту сильної зв'язності - людей, які підтримують двосторонню комунікацію з іншими учасниками дослідницької установи. Тепер, коли ми знаємо структуру нашого графа і вміємо його візуалізувати, давайте перейдемо до більш детального аналізу. У кожного вузла в графі є ступінь - кількість найближчих сусідів цього вузла. У спрямованих мережах можна розрізняти як ступінь входу (кількість вхідних зв'язків із вузлом), так і ступінь виходу (кількість вихідних зв'язків із вузла). Якщо для кожного вузла графа порахувати ступінь, можна визначити розподіл ступенів вузлів. Давайте подивимося на нього для графа з

електронним листуванням (рис 2.4).

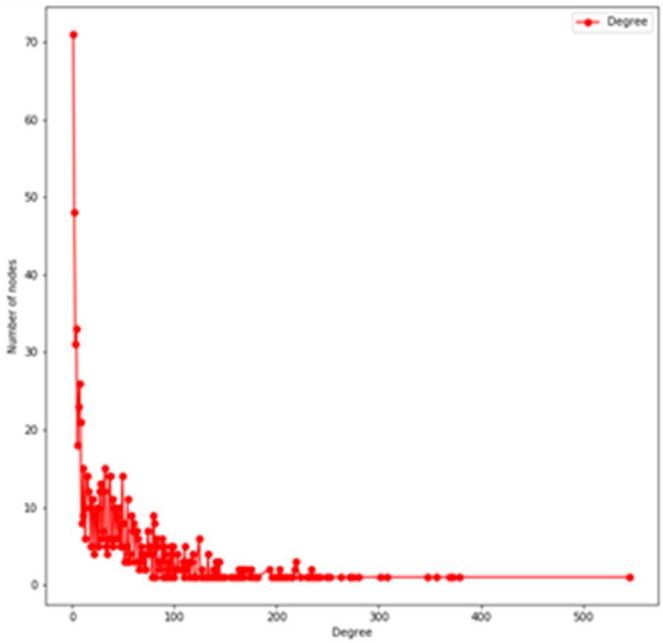


Рис. 2.4. Розподіл ступенів у графі з інформацією про всі вхідні та вихідні електронні повідомлення між членами дослідницької установи.

На отриманому графіку ми бачимо розподіл ступенів вузлів: велика кількість вузлів має нечисленні зв'язки із сусідами, але є невелика кількість великих вузлів, у яких кількість зв'язків з іншими учасниками величезна. Така тенденція називається степеневим законом розподілу, і вона дуже характерна для великих мереж. Цим законом можна описати розподіл населення різних міст, ранжування сайтів в Інтернеті і навіть розподіл матеріальних благ серед людей.

Тепер давайте визначимо те, наскільки учасники нашого графа пов'язані між собою. Для початку поговоримо про різні типи відстаней між вузлами. Будь-яка послідовність ребер, яка з'єднує вузли, називається шлях. Найчастіше в дослідженнях розглядають простий шлях, тобто шлях без циклів і вузлів, що повторюються. Найкоротший шлях між двома вузлами називають геодезичною відстанню. Найдовший найкоротший шлях у графі називають його діаметром, проте він дуже чутливий до відхилень (один ланцюжок у багатомільйонному

графі може змінити його діаметр). У спрямованих графах поняття діаметра можна застосувати тільки для компоненти сильної зв'язності, адже для підрахунку діаметра необхідно, щоб для кожної пари вузлів існував шлях між ними. У ненаправлених графах достатньо того, щоб компонента, яку розглядають, була зв'язною. Ще однією дуже інформативною характеристикою вважається середня відстань між вузлами, яку можна отримати, взявши середнє значення всіх найкоротших шляхів у графі. Середня відстань визначається структурою графа: якщо граф побудований у формі ланцюжка, вона буде великою, але чим тісніше пов'язані вузли, тим меншою стає і середня відстань. Середню відстань можна порахувати як для компоненти сильної зв'язності, так і для компоненти слабкої зв'язності:

Діаметр: 6

Середня відстань у компоненті сильної зв'язності: 2.5474824768713336

Середня відстань у компоненті слабкої зв'язності: 2.164486568301397

Давайте розберемо отримані результати. Діаметр у цьому випадку показує нам максимальну відстань між двома незнайомими людьми, і тут, як і у відомій теорії про шість рукостискань, ця відстань дорівнює 6. Середня відстань у компонентах також невелика, у середньому двом незнайомим людям достатньо 2 "рукостискань". Тут можна також побачити цікавий феномен: середня відстань у компоненті сильної зв'язності дещо нижча, ніж у компоненті слабкої зв'язності. Пояснити це можна тим, що для компоненти слабкої зв'язності не враховується спрямованість зв'язків (тільки сам факт її наявності). Через це зв'язок у слабкій компоненті з'являється там, де він був відсутній для сильної компоненти.

З відстанями між учасниками розібралися, давайте перейдемо до інших явищ, що відображають, наскільки учасники в графі пов'язані між собою: кластеризації та спільнот. Кластерний коефіцієнт показує те, що два елементи графа, пов'язані через третій елемент, з високою часткою ймовірності пов'язані й один з одним. Навіть якщо вони не пов'язані, то ймовірність того, що вони виявляться пов'язаними в майбутньому, набагато вища, ніж у двох інших

випадково взятих вузлів. Це явище, зване кластеризацією або транзитивністю, надзвичайно поширене в соціальних графах. Для графів із високим ступенем кластеризації характерна присутність значної кількості пов'язаних трійок (трьох вузлів, пов'язаних один з одним). Вони є будівельним блоком багатьох соціальних мереж, де кількість подібних трикутників дуже велика. Часто виникають навіть не трикутники, а цілі кластерні утворення, звані спільнотами, які пов'язані між собою тісніше, ніж з рештою графа.

Подивимося на кластеризацію і кластерний коефіцієнт для компоненти слабкої зв'язності:

Кластеризація: 0.2201493109315837

Кластерний коефіцієнт: 0.37270757578876434

Для компоненти сильної зв'язності ми можемо отримати ті самі характеристики, а також визначити кількість центральних вузлів (вузлів, які найсильніше пов'язані з іншими) і кількість вузлів на периферії графа:

Кластеризація: 0.2328022090200813

Кластерний коефіцієнт: 0.3905903756516427

Кількість центральних вузлів: 46

Кількість вузлів на периферії: 3

У другому випадку кластеризація та кластерний коефіцієнт збільшилися, це відображає, що в компоненті сильної зв'язності міститься більша кількість пов'язаних трійок. Давайте спробуємо разом визначити основні спільноти в компоненті слабкої зв'язності:

Кількість спільнот: 8

Кількість елементів у виділених спільнотах: 113, 114, 125, 131, 169, 188, 54, 92

Отже, у графі з електронним листуванням можна виокремити 8 спільнот, які пов'язані між собою тісніше, ніж з рештою графа. Найменша зі спільнот містить 54 учасники, а найбільша - 188 учасників. Для мереж, що містять спільноти, які перекриваються або вкладені, визначити оптимальне розбиття може виявитися складніше. Тому під час кожного запуску коду склад спільнот

може відрізнятися (рис 2.5).

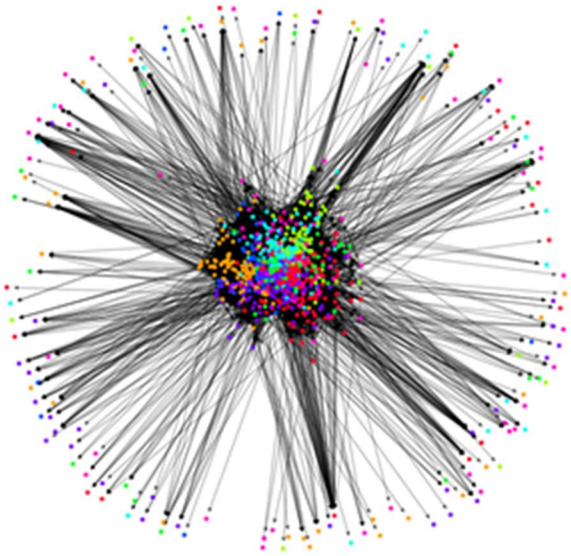


Рис. 2.5. Відображення різних спільнот у компоненті слабкої зв'язності з інформацією про всі вхідні та вихідні електронні повідомлення між членами дослідницької установи.

На зображеному вище графі ми бачимо розподіл учасників за спільнотами, де колір вузлів описує приналежність до тієї чи іншої спільноти. Крім уже описаних властивостей також існує таке поняття, як взаємність у спрямованій мережі. Ця характеристика описує, який відсоток вихідних зв'язків має зворотний, вхідний, зв'язок. Для того, щоб це дізнатися, використовуємо спеціальну функцію overall_reciprocity бібліотеки networkx і подивимося на рівень взаємності в графі та його компонентах:

Рівень взаємності графа: 0.6933635759258535

Рівень взаємності компоненти слабкої зв'язності: 0.6938791484032562

Рівень взаємності компоненти сильної зв'язності: 0.7169719762222492 У 71% випадків у компоненті сильної зв'язності користувачі отримували відповіді на свої повідомлення. Для компоненти слабкої зв'язності та всього графа загалом рівень передбачувано нижчий.

Підбиваючи підсумок, складні мережі, загалом, володіють певними властивостями, і деякі з них характерні для багатьох мереж. Проведений нами

аналіз датасету з електронними повідомленнями добре підтверджує ці тенденції:

1. Розподіл ступенів вузлів. У всіх мережах є багато вузлів із низьким ступенем, водночас є невелика кількість величезних вузлів, у яких сусідів дуже багато. Це логічно: якщо ми подивимося на посилання між різними web-сайтами, то виявимо, що існує невелика кількість великих сайтів, на які у великій кількості посилаються всі інші (Wikipedia, Microsoft). Водночас на середньостатистичні сайти посилання зустрічаються набагато рідше, хоча більшість сайтів належать саме до такого типу.
2. Діаметр і середня відстань у графі. Великі мережі мають такий устрій, що середній діаметр у них дуже невеликий, це явище в аналізі соціальних мереж називається явищем "малого світу". Воно добре описується через теорію шести рукостискань: незважаючи на величезну кількість людей, середня відстань між двома незнайомими людьми дорівнюватиме шести.
3. У великих мережах, як правило, присутні гігантські зв'язні компоненти: понад 80% вузлів пов'язані між собою, решта представлені більш дрібними компонентами. При цьому в кожній з великих компонент можна зустріти спільноти - групи об'єктів, які пов'язані між собою тісніше, ніж з рештою графа. Наявність кластеризації, тобто великої кількості таких спільнот, надзвичайно поширена в соціальних графах.
4. У багатьох соціальних графах діє принцип взаємності, коли за наявності вихідного зв'язку дуже висока ймовірність зустріти і вхідний зв'язок. Ця концепція специфічна для спрямованих мереж, оскільки взаємність і обмін є фундаментальними соціальними процесами. Усі перелічені вище тенденції ми розібрали і підтвердили для датасету з електронним листуванням: у графі виявилася велика зв'язкова компонента, що містить понад 80% усіх вузлів. Усередині цієї компоненти більшість вузлів характеризувалася невеликою кількістю, проте існував невеликий відсоток учасників, у яких кількість сусідів була величезною. Також ми побачили, що діаметр і середня відстань між учасниками графа невелика:

середня відстань у компоненті слабкої зв'язності, яка містила 986 учасників, становила лише 2.1, що означає, що більшість учасників пов'язані один з одним лише через два "рукостискання". Крім того, граф характеризується високим ступенем взаємності: 69% усіх учасників підтримували двосторонній контакт між собою.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Обґрунтування вибору інструментальних засобів розробки

Наступним етапом програмної реалізації інформаційної технології є вибір інструментальних засобів (бібліотеки мови програмування, модулі, програмні засоби), використовуючи які, будуть реалізовані основні модулі та спроектована система в цілому.

