**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**
**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**
**імені ІГОРЯ СІКОРСЬКОГО»**
**НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ**
**Кафедра інформаційної безпеки**

До захисту допущено

Завідувач кафедри
_____ **Дмитро ЛАНДЕ**

(підпис)

« 20 » червня 2025 р.

# Дипломна робота

**на здобуття ступеня бакалавра**

**за освітньо-професійною програмою «Системи, технології та математичні методи кібербезпеки»**
**спеціальності 125 «Кібербезпека»**

на тему: методи візуалізації та аналізу мережевих зв'язків для виявлення підозрілих акаунтів у соціальних мережах

Виконав (-ла): здобувач вищої освіти **IV** курсу, групи ФБ-12

Кузнєцов Олексій Миколайович

(прізвище, ім'я, по батькові) (підпис)

Керівник роботи: Полуциганова Вікторія Ігорівна д. філос., асистент кафедри інформаційної безпеки

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові) (підпис)

Рецензент: Терещенко Іван Миколайович доцент, к.ф.-м.н

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище, ім'я, по батькові) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Здобувач вищої освіти _____
(підпис)

Київ – 2025 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**
**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**
**імені ІГОРЯ СІКОРСЬКОГО»**
**НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ**
Кафедра інформаційної безпеки

Рівень вищої освіти – перший (бакалаврський)
Спеціальність – 125 «Кібербезпека»
Освітньо-професійна програма «Системи, технології та математичні методи кібербезпеки»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Дмитро ЛАНДЕ
(підпис)
« 20 »_червня____2025 р.

**ЗАВДАННЯ**
**на дипломну роботу здобувачу вищої освіти**

Кузнєцову Олексію Миколайовичу
(прізвище, ім’я, по батькові)

1. Тема роботи. Методи візуалізації та аналізу мережевих зв'язків для виявлення підозрілих акаунтів у соціальних мережах; ,

керівник роботи Полуциганова Вікторія Ігорівна д. філос., асистент кафедри інформаційної безпеки; ,
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від « 26 » травня 2025 р. № 1761-с

2. Термін подання роботи здобувачем вищої освіти «13» червня 2025 р.

3. Вихідні дані до роботи: Попередні дослідження та відомості за темою роботи;

4. Зміст роботи: Теоретичні відомості про соціальні мережі, типи акаунтів, підозрілі акаунти, ознаки підозрілих акаунтів; побудова алгоритму встановлення ступеню

підозрілості акаунта; побудова соціального графа; розробка власного програмного забезпечення для автоматизованого аналізу;

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): презентація;

6. Дата видачі завдання: 12.09.2024;

Календарний план

| № з/п | Назва етапів виконання дипломної роботи | Термін виконання етапів дипломної роботи | Примітка |
|---|---|---|---|
| 1 | Отримання завдання для дипломної роботи | 12.09.2024 | Виконав |
| 2 | Огляд літературних джерел | 01.03.2025 - 09.03.2025 | Виконав |
| 3 | Робота над першим розділом | 10.03.2025 - 23.03.2025 | Виконав |
| 4 | Визначення характерних ознак підозрілих акаунтів | 24.03.2025 - 29.03.2025 | Виконав |
| 5 | Робота над другим розділом | 30.03.2025 - 13.04.2025 | Виконав |
| 6 | Проходження переддипломної практики | 14.04.2025 - 18.05.2025 | Виконав |
| 7 | Розробка програмного забезпечення та тестування програми | 19.05.2025 - 26.05.2025 | Виконав |
| 8 | Робота над третім розділом | 26.05.2025 - 07.06.2025 | Виконав |
| 9 | Створення презентації для передзахисту диплому | 08.06.2025 - 12.06.2025 | Виконав |
| 10 | Передзахит дипломної роботи | 13.06.2025 | Виконав |
| 11 | Внесення остаточних правок | 14.06.2025 - 19.06.2025 | Виконав |
| 12 | Захист дипломної роботи | 20.06.2025 | Виконав |

Здобувач вищої освіти _____ Олексій Кузнєцов
(підпис) **(Власне ім'я, ПРІЗВИЩЕ)**

Керівник роботи _____ Вікторія Полуциганова
(підпис) **(Власне ім'я, ПРІЗВИЩЕ)**

# РЕФЕРАТ

Дипломна робота має обсяг 74 сторінок; містить 21 рисунок, 6 таблиць, 5 формул, 26 джерел та 1 додаток.