Для розробки програмного засобу буде використано мову програмування високого рівня Python. Перевагами вказаної мови є наявність великої кількості бібліотек, зокрема для роботи з моделями машинного навчання, простий синтаксис.

Наступним кроком є обрання середовища розробки. Запуск програмного засобу буде відбуватися на персональному комп'ютері користувача (фахівець з проведення тестування на проникнення, адміністратор мережі тощо).

Серед найбільш популярних середовищ варто відокремити наступні, оскільки вони передбачають комплексну та швидку розробку програмного забезпечення [16]:

1) Sublime Text 3 – умовно безкоштовне середовище розробки, яке підтримує велику кількість мов програмування, включаючи Python. За замовчуванням має базову підтримку Python, проте для полегшення та пришвидшення розробки необхідно встановлювати пакети доповнень.
2) PyCharm – найпопулярніша середа розробки. Має набір додатків та функцій, які прискорюють розробку програмного забезпечення, які встановлюютсья із середовищем за замовчуванням. Має безкоштовну та платну версії.
3) Visual Studio Code – загальне середовище розробки, яке підтримує різні мови програмування. Постачається з системою автодоповнення. Для розробки програмних засобів мовою Python, необхідно завантажити розширення та виконати налаштування середовища розробки.

Під час розробки програмного засобу також будуть використані наступні

бібліотеки мови Python:

1) NetworkX – один з найбільш широко використовуваних програмних пакетів для роботи з графами та мережами у Python. Цей пакет надає користувачеві різноманітні функції, такі як: опитування, редагування та оновлення графа, пналіз мережі, такий як пошук шляхів, кліки та обернених айлендів, визначення структури мережі, наприклад, коефіцієнта кластеризації та гіперконнективності, симуляція розсіювання та розсіяння, які використовуються для аналізу адаптивної поведінки мережі, обрахунок мережевих розмірів, таких як діаметр, радіус-відстань та централізація, опрацювання інформації про вершини та ребра графа, такі як пунктування та орієнтація ребер, підтримка інтерфейсу betweenness [17].
2) scikit-learn – бібліотека мови Python, яка побудована на NumPy, SciPy та Matplotlib. Призначення бібліотеки: надання ефективних інструментівдля роботи із різними моделями машинного навчання, які включають у себе моделі для класифікації, кластеризації [18].
3) matplotlib – бібліотека Python, яка призначеня для побудови візуалізації різного роду дій у мові Python. Бібліотека була використана під час етапу моделювання моделей машинного навчання для наочної візуалізації результатів навчання та тестування моделей, та їх структури [19].
4) PyQt5 – комплексний набір бібліотек мови Python для створення графічного інтерфейсу програмного засобу для різних операційних систем, включаючи Windows, Linux, iOS та Android. Має вбудовний редактор вікон, який дозволяє пришвидшити розробку програмних засобів із графічним інтерфейсом [20].

Наявність розширень, які встановлені за замовчуванням, власного переліку доповнень, які не вимагають додаткових дій користувача для встановлення, доступ до термналів Bash та Python є основними чинниками, які

роблять середовище PyCharm вибором для розробки програмного засобу, що буде реалізовувати інформаційну технологію.

3.2 Програмна реалізація

Відповідно до алгоритму роботи програмного засобу було створено ряд класів, кожен з яких виконує необхідні для роботи функції:

Клас CyberThreatDetectionApp:

Метод __init__(self, root) - ініціалізує об'єкт CyberThreatDetectionApp і налаштовує основне вікно програми, створює рамки для результатів та відображення графів.

Метод process_graph(self, G) - обробляє граф і визначає спільноти, загрози та інші характеристики графа, формує результати аналізу графа для відображення та збереження.

Метод process_large_graph(self) - обробляє великі файли графів в окремому потоці, завантажує файл, створює граф та запускає процес обробки графа.

Метод load_graph(self) - завантажує граф з файлу, обробляє його та відображає результати.

detect_communities(self, graph): Ця функція призначена для виявлення спільнот (груп) в графі. Функція повертає списки спільнот, знайдених алгоритмами.

get_edge_weights(self, graph): Ця функція витягує ваги ребер у графі. Вона використовує nx.get_edge_attributes для отримання атрибуту 'weight' для кожного ребра у графі. Якщо атрибут відсутній, функція поверне порожній словник.

graph_info(self, graph): Ця функція повертає загальну інформацію про граф, таку як кількість вузлів та ребер у графі, а також списки вузлів та ребер.

detect_threats(self, graph): Функція використовує алгоритм Girvan-Newman для виявлення загроз у графі. Вона аналізує вузли та їх зв'язки для визначення

потенційних загроз. Повертає список топ-рівневих спільнот, список загроз, сильні та слабкі спільноти.

strong_communities(self, graph, communities): Ця функція призначена для виявлення "міцних" спільнот, де кількість внутрішніх зв'язків більше, ніж кількість зовнішніх зв'язків для кожної спільноти.

weak_communities(self, graph, communities): Функція виявляє "слабкі" спільноти, у яких кількість зовнішніх зв'язків перевищує кількість внутрішніх зв'язків для кожної спільноти.

Метод display_results(self, ...) та draw_graph(self, graph, communities) - відображення результатів аналізу графа та візуалізація графа.

Основна функція main() - ініціалізує головне вікно програми та створює об'єкт CyberThreatDetectionApp.

Цей код містить обробку графів, визначення спільнот, ваг ребер, загрози та візуалізацію графів. Також передбачена можливість збереження результатів у файлі.

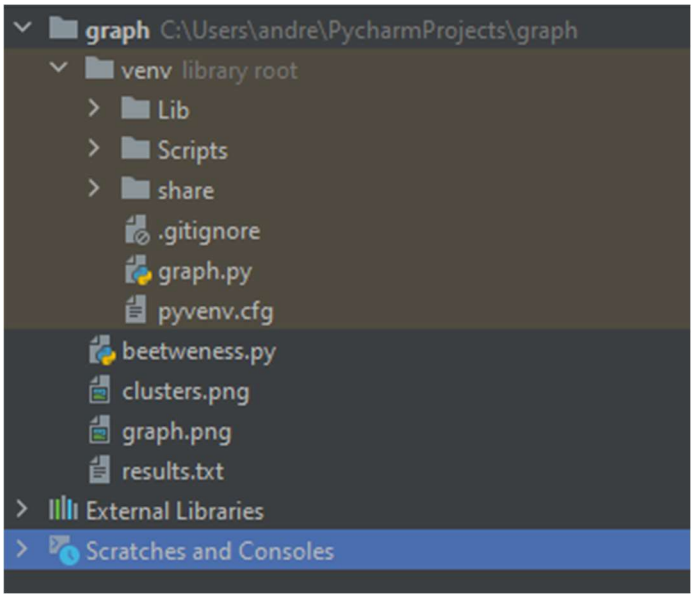


Рисунок 3.1 – Структура проекту

Текст програмного засобу наведено у додатку Б. Для перевірки розробленого програмного засобу необхідно провести тестування.

3.3 Тестування програмного засобу

Тестування буде проведено з використанням двох сценаріїв:

- перевірка правильної роботи програмного засобу;
- перевірка роботи програмного засобу при використанні неправильних даних.

Для запуску програмного засобу необхідно відкрити файл.

Користувачеві буде відображено головне вікно програми (рис. 3.2).

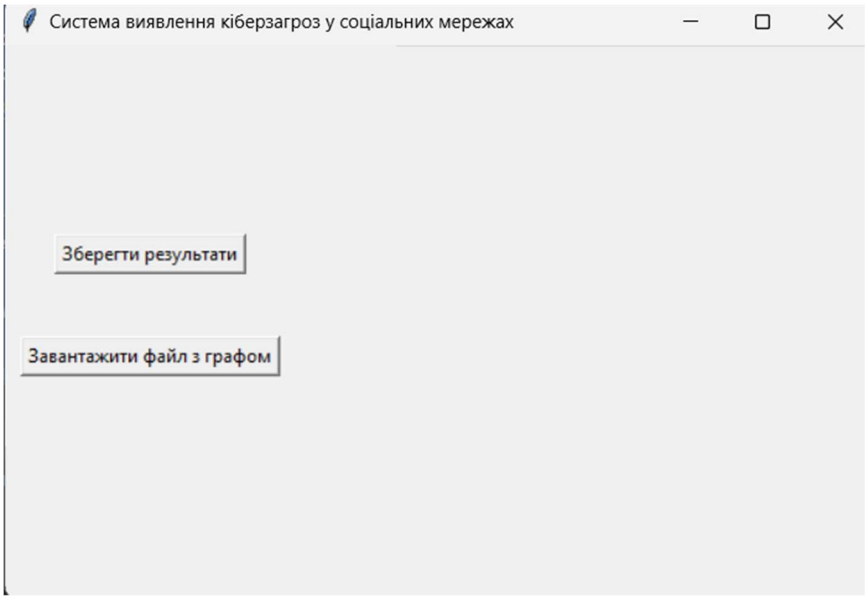


Рисунок 3.2 – Головне вікно програми

У вікні є кнопки керування (завантаження графу та початок його аналізу, обрати файл для аналізу, збереження результатів) та кнопка завершення роботи. Після завантаження файлу розпочнеться процес обрахунків та виведення інформації про граф.

Для перевірки роботи програмного засобу вибраний простий граф невеликої групи спільнот (рис 3.2).

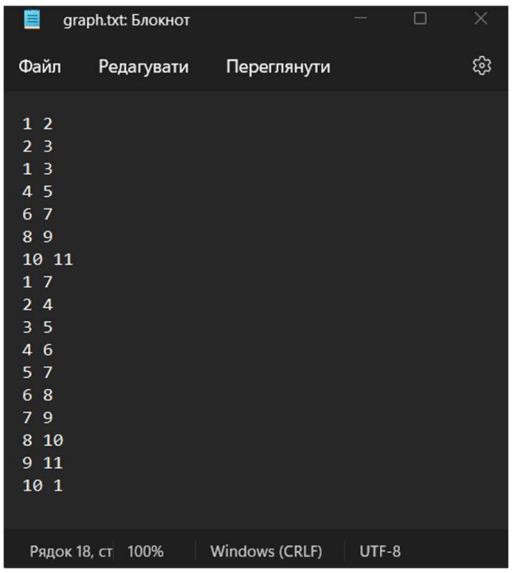


Рисунок 3.3 – Тестовий граф

Оскільки файл завантажений правильний, через деякий час отримуємо результат аналізу, а саме – операційна система та достовірність того, що виявлена операційна система належить визначеному сімейству та достовірність визначення операційної системи всередині сімейства (рис 3.3)

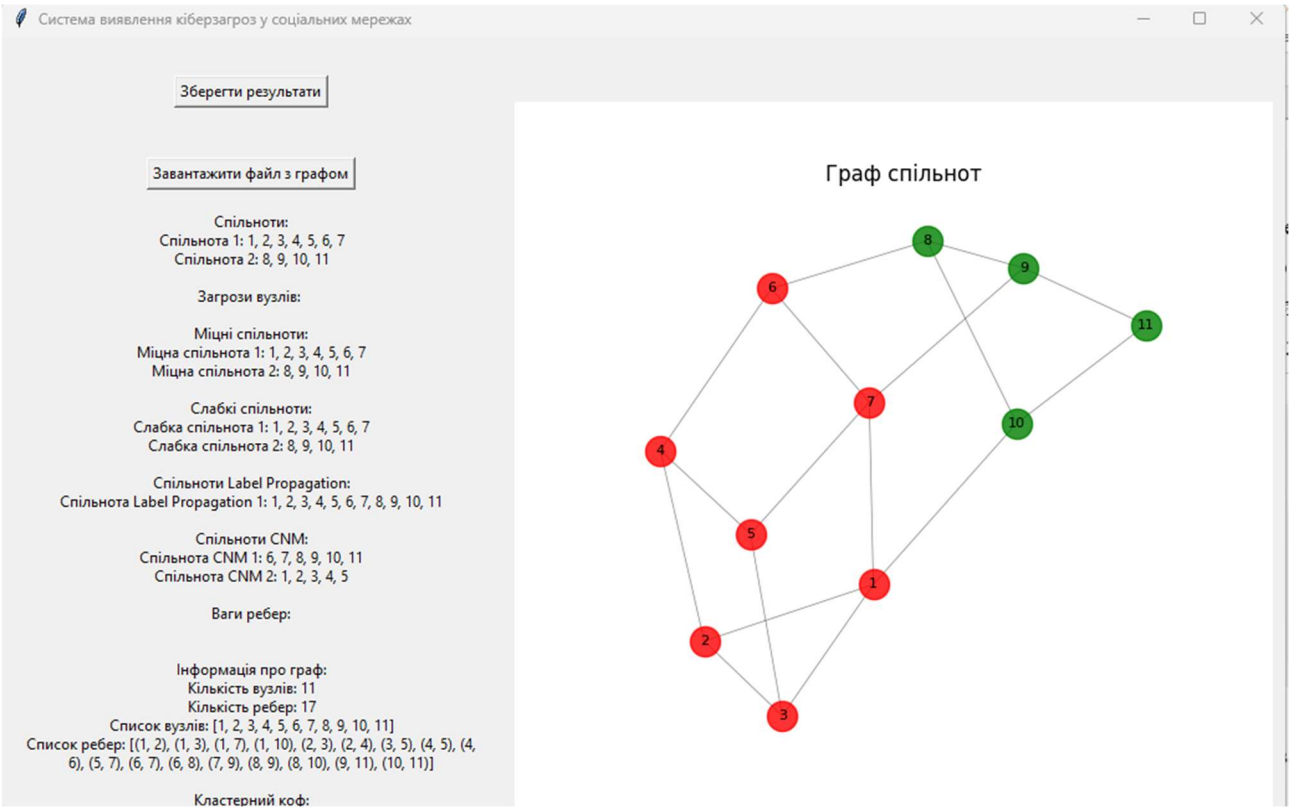


Рисунок 3.4 – Результат аналізу графа

Після кожного сканування зберігається звіт у папці з програмним засобом у форматі файлу txt. Звіт (вміст файлу та приклад завантаження звіту у текстовий редактор) наведено на рис. 3.5.

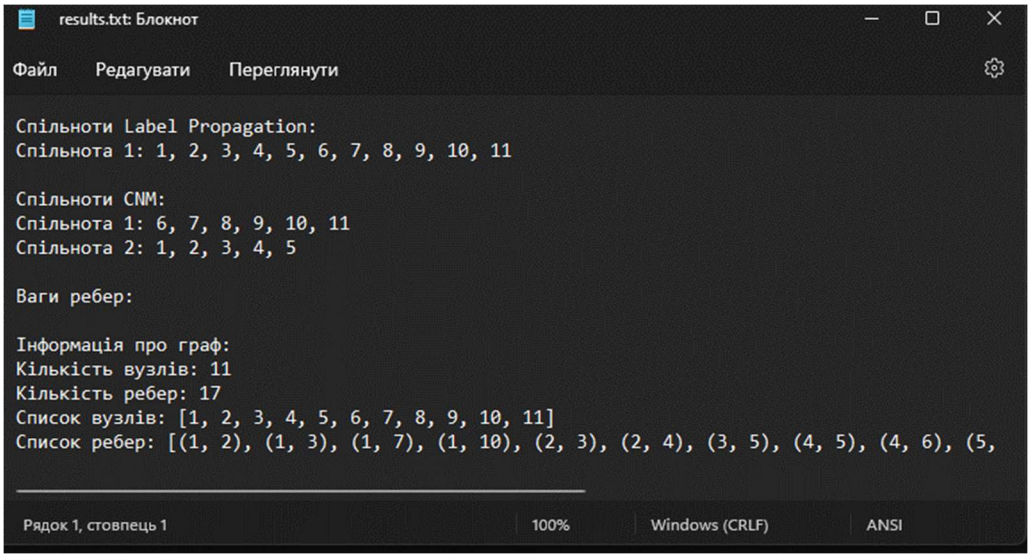


Рисунок 3.5 – Вміст звіту

Проаналізувавши усі отримані результати можна зробити висновок, що розроблений програмний засіб є потужним інструментом для виявлення кіберзагроз у соціальних мережах. Він дозволяє користувачеві ефективно аналізувати графи мереж, визначати спільноти, а також виявляти можливі загрози для мережевих вузлів. Інтерфейс програми забезпечує зручність та доступність для взаємодії з користувачем, включаючи завантаження файлів з графами, візуалізацію графів та відображення результатів аналізу.

Програма використовує різні методи для визначення спільнот та виявлення загроз, а також враховує внутрішні та зовнішні зв'язки для визначення міцних та слабких спільнот. Крім того, програма забезпечує високу надійність та стабільність роботи, оскільки вона ефективно обробляє можливі помилки введення даних та відображає відповідні повідомлення про помилки користувачеві. Загалом, дана програма є цінним інструментом для аналізу соціальних мереж з точки зору виявлення потенційних кіберзагроз, допомагаючи користувачеві ефективно аналізувати та захищати мережу від можливих загроз.

4 ЕКОНОМІЧНА ЧАСТИНА

Для успішного впровадження науково-технічної розробки надзвичайно важливо, щоб вона відповідала сучасним вимогам науково-технічного прогресу та враховувала економічні аспекти. Оцінка економічної ефективності результатів науково-дослідної роботи є критичною частиною цього процесу. Дослідження, яке представлене у магістерській роботі і присвячене розробці та вивченню «Система виявлення кіберзагроз, що створюються спільнотами в соціальних мережах. Частина 1. Модуль моніторингу та аналізу спільнот за допомогою теорії графів», належить до науково-технічних проектів, спрямованих на введення на ринок. Рішення про комерціалізацію розробки може бути прийняте протягом виконання самої роботи, відкриваючи можливості для подальшого введення на ринок. Цей напрямок визначається як пріоритетний, оскільки розроблені результати можуть бути корисними для різних зацікавлених сторін і приносити економічні вигоди. Однак для успішної реалізації цього процесу вирішальним є залучення зацікавленого інвестора, який виявить інтерес до втілення даного проекту, і переконання його у доцільності інвестування у цю розробку. З метою досягнення цього завдання були визначені такі етапи виконання робіт:

1. Проведення комерційного аудиту науково-технічної розробки, включаючи визначення науково-технічного рівня та комерційного потенціалу.
2. Розрахунок витрат на реалізацію науково-технічної розробки.
3. Проведення розрахунку економічної ефективності впровадження та комерціалізації науково-технічної розробки для потенційного інвестора, а також обґрунтування економічної доцільності комерціалізації з точки зору інвестора.

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Система виявлення кіберзагроз, що створюються спільнотами в соціальних мережах. Частина 1. Модуль моніторингу та аналізу спільнот за допомогою теорії графів» є покращення кібербезпеки шляхом створення системи для пошуку та аналізу небезпечного контенту.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 4.1 [22].

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в	Технічні та споживчі влас-тивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в	Технічні та споживчі властивості продукту значно кращі, ніж в
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					

6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуютьс я у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь- які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці. Для оцінки науково-технічного рівня і комерційного потенціалу розробки експертами було

запрошено трьох незалежних експертів Вінницького національного технічного університету з кафедри «Захисту інформації»: Кондратенко Наталія Романівна к. т. н., професор, Дудатьєв Андрій Веніамінович к. т. н., доцент, Лужецький Володимир Андрійович д. т. н., професор

Таблиця 4.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	Лужецький В.А	Дудатьєв А.В.	Кондратенко Н.Р.
	Бали, виставлені експертами:		
1.Технічна здійсненність концепції	3	4	4
2.Ринкові переваги (наявність аналогів)	2	2	2
3. Ринкові переваги (ціна продукту)	3	4	3
4. Ринкові переваги (технічні властивості)	2	3	3
5. Ринкові переваги (експлуатаційні витрати)	3	3	2
6. Ринкові перспективи (розмір ринку)	1	1	2
7. Ринкові перспективи (конкуренція)	3	2	3
8. Практична здійсненність (наявність фахівців)	4	4	4
9. Практична здійсненність (наявність фінансів)	4	4	4
10. Практична здійсненність (необхідність нових матеріалів)	4	4	3
11. Практична здійсненність (термін реалізації)	4	4	4
12. Практична здійсненність (розробка документів)	4	4	4
Сума балів	$СБ_1$=37	$СБ_2$=39	$СБ_3$=38
Середньоарифметична сума балів $СБ_с$	$\overline{СБ} = \frac{\sum_1^i СБ_i}{i} = \frac{37+39+38}{3} = 38$		

За результатами розрахунків, наведених в таблиці 4.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 4.3 [22].

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ , розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41…48	Високий
31…40	Вище середнього
21…30	Середній
11…20	Нижче середнього
0…10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Система виявлення кіберзагроз, що створюються спільнотами в соціальних мережах. Частина 1. Модуль моніторингу та аналізу спільнот за допомогою теорії графів» становить 38 балів, що, відповідно до таблиці 4.3 рівень комерційного потенціалу розробки вище середнього, що свідчить про комерційну важливість проведення даних досліджень.

Магістерська кваліфікаційна робота «Система виявлення кіберзагроз, що створюються спільнотами в соціальних мережах. Частина 1. Модуль моніторингу та аналізу спільнот за допомогою теорії графів» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок, тобто при цьому відбувається комерціалізація науково-технічної розробки. Цей напрямок є для нас пріоритетним, оскільки результатами розробки можуть користуватися не тільки самі розробники, а й інші споживачі, отримуючи при цьому суттєвий економічний ефект.

Результатом магістерської роботи є програмний засіб аналізу кіберзагроз в соціальних мережах шляхом виявлення спільнот, який може бути корисним компаніям з кібербезпеки, центрам обробки даних та аналітики, організаціям, що займаються боротьбою з кіберзлочинністю, соціальним платформам та мережам.

4.2 Визначення рівня конкурентоспроможності розробки

В процесі визначення економічної ефективності науково-технічної розробки також доцільно провести прогноз рівня її конкурентоспроможності за сукупністю параметрів, що підлягають оцінюванню.

В якості аналога для розробки було обрано GraphAware. Основними недоліками аналога є проблема яка потребує від користувачів глибоких технічних знань у сфері графових баз даних та аналізу графів. Це може бути складним для тих, хто не має великого досвіду у цій області. Також до недоліків можна віднести обмежені можливості без платної версії:

У розробці дана проблема вирішується за зниження вартості, спрощення використання, розширення можливостей без обмежень та уникнення технічних складнощів у порівнянні з аналогами. Також система випереджає аналог за такими параметрами як: швидкістю виявлення, точністю, простотою використання, обробкою великих обсягів даних, адаптивністю до змін та підтримкою різних платформ шифрування даних.

Одиничний параметричний індекс розраховуємо за формулою [22]:

$$q_i = \frac{P_i}{P_{баз\,i}}. \tag{4.1}$$

де q_i – одиничний параметричний індекс, розрахований за i-м параметром;

P_i – значення i-го параметра виробу;

$P_{базі}$ – аналогічний параметр базового виробу-аналога, з яким проводиться порівняння.

Загальні технічні та економічні характеристики розробки представлено в таблиці 4.4.

Таблиця 4.4 – Основні техніко-економічні показники аналога та розробки, що проектується

Показник	Варіанти		Відносний показник якості	Коефіцієнт вагомості параметра
	Базовий (товар-конкурент)	Новий (інноваційне рішення)		
1	2	3	4	5
Точність виявлення	60	100	1,7	30%
Функціонал	5	12	2,4	11%
Похибка, %	2	1	2	30%
Розмір програми, МБ	400	130	3,1	29%

Нормативні параметри оцінюємо показником, який отримує одне з двох значень: 1 – пристрій відповідає нормам і стандартам; 0 – не відповідає.

Груповий показник конкурентоспроможності за нормативними параметрами розраховуємо як добуток частинних показників за кожним параметром за формулою [22]:

$$I_{НП} = \prod_{i=1}^{n} q_i \,, \tag{4.2}$$

де $I_{нп}$ – загальний показник конкурентоспроможності за нормативними параметрами;

q_i – одиничний (частинний) показник за i-м нормативним параметром;

n – кількість нормативних параметрів, які підлягають оцінюванню.

За нормативними параметрами розроблюваний пристрій відповідає вимогам ДСТУ, тому $I_{нп} = 1$.

Значення групового параметричного індексу за технічними параметрами визначаємо з урахуванням вагомості (частки) кожного параметра [22]:

$$I_{ТП} = \sum_{i=1}^{n} q_i \cdot \alpha_i \,, \tag{4.3}$$

де $I_{тп}$ – груповий параметричний індекс за технічними показниками (порівняно з виробом-аналогом);

q_i – одиничний параметричний показник i-го параметра;

α_i – вагомість i-го параметричного показника, $\sum_{i=1}^{n}\alpha_i = 1$;

n – кількість технічних параметрів, за якими оцінюється конкурентоспроможність.

Проведемо аналіз параметрів згідно даних таблиці 4.4.

$$I_{mn} = 1{,}7 \cdot 0{,}3 + 2{,}4 \cdot 0{,}11 + 2 \cdot 0{,}3 + 3{,}1 \cdot 0{,}29 = 2{,}2.$$

Груповий параметричний індекс за економічними параметрами розраховуємо за формулою [23]:

$$I_{ЕП} = \sum_{i=1}^{m} q_i \cdot \beta_i , \qquad (4.4)$$

де $I_{ЕП}$ – груповий параметричний індекс за економічними показниками;

q_i – економічний параметр i-го виду;

β_i – частка i-го економічного параметра, $\sum_{i=1}^{m}\beta_i = 1$;

m – кількість економічних параметрів, за якими здійснюється оцінювання.

Проведемо аналіз параметрів згідно даних таблиці .

$$I_{ЕП} = 0{,}75 \cdot 0{,}5 + 0{,}86 \cdot 0{,}5 = 0{,}80.$$

На основі групових параметричних індексів за нормативними, технічними та економічними показниками розрахуємо інтегральний показник конкурентоспроможності за формулою [22]:

$$K_{ІНТ} = I_{НП} \cdot \frac{I_{ТП}}{I_{ЕП}}, \qquad (4.5)$$

$$K_{ІНТ} = 1 \cdot 2{,}2 / 0{,}80 = 2{,}8.$$

Інтегральний показник конкурентоспроможності $К_{ІНТ} > 1$, отже розробка переважає відомі аналоги за своїми техніко-економічними показниками.

4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов’язані з проведенням науково-дослідної роботи на тему «Система виявлення кіберзагроз, що створюються спільнотами в соціальних

мережах. Частина 1. Модуль моніторингу та аналізу спільнот за допомогою теорії графів», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

4.3.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників ($З_о$) розраховуємо у відповідності до посадових окладів працівників, за формулою [22]:

$$З_о = \sum_{i=1}^{k} \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.6)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, T_p=21 дні.

$$З_о = 15000 \cdot 5 / 21 = 3409 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.5 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту	15000	681,8	5	3409
Програміст	10000	454,5	35	15909
Всього				19318

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт НДР на тему «Система виявлення кіберзагроз, що створюються спільнотами в соціальних мережах. Частина 1. Модуль моніторингу та аналізу спільнот за допомогою теорії графів» розраховуємо за формулою:

$$З_р = \sum_{i=1}^{n} C_i \cdot t_i \,, \tag{4.7}$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \tag{4.8}$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), приймемо $Мм$=6500 грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду [23];

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об’єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середнє число робочих днів в місяці, приблизно T_p = 21 дн;

$t_{зм}$ – тривалість зміни, год.

$C_1 = 6700{,}00 \cdot 1 \cdot 1{,}65 / (21 \cdot 8) = 65{,}8$ грн.
$З_{р1} = 65{,}8 \cdot 2 = 131{,}6$ грн.

Таблиця 4.6 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
1. Підготовка робочого місця інженера-розробника ПЗ	2	1	65,8	131,6
2. Інсталяція програмного забезпечення середовищ моделювання та розробки	2	3	88,8	177,7
3.Розробка програмної архітектури та алгоритмів	4	5	111,9	447,5
4. Написання програмного коду модулів	6	2	72,4	434,3
5. Програмне тестування дослідного зразка	4	4	59,8	239,3
Всього				1430,3

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 … 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{дод} = (З_о + З_р) \cdot \frac{Н_{дод}}{100\%}, \quad (4.9)$$

де $Н_{дод}$ – норма нарахування додаткової заробітної плати. Приймемо 11%.

$З_{дод}$ = (19318+1430,3) · 11 / 100% = 2282,34 грн.

4.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$З_н = (З_о + З_р + З_{дод}) \cdot \frac{Н_{зп}}{100\%} \quad (4.10)$$

де $Н_{зп}$ – норма нарахування на заробітну плату. Приймаємо 22%.

$Зн$ = (19318+1430,3+2282,34) · 22 / 100% = 5066,79 грн.

4.3.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Система виявлення кіберзагроз, що створюються спільнотами в соціальних мережах. Частина 1. Модуль моніторингу та аналізу спільнот за допомогою теорії графів».

Витрати на матеріали (*М*), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^{n} H_j \cdot Ц_j \cdot К_j - \sum_{j=1}^{n} B_j \cdot Ц_{вj}, \qquad (4.11)$$

де H_j – норма витрат матеріалу *j*-го найменування, кг;

n – кількість видів матеріалів;

$Ц_j$ – вартість матеріалу *j*-го найменування, грн/кг;

$К_j$ – коефіцієнт транспортних витрат, ($К_j$ = *1,1 ... 1,15*);

B_j – маса відходів *j*-го найменування, кг;

$Ц_{вj}$ – вартість відходів *j*-го найменування, грн/кг.

Проведені розрахунки зведемо до таблиці.

Таблиця 4.7 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Вартість витраченого матеріалу, грн
Папір канцелярський офісний (А4)	170	1	170
ручка	50	1	50
Флешка	250	1	250
Всього			470
З врахуванням коефіцієнта транспортування			517

4.3.4 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_б}{T_в} \cdot \frac{t_{вик}}{12}, \quad (4.12)$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_в$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$A_{обл} = (2600 \cdot 2) / (2 \cdot 12) = 216{,}67$ *грн.*

Проведені розрахунки зведемо до таблиці.

Таблиця 4.8 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
1. IDE PyCharm	2600	2	2	216,67
2.Компютерна техніка	14000	2	2	1166,67
3. Сервіс шарингу	3000	2	1	125,00
Всього				1508,33

4.3.5 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію ($B_е$) розраховуємо за формулою:

$$B_е = \sum_{i=1}^{n} \frac{W_{уі} \cdot t_i \cdot Ц_е \cdot K_{впі}}{\eta_i}, \quad (4.13)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

$Ц_e$ – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), приймемо $Ц_e$ = 7,5 грн;

$K_{вп i}$ – коефіцієнт, що враховує використання потужності, $K_{вп i} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$В_e = 0,25 \cdot 320,0 \cdot 7,5 \cdot 0,5 / 0,8 = 375$ грн.

4.3.6 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Система виявлення кіберзагроз, що створюються спільнотами в соціальних мережах. Частина 1. Модуль моніторингу та аналізу спільнот за допомогою теорії графів» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20…25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{св} = (З_o + З_p) \cdot \frac{H_{св}}{100\%}, \qquad (4.14)$$

де $H_{св}$ – норма нарахування за статтею «Службові відрядження», приймемо $Н_{св}$ = 20%.

$В_{св} = (19318+1430,3) \cdot 20 / 100\% = 4149,70$ грн.

4.3.8 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50…100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_в = (З_о + З_р) \cdot \frac{Н_{ів}}{100\%}, \quad (4.15)$$

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати», приймемо $Н_{ів}$ = 50%.

$І_в = (19318+1430,39) \cdot 50 / 100\% = 10374,26$ грн.

4.3.9 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100…150% від суми основної заробітної плати дослідників та робітників за формулою:

$$В_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \quad (4.16)$$

де $Н_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», приймемо $Н_{нзв}$ = 100%.

$В_{нзв} = (19318+1430,3) \cdot 100 / 100\% = 20748,51$ грн.

Витрати на проведення науково-дослідної роботи на тему «Система виявлення кіберзагроз, що створюються спільнотами в соціальних мережах. Частина 1. Модуль моніторингу та аналізу спільнот за допомогою теорії графів» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = З_о + З_р + З_{дод} + З_н + М + К_в + В_{спец} + В_{прг} + А_{обл} + В_е + В_{св} + В_{сп} + І_в + В_{нзв}. \quad (4.17)$$

$B_{заг}$=19318+1430,3+2282,34+5066,79+517+1508,33+375+4149,70+10374,26 +20748,51= 65770,44 грн.

Загальні витрати *ЗВ* на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ЗВ = \frac{B_{заг}}{\eta}, \quad (4.18)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, приймемо η=0,7.

ЗВ = 65770,44 / 0,7= 93957,77 грн.

4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою «Система виявлення кіберзагроз, що створюються спільнотами в соціальних мережах. Частина 1. Модуль моніторингу та аналізу спільнот за допомогою теорії графів» передбачають комерціалізацію протягом 3-х років реалізації на ринку.

В цьому випадку основу майбутнього економічного ефекту будуть формувати:

ΔN – збільшення кількості споживачів яким надається відповідна інформаційна послуга у періоди часу, що аналізуються;

N – кількість споживачів яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки, приймемо 1 особа

$Ц_б$ – вартість послуги у році до впровадження інформаційної системи, приймемо 2000,00 грн;

$\pm\Delta Ц_о$ – зміна вартості послуги від впровадження результатів, приймемо зростання на 500,00 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta П_i$ для кожного із 3-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [21]:

$$\Delta П_i = (\pm\Delta Ц_о \cdot N + Ц_о \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \quad (4.19)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2023 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту). Приймемо ρ =40%;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році $\vartheta = 18\%$;

Збільшення чистого прибутку 1-го року:

$\Delta П_1 = (1 \cdot 500 + 20000 \cdot 70) \cdot 0,83 \cdot 0,4 \cdot (1\text{-}0,18/100\%) = 245221,44$ грн.

Збільшення чистого прибутку 2-го року:

$\Delta П_2 = (1 \cdot 500 + 20000 \cdot (70+60)) \cdot 0{,}83 \cdot 0{,}4 \cdot (1-0{,}18/100\%) = 455752{,}62$ грн.

Збільшення чистого прибутку 3-го року:

$\Delta П_3 = (1 \cdot 500 + 20000 \cdot (70+60+50)) \cdot 0{,}83 \cdot 0{,}4 \cdot (1-0{,}18/100\%) = 630849{,}79$ грн.

Приведена вартість збільшення всіх чистих прибутків *ПП*, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \tag{4.20}$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, τ =18%;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$ПП = 245221{,}44 /(1+0{,}18)^1 + 455752{,}62 /(1+0{,}18)^2 + 630849{,}79 /(1+0{,}18)^3 =$

$= 887619{,}21$ грн.

Величина початкових інвестицій PV, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{інв} \cdot ЗВ, \tag{4.21}$$

де $k_{інв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{інв}$=2;

$ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 93957,77 грн.

$PV = k_{інв} \cdot ЗВ = 2 \cdot 93957{,}77 = 187915{,}54$ грн.

Абсолютний економічний ефект $E_{абс}$ для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV \tag{4.22}$$

де $ПП$ – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 887619,21 грн;

PV – теперішня вартість початкових інвестицій, 187915,54 грн.

$E_{абс} = ПП - PV = 887619{,}21 - 187915{,}54 = 699703{,}67$ грн.

Внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1, \tag{4.23}$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, грн;

PV – теперішня вартість початкових інвестицій, грн;

$T_ж$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримування позитивних результатів від її впровадження, 3 роки.

$$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1 = (1+699703{,}67 \ /187915{,}54)^{1/3} - 1 = 1{,}04.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{мін}$:

$$\tau_{мін} = d + f, \tag{4.24}$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні d =0,1;

f – показник, що характеризує ризикованість вкладення інвестицій, приймемо 0,25.

$\tau_{мін}$ = 0,1+0,25 = 0,35 < 1,04 свідчить про те, що внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Система виявлення кіберзагроз, що створюються спільнотами в соціальних мережах. Частина 1. Модуль моніторингу та аналізу спільнот за допомогою теорії графів» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_в}, \tag{4.25}$$

де $E_в$ – внутрішня економічна дохідність вкладених інвестицій.

$T_{ок}$ = 1 / 1,04 = 1 р.

$T_{ок}$ < 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Система виявлення кіберзагроз, що створюються спільнотами в соціальних мережах. Частина 1. Модуль моніторингу та аналізу спільнот за допомогою теорії графів» становить 38 бали, що, свідчить про комерційну важливість проведення даних досліджень оскільки рівень комерційного потенціалу розробки вище середнього.

При оцінюванні рівня конкурентоспроможності, згідно узагальненого коефіцієнту конкурентоспроможності розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 2,8 рази.

Також термін окупності становить 1 рік, що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Система виявлення кіберзагроз, що створюються спільнотами в соціальних мережах. Частина 1. Модуль моніторингу та аналізу спільнот за допомогою теорії графів».

ВИСНОВКИ

У ході виконання магістерської кваліфікаційної роботи було виконано усі поставлені задачі. Було проведено аналіз графів та їх особливостей. Було проаналізовано актуальність вирішення задачі визначення кіберзагроз у графах. Проведено аналіз методів та існуючих засобів виявлення спільнот у графах. Визначено, що проблема є доволі актуальною. Прийнято рішення покращити процес виявлення спільнот. Також було обрано технології програмування для вирішення задачі.

Виконано обґрунтування вибору інструментальних засобів розробки, результати: мова програмування – Python, середовище розробки – PyCharm. На основі створених вимог до програмного засобу, створених алгоритмів та архітектури системи, створено програмний засіб, що реалізує кластеризацію графів, визначення міцних та слабких спільнот за певними критеріями, а також відображення ваг ребер, інформації про граф, та виведення результатів аналізу у вигляді текстових даних та візуалізації графів. Створений програмний засіб має простий та зрозумілий інтерфейс. Для використання потрібно лише запустити виконуваний файл, який знаходиться у папці разом із необхідними файлами.

Програма була протестована для різних сценаріїв роботи з використанням правильних та неправильних даних, де була виявлена здатність програми обробляти помилки та надавати відповідні повідомлення для користувача щодо усунення проблем.

Згідно проведених досліджень щодо рівня комерційного потенціалу розробки, результати вказують на комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки є вищим за середній). При оцінюванні рівня конкурентоспроможності, було визначено, що науково-технічна розробка переважає існуючі аналоги приблизно в 2,8 рази. Також термін окупності становить 1 р., що свідчить про комерційну привабливість науково-технічної розробки. Вказані результати свідчать про доцільність

проведення наукових досліджень.

Даний програмний засіб є корисним застосунком, який покращує процес виявлення спільнот, а саме аналіз графу. Також адміністратори мереж можуть використовувати розроблений програмний засіб для визначення спільнот, кіберзагроз, та інших характеристик графа.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Основи теорії і практики інтелектуального аналізу даних у сфері кібербезпеки: навчальний посібник / Д.В. Ланде, І.Ю. Субач, Ю.Є. Бояринова. – К.: ІСЗЗІ КПІ ім. Ігоря Сікрорського, 2018. – 300 с. ISBN 978-966-2577-12-9.
2. Сучасні інформаційні війни в соціальних онлайн-мережах / О. В. Курбан // Інформаційне суспільство. – 2016. – Вип. 23. – С. 85–90
3. Дослідження властивостей інформації та методів її поширення з точки зору інформаційної безпеки в соціальних мережах / Є.В. Мелешко, Л.В. Константинова, О.С. Улічев // Системи управління, навігації та зв'язку. – 2015. – Вип. 3(35). – С. 98-106.
4. A graph-theoretic generalization of the clique concept / S. B. Seidman, B. L. Foster // Journal of Mathematical Sociology. – 1978. – Vol. 6, No. 5. – P. 139–154.
5. Алгоритми та структури даних: навч. посібник / В. О. Гороховатський. – Харків: ХНУРЕ, 2017. – 50 с.
6. Комп'ютерна дискретна математика: навч. посібник / В. Білоус. – Харків: ХНУРЕ, 2017.
7. Інформаційні операції та безпека суспільства: загрози, протидія, моделювання: монографія / В. П. Горбулін, О. Г. Додонов, Д. В. Ланде. – К.: Інтертехнологія, 2009. – 164 с.
8. Вплив віртуальних спільнот на інформаційну безпеку: сучасний стан та тенденції розвитку / І. Гриненко, Д. Прокофьєва-Янчиленко // Правове, нормативне та метрологічне забезпечення систем захисту інформації в Україні. – 2012. – № 1 (23). – С. 18 – 23.
9. Community detection in graphs / S. Fortunato // Physics Reports. – 2010. – Vol. 486, No. 3-5. – P. 75-174.
10. Facebook Graph API. – URL:https://developers.facebook.com/docs/graph-api/overview/ (дата завернення:21.09.2023)
11. Twitter API. – URL:https://developer.twitter.com/en/docs (дата завернення: 21.09.2023)
12. Мелешко Є.В. Дослідження методів побудови рекомендаційних систем в мережі Інтернет / Є.В. Мелешко, Г.С. Семенов, В.Д. Хох. // Збірник наукових праць

"Системи управління, навігації та зв'язку". Випуск 1(47). – Полтава: ПНТУ ім. Ю. Кондратюка. – 2018. – С. 131–136.

13. Recommender Systems Handbook / Editors Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor. – 1st edition. – New York, NY, USA: Springer-Verlag New York, Inc., 2010. – 842 с.
14. Meleshko, Ye. "МЕТОДИ КЛАСТЕРИЗАЦІЇ ГРАФІВ СОЦІАЛЬНИХ МЕРЕЖ ДЛЯ ПОБУДОВИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ". Системи управління, навігації та зв'язку. Збірник наукових праць 2, № 54 (11 квітня 2019): 129–34.
15. Shutka, P., та A. Filonenko. "ОГЛЯД ПРОБЛЕМНО-ОРІЄНТОВАНИХ МОВ ПРОГРАМУВАННЯ ДЛЯ ПАРАЛЕЛЬНОГО АНАЛІЗУ СТАТИЧНИХ ГРАФІВ". Системи управління, навігації та зв'язку. Збірник наукових праць 6, № 52 (13 грудня 2018): 126–29.
16. Стєганцева, П. Г., та А. О. Артеменко. "РЕКУРЕНТНІ СПІВВІДНОШЕННЯ ДЛЯ ЧИСЛА НЕІЗОМОРФНИХ (n,m)-ГРАФІВ". Visnyk of Zaporizhzhya National University Physical and Mathematical Sciences, № 1 (6 вересня 2021): 51–56.
17. 9 Best Python IDEs and Code Editors : веб-сайт. URL : https://www.programiz.com/python-programming/ide (дата звернення 10.11.2022).
18. Scikit Learn – Introduction : URL: https://www.tutorialspoint.com/scikit_learn/scikit_learn_introduction.htm (дата звернення 05.12.2022).
19. Matplotlib: Visualization with Python : URL: https://matplotlib.org/ (дата звернення 05.12.2022).
20. What is PyQt? : URL: https://www.riverbankcomputing.com/software/pyqt/ (дата звернення 06.12.2022).
21. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько,В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.
22. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа – Вінниця : ВНТУ,2016. – 113 с.

ДОДАТКИ

Додаток А

ПРОТОКОЛ ПЕРЕВІРКИ
МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Система виявлення кіберзагроз, що створюються спільнотами в соціальних мережах. Частина 1. Модуль визначення спільнот з використанням компоновки в графах

Автор роботи: Ніколайчук Андрій Вадимович

Тип роботи: магістерська кваліфікаційна робота

Підрозділ: кафедра захисту інформації ФІТКІ
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність – 88,92%. Схожість– 11,8%.

Аналіз звіту подібності (відмітити потрібне):

√ 1. Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

□ 2. Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостій- ності її виконання автором. Роботу направити на розгляд експертної комісіїкафедри.

□ 3. Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ (підпис) Валентина КАПЛУН (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ (підпис) Андрій НІКОЛАЙЧУК (прізвище, ініціали)

Керівник роботи _____ (підпис) Наталія КОНДРАТЕНКО (прізвище, ініціали)

Додаток Б
Текст програми

```
import networkx as nx
import matplotlib.pyplot as plt
import tkinter as tk
from tkinter import filedialog, messagebox
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
import threading

from networkx.algorithms.approximation import clustering_coefficient


class CyberThreatDetectionApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Система виявлення кіберзагроз у соціальних мережах")
        self.root.geometry("1000x800")

        self.result_frame = tk.Frame(self.root)
        self.result_frame.pack(side=tk.LEFT, padx=10, pady=10)
        self.save_button = tk.Button(self.result_frame, text="Зберегти результати",
command=self.save_results)
        self.save_button.pack(pady=20)
        self.graph_frame = tk.Frame(self.root)
        self.graph_frame.pack(side=tk.RIGHT, padx=10, pady=10)

        self.load_button = tk.Button(self.result_frame, text="Завантажити файл з
графом", command=self.load_graph)
        self.load_button.pack(pady=20)

        self.result_label = tk.Label(self.result_frame, text="", wraplength=380)
        self.result_label.pack()

        self.canvas = tk.Canvas(self.graph_frame, width=800, height=600)
        self.canvas.pack()

    def process_graph(self, G):
        try:
            communities_label, communities_cnm = self.detect_communities(G)
            edge_weights = self.get_edge_weights(G)
            graph_info = self.graph_info(G)
            top_level_comm, threats, strong_comms, weak_comms =
self.detect_threats(G)

            communities_label_str = "\n".join(
                [f"Спільнота Label Propagation {i + 1}: {', '.join(map(str, comm))}"
for i, comm in
                 enumerate(communities_label)])
            communities_cnm_str = "\n".join(
                [f"Спільнота CNM {i + 1}: {', '.join(map(str, comm))}" for i, comm
in enumerate(communities_cnm)])

            edge_weights_str = "\n".join([f"Ребро {edge}: Вага {weight}" for edge,
weight in edge_weights.items()])
            edge_info = f"Ваги ребер:\n{edge_weights_str}\n\n"

            graph_info_str = f"Інформація про граф:\n{graph_info}\n"

            result_text = f"Спільноти Label
Propagation:\n{communities_label_str}\n\n"
```

```
            result_text += f"Спільноти CNM:\n{communities_cnm_str}\n\n"
            result_text += edge_info
            result_text += graph_info_str

            filename = "results.txt"
            self.save_results_to_file(filename, result_text)
        except Exception as e:
            messagebox.showerror("Помилка", f"Помилка обробки графа: {e}")

    def process_large_graph(self):
        file_path = filedialog.askopenfilename(filetypes=[("Text files", "*.txt"),
("All files", "*.*")])
        if file_path:
            try:
                with open(file_path, 'r') as file:
                    lines = file.readlines()
                    edges = [tuple(map(int, line.strip().split())) for line in
lines]
                G = nx.Graph(edges)
                processing_thread = threading.Thread(target=self.process_graph,
args=(G,))
                processing_thread.start()
            except Exception as e:
                messagebox.showerror("Помилка", f"Помилка завантаження графа: {e}")

    def load_graph(self):
        file_path = filedialog.askopenfilename(filetypes=[("Text files", "*.txt"),
("All files", "*.*")])
        if file_path:
            try:
                with open(file_path, 'r') as file:
                    lines = file.readlines()
                    edges = [tuple(map(int, line.strip().split())) for line in
lines]
                G = nx.Graph(edges)
                communities_label, communities_cnm = self.detect_communities(G)
                edge_weights = self.get_edge_weights(G)
                graph_info = self.graph_info(G)
                top_level_comm, threats, strong_comms, weak_comms =
self.detect_threats(G)
                self.display_results(top_level_comm, threats, strong_comms,
weak_comms,
                                     communities_label, communities_cnm,
edge_weights, graph_info)
                self.draw_graph(G, top_level_comm)
                self.save_results(communities_label, communities_cnm, edge_weights,
graph_info)
            except Exception as e:
                messagebox.showerror("Помилка", f"Помилка завантаження графа: {e}")
    def calculate_clustering_coefficient(self, graph):
        return nx.average_clustering(graph)
    def detect_communities(self, graph):
        communities_label =
list(nx.algorithms.community.label_propagation.label_propagation_communities(graph))
        communities_cnm =
list(nx.algorithms.community.modularity_max.greedy_modularity_communities(graph))
        return communities_label, communities_cnm

    def get_edge_weights(self, graph):
        edge_weights = nx.get_edge_attributes(graph, 'weight')
        return edge_weights if edge_weights else {}  # Перевірка на наявність ваг
ребер
```

```
    def graph_info(self, graph):
        info = f"Кількість вузлів: {len(graph.nodes)}\n"
        info += f"Кількість ребер: {len(graph.edges)}\n"
        info += f"Список вузлів: {list(graph.nodes)}\n"
        info += f"Список ребер: {list(graph.edges)}\n"
        return info

    def save_results(self, communities_label, communities_cnm, edge_weights,
graph_info):
        with open("results.txt", "w") as file:
            file.write("Спільноти Label Propagation:\n")
            for idx, comm in enumerate(communities_label):
                file.write(f"Спільнота {idx + 1}: {', '.join(map(str, comm))}\n")

            file.write("\nСпільноти CNM:\n")
            for idx, comm in enumerate(communities_cnm):
                file.write(f"Спільнота {idx + 1}: {', '.join(map(str, comm))}\n")

            file.write("\nВаги ребер:\n")
            for edge, weight in edge_weights.items():
                file.write(f"{edge}: {weight}\n")

            file.write("\nІнформація про граф:\n")
            file.write(graph_info)

    def detect_threats(self, graph):
        communities_gen = nx.algorithms.community.girvan_newman(graph)
        top_level_comm = [set(comm) for comm in next(communities_gen)]
        threats = []

        for node in graph.nodes():
            in_communities = [comm for comm in top_level_comm if node in comm]
            internal_links = sum(1 for comm in in_communities if node in comm)
            external_links = len(graph[node]) - internal_links
            if internal_links > external_links:
                threats.append(node)

        strong_communities = self.strong_communities(graph, top_level_comm)
        weak_communities = self.weak_communities(graph, top_level_comm)

        return top_level_comm, threats, strong_communities, weak_communities

    def strong_communities(self, graph, communities):
        strong_comms = []
        for comm in communities:
            internal_links = sum(graph.subgraph(comm).degree(n) for n in comm)
            external_links = sum(len(graph.subgraph(comm).edges(n)) for n in comm)
            if internal_links >= external_links:  # Міцна спільнота: >= замість >
                strong_comms.append(comm)
        return strong_comms

    def weak_communities(self, graph, communities):
        weak_comms = []
        for comm in communities:
            internal_links = sum(graph.subgraph(comm).degree(n) for n in comm)
            external_links = sum(len(graph.subgraph(comm).edges(n)) for n in comm)
            if internal_links <= external_links:  # Слабка спільнота: <= замість <
                weak_comms.append(comm)
        return weak_comms

    def display_results(self, communities, threats, strong_communities,
weak_communities,
```

```
                        communities_label, communities_cnm, edge_weights,
graph_info):
        community_str = "\n".join(
            [f"Спільнота {i + 1}: {', '.join(map(str, comm))}" for i, comm in
enumerate(communities)])
        threats_str = f"Загрози вузлів: {', '.join(map(str, threats))}"
        strong_communities_str = "\n".join(
            [f"Міцна спільнота {i + 1}: {', '.join(map(str, comm))}" for i, comm in
enumerate(strong_communities)])
        weak_communities_str = "\n".join(
            [f"Слабка спільнота {i + 1}: {', '.join(map(str, comm))}" for i, comm in
enumerate(weak_communities)])

        communities_label_str = "\n".join(
            [f"Спільнота Label Propagation {i + 1}: {', '.join(map(str, comm))}" for
i, comm in
             enumerate(communities_label)])
        communities_cnm_str = "\n".join(
            [f"Спільнота CNM {i + 1}: {', '.join(map(str, comm))}" for i, comm in
enumerate(communities_cnm)])

        edge_weights_str = "\n".join([f"Ребро {edge}: Вага {weight}" for edge,
weight in edge_weights.items()])
        edge_info = f"Ваги ребер:\n{edge_weights_str}\n\n"
        result_text = f"Спільноти:\n{community_str}\n\n{threats_str}\n\nМіцні
спільноти:\n{strong_communities_str}\n\nСлабкі
спільноти:\n{weak_communities_str}\n\n"
        result_text += f"Спільноти Label Propagation:\n{communities_label_str}\n\n"
        result_text += f"Спільноти CNM:\n{communities_cnm_str}\n\n"
        result_text += edge_info
        result_text += f"Інформація про граф:\n{graph_info}\n"
        result_text += f"Кластерний коф:\n{clustering_coefficient}\n"

        self.result_label.config(text=result_text)

    def draw_graph(self, graph, communities):
        plt.figure(figsize=(8, 6))
        pos = nx.spring_layout(graph)
        colors = ['r', 'g', 'b', 'y', 'c', 'm']

        for i, comm in enumerate(communities):
            nodes = list(comm)
            nx.draw_networkx_nodes(graph, pos, nodelist=nodes, node_color=colors[i],
node_size=300, alpha=0.8)

        nx.draw_networkx_edges(graph, pos, width=0.5, alpha=0.5)
        nx.draw_networkx_labels(graph, pos, font_size=8, font_color='black')  #
Відобразити позначення вузлів
        plt.title('Граф спільнот')
        plt.axis('off')

        # Convert Matplotlib figure to Tkinter-compatible canvas and display
        canvas = FigureCanvasTkAgg(plt.gcf(), master=self.canvas)
        canvas.draw()
        canvas.get_tk_widget().pack(side=tk.BOTTOM, fill=tk.BOTH, expand=1)

def main():
    root = tk.Tk()
    app = CyberThreatDetectionApp(root)
    root.mainloop()

if __name__ == "__main__":
    main()
```

Додаток В

ІЛЮСТРАТИВНА ЧАСТИНА

СИСТЕМА ВИЯВЛЕННЯ КІБЕРЗАГРОЗ, ЩО СТВОРЮЮТЬСЯ СПІЛЬНОТАМИ В СОЦІАЛЬНИХ МЕРЕЖАХ. ЧАСТИНА 1. МОДУЛЬ ВИЗНАЧЕННЯ СПІЛЬНОТ З ВИКОРИСТАННЯМ КОМПОНОВКИ В ГРАФАХ

Актуальність, мета та задачі МКР

Актуальність

Злочинці, використовуючи соціальні мережі, здатні поширювати дезінформацію, проводити кібератаки, зламувати особисту інформацію та використовувати ці можливості для своїх цілей. Актуальність даної роботи полягає в необхідності розробки модуля визначення спільнот для системи яка здатна виявляти та запобігати цим кіберзагрозам, зокрема тим, які формуються в межах спільнот соціальних мереж.

Предмет та об’єкт дослідження

Предмет – моделі визначення спільнот за допомогою компонування в графах та коефіцієнта кластеризації.
Об'єкт – процес виявлення спільнот для аналізу кіберзагроз в соціальних мережах.

Мета

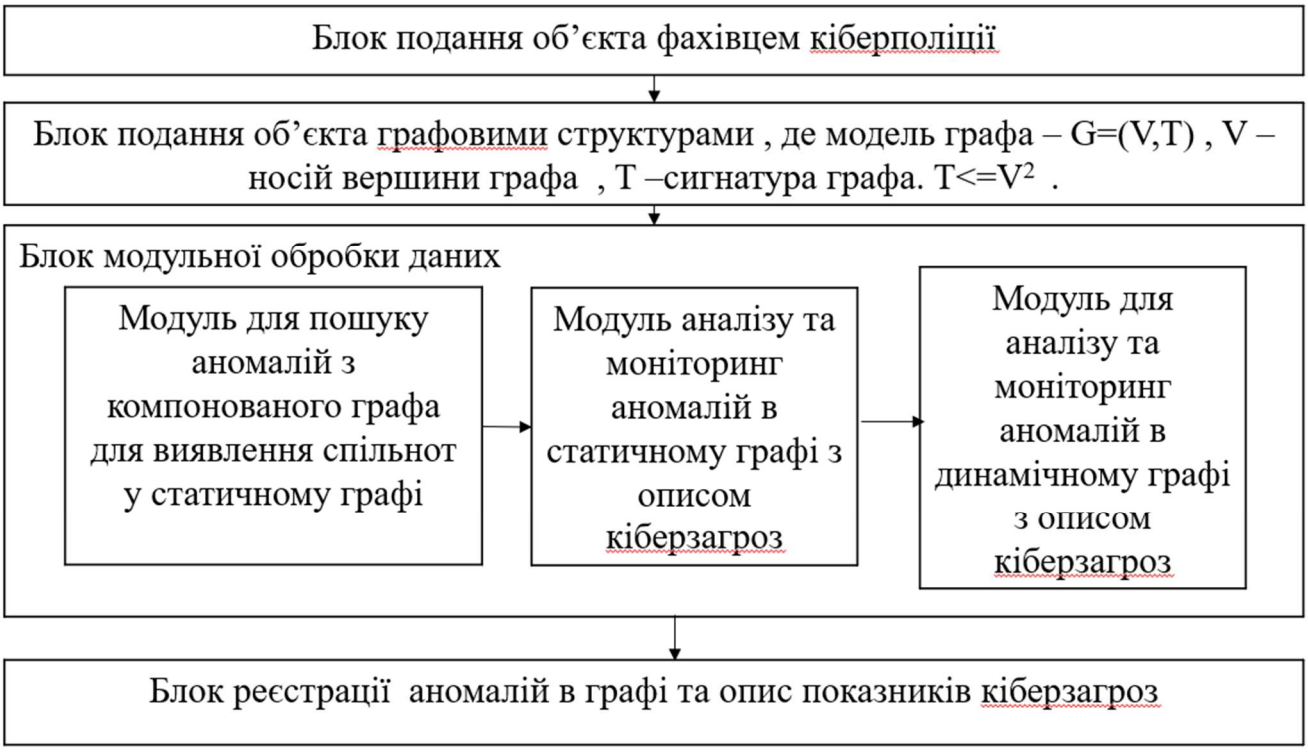
Покращення можливостей для аналізу кіберзагроз в соціальних мережах шляхом виявлення спільнот

2

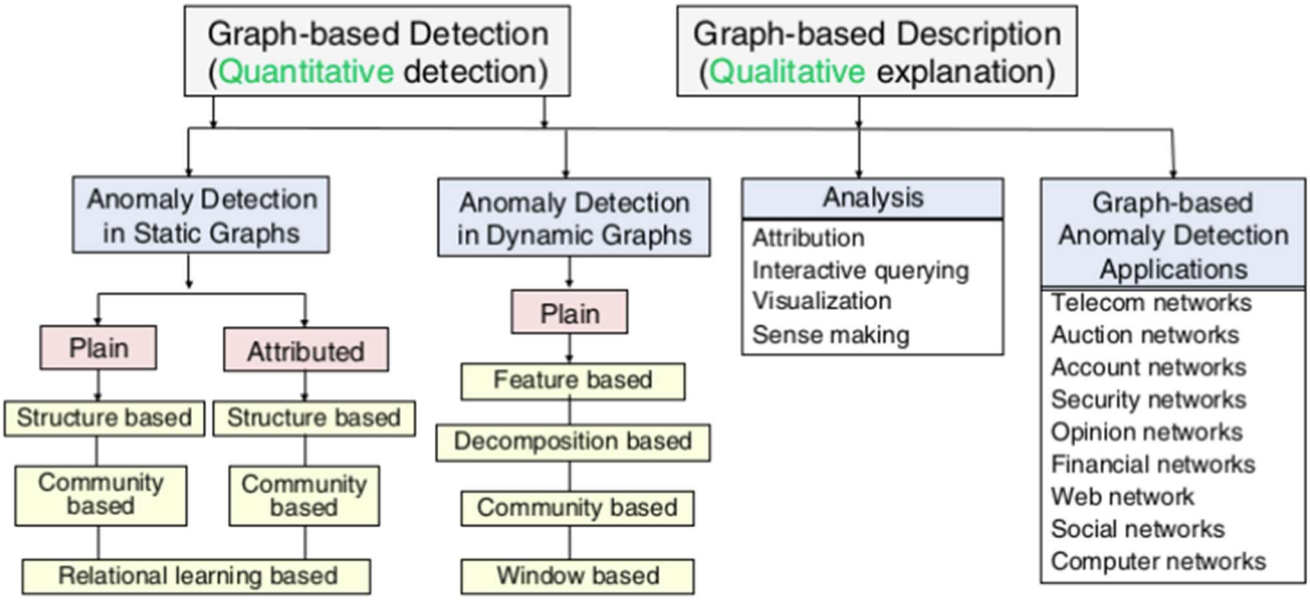
Методи визначення та аналізу спільнот в соціальних мережах

Методи визначення та аналізу спільнот в соціальних мережах		
Підхід до виявлення спільнот	Математичні моделі для виявлення	Способи розрахунків показників виявлення спільнот
1. Методи компонування на основі графів. Підхід ґрунтується на розділенні графа на окремі компоненти зв'язності. Розрізання графа на компоненти зв'язності дозволяє визначити ізольовані групи вершин, які можуть утворювати потенційні спільноти в мережі.	Математична модель - виділення субграфів (кластерів) у великому графі. Це досягається шляхом виявлення розрізу в графі, який розділяє вершини на різні групи. Ці субграфи аналізуються на предмет їх структури та взаємозв'язків між вершинами.	1.Теорія множин для визначення зв'язків між вершинами у графі та їх групування. Наприклад, визначення групи вершин, які мають спільних сусідів. 2.Алгоритм Лювена для виявлення спільнот на основі оптимізації модулярності графа. 3.Порівняння внутрішніх зв'язків між вершинами групи з їх зовнішніми зв'язками з рештою графа. Спільнота вважається, якщо кількість зв'язків між вершинами у групі більше, ніж кількість зв'язків цих вершин з іншими вершинами в графі. 4.Коефіцієнт кластеризації обчислюється за допомогою алгоритму Girvan-Newman (GN), який обчислює міру центральності ребер у графі. Міра центральності ребра вказує на те, як сильно воно сполучає вершини та може бути використане шахраями або експертами для знаходження ребер, які мають важливість для спільноти. 5.Коефіцієнт кластеризації це середня ступінь кластеризації для всіх вершин у графі. Вимірює, як часто вершини утворюють з'єднання з іншими вершинами в своєму ж графічному околі.
7		
2. Метод k-середніх. Розділяє вершини графа на k кластерів, де кожен кластер представлений центроїдом. Центроїд є "середнім" представником вершин у кластері та визначається як середнє арифметичне координат вершин у кластері.	Математична модель – мінімізація внутрішньокластерної дисперсії. Для кожного кластера обчислюється сума відстаней між кожною вершиною та центроїдом кластера, і ця сума мінімізується.	1.Для кожної вершини обчислюється її відстань до центроїду кожного кластера. Кожна вершина призначається до кластера з найменшою відстанню. Метод базується на кластеризації вершин у k кластерів, критерій визначення спільноти може бути виражений через принципи кластеризації. 2.Кластеризація k-середніх використовує міру інерції (inertia) або внутрішньокластерний показник розкиду. Це сума квадратів відстаней між кожною точкою та центроїдом її кластера. Кластеризація намагається мінімізувати це значення.
3. **Ієрархічна кластеризація.** Створює ієрархію кластерів у вигляді дерева. Кожен вузол дерева представляє кластер, а злиття або розбиття кластерів відбувається на різних рівнях ієрархії. Цей метод може бути агломеративним (починає з окремих вершин і об'єднує їх) або дізгломеративним (починає з усіх вершин і розділяє їх).	Математичні моделі для обчислення подібності між вершинами та групами вершин на різних рівнях ієрархії. Різноманітні метрики подібності, такі як відстань між кластерами чи кореляція між їх характеристиками.	1.Методи об'єднання або розбиття, які базуються на метриках відстаней або подібності між вершинами або групами вершин. Різні метрики для визначення спільнот. Одним із можливих критеріїв це зростання внутрішньокластерних зв'язків порівняно зі зв'язками між кластерами. 2.Алгоритм GN для обчислення коефіцієнта кластеризації, щоб визначити важливість ребер для спільнот. Коефіцієнт кластеризації в ієрархічній кластеризації це середнє значення міри кластеризації для різних рівнів ієрархії, якщо існують різні рівні об'єднання кластерів.
8		

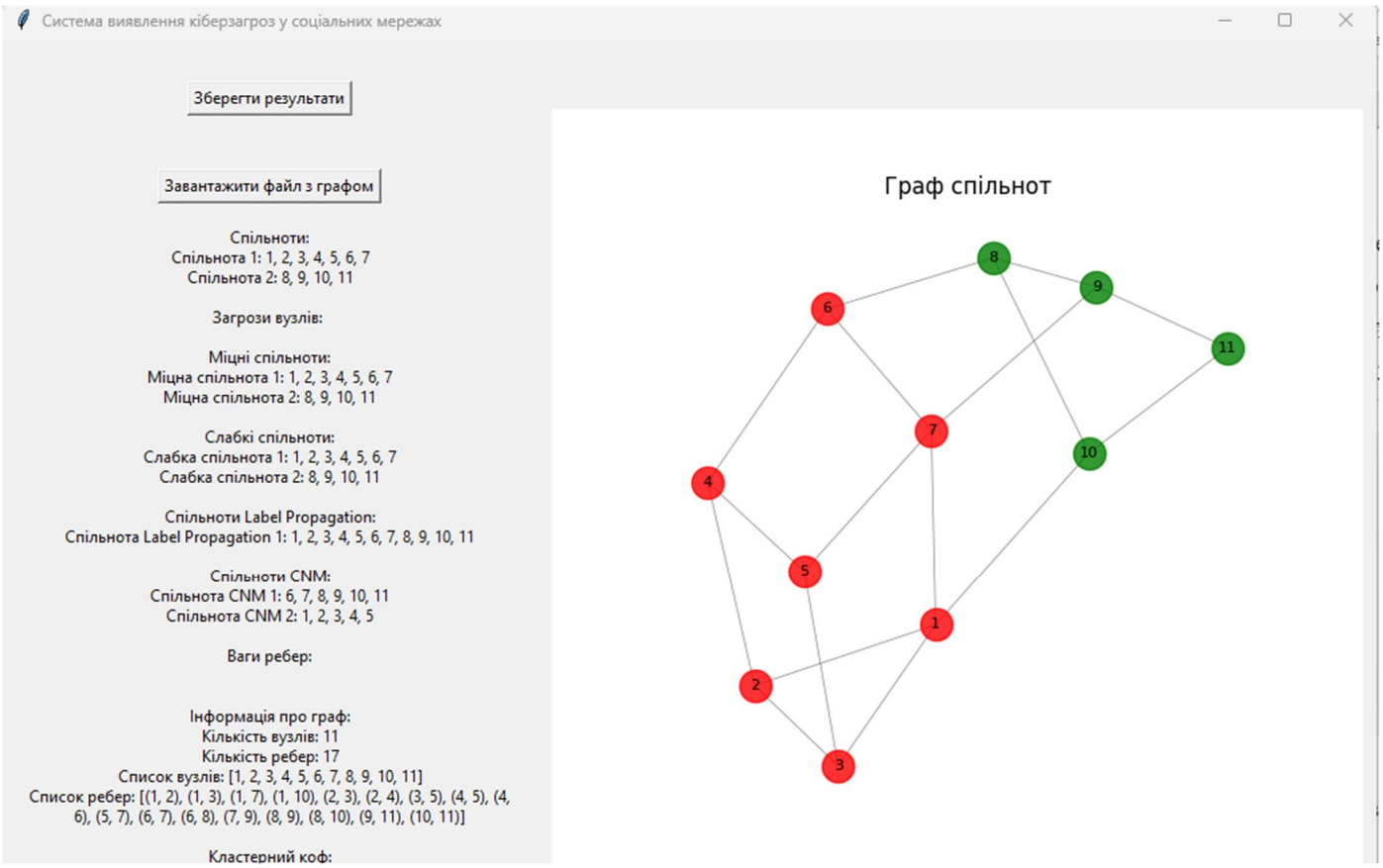
Система виявлення кіберзагроз, що утворюються спільнотами соціальних мереж на основі теорії графів



Сучасні підходи до виявлення аномалій в графових структурах

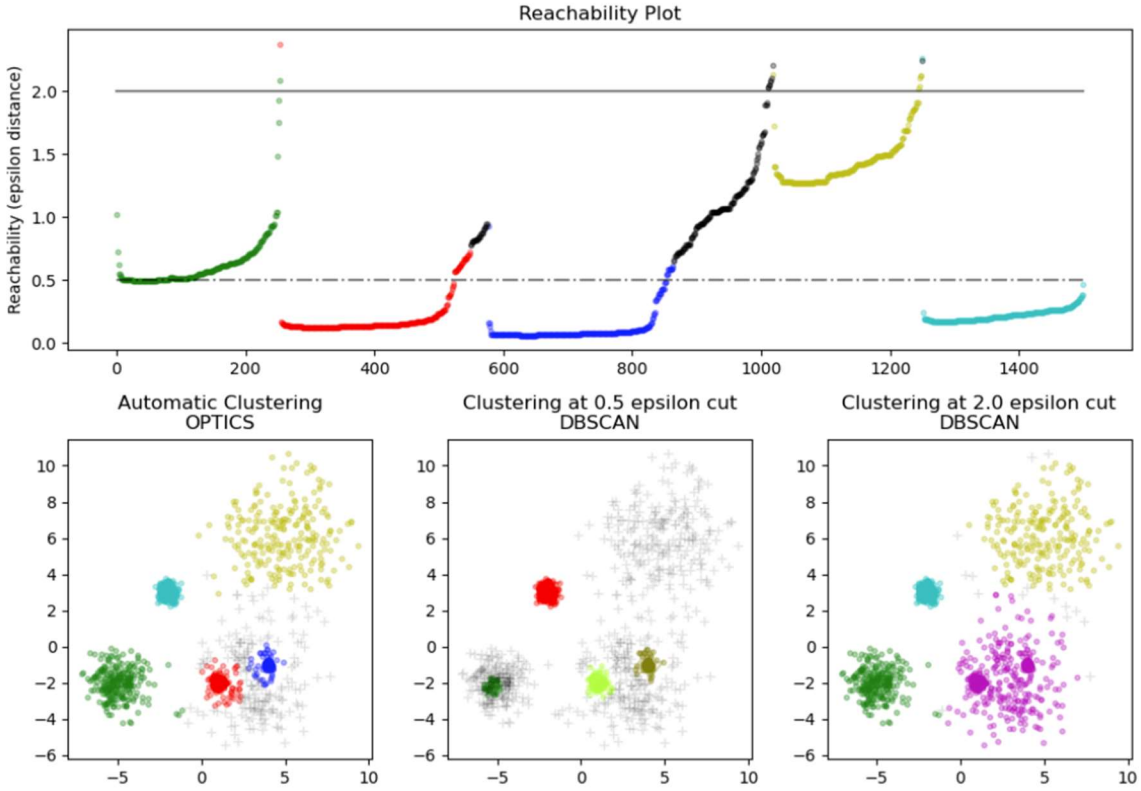


Інтерфейс програмного засобу

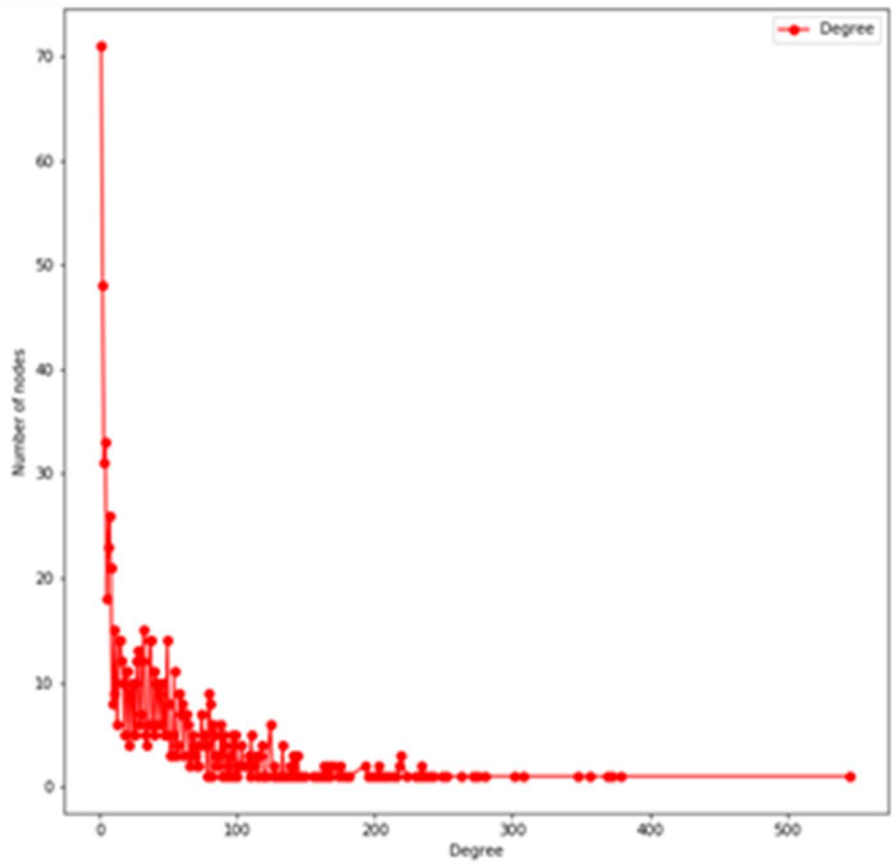


Алгоритм роботи методу k-середніх

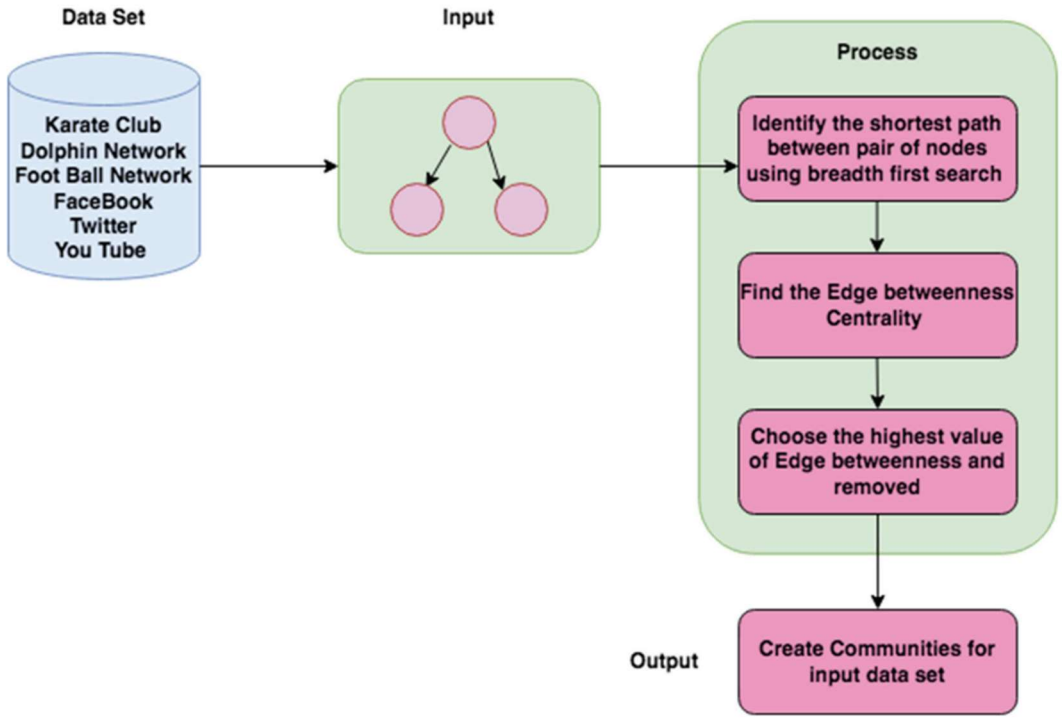
Алгоритм OPTICS



Розподіл ступенів вузлів



Архітектура алгоритму GN



Алгоритм роботи методу k-means++