Метою роботи є розробка програмного забезпечення для автоматизованого аналізу акаунтів користувачів у соціальних мережах із використанням методів експертного аналізу та машинного навчання. У роботі розглянуто актуальні виклики кібербезпеки, зумовлені поширенням фейкових акаунтів, бот-мереж і дезінформації. Проведено аналіз характерних ознак підозрілої активності, досліджено існуючі методи виявлення таких акаунтів, запропоновано власний підхід, розроблено та протестовано відповідне програмне забезпечення.

Основу методу становить побудова соціального графа на основі обчислених показників підозрілості та візуалізація мережевих зв’язків для подальшого аналізу. Отримані результати мають практичну цінність і можуть бути використані в системах моніторингу соцмереж, а також інтегровані в інструменти кіберзахисту.

Результати роботи апробовані на XXIII Всеукраїнській науково-практичній конференції студентів, аспірантів та молодих вчених, опубліковані у збірнику матеріалів конференції.

**Ключові слова: соціальні мережі, підозрілі акаунти, соціальний граф, машинне навчання, візуалізація зв’язків, метод попарних порівнянь**

# ABSTRACT

The thesis has a volume of 74 pages; contains 21 figures, 6 tables, 5 formulas, 26 sources and 1 appendix.

The purpose of the work is to develop software for automated analysis of user accounts in social networks using expert analysis and machine learning methods. The work considers current cybersecurity challenges caused by the spread of fake accounts, botnets and disinformation. The characteristic features of suspicious activity were analyzed, existing methods for detecting such accounts were investigated, an own approach was proposed, and appropriate software was developed and tested.

The basis of the method is the construction of a social graph based on calculated indicators of suspicion and visualization of network connections for further analysis. The results obtained have practical value and can be used in social network monitoring systems, as well as integrated into cyber security tools.

The results of the work were tested at the XXIII All-Ukrainian Scientific and Practical Conference of Students, Postgraduate Students and Young Scientists, and published in the conference proceedings.

**Keywords: social networks, suspicious accounts, social graph, machine learning, visualization of connections, pairwise comparison method**

# ЗМІСТ

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ІПСО – інформаційно-психологічна операція;

ПЗ – програмне забезпечення;

Q/A – Question and Answer, запитання та відповіді (формат форумів);

C&C – Command and Control, система управління ботнетом;

ММН – методи машинного навчання;

ML – Machine Learning, машинне навчання;

Ratio – співвідношення кількості друзів до кількості підписників;

Бот – автоматизований акаунт у соціальній мережі;

Кіборг – акаунт, який частково керується людиною, частково автоматизований;

Фейковий акаунт – це обліковий запис, створений для обману людей і видавання себе за когось чи щось. Такі облікові записи часто використовуються для кіберзалякування, онлайн-шахрайства та політичних цілей;

Нормалізація – процес приведення значень ознак до єдиного масштабу;

Вага критерію – числове значення, що відображає важливість ознаки в моделі;

# ВСТУП

З появи перших соціальних мереж у 1995 році ці платформи еволюціонували з нішових веб-сайтів до невід'ємної частини повсякденного життя мільйонів людей. Соціальні мережі дозволяють миттєвий обмін повідомленнями, фото та відео котнетном з людьми в будь якій точці світу, що сприяє налагодженню нових знайомств, підтримувати зв'язок та поширювати інформацію. Поширеність соціальність мереж зробила їх потужним інтрументом соціального зв'язку.

Однак, разом з позитивним впливом, соціальні мережі створили нові виклики в сфері інформаційної безпеки. Вони можуть використовуватися для фішингу, поширення фейків, торгівлі забороненими речами, зміни суспільної думки тощо. Завдяки функціоналу платформ, зловмисники можуть створювати мережі фейкових акаунтів чи мереж ботів, які в подальшому використовуються для недоброчесних дій.

У цьому контексті виявлення підозрілих акаунтів  та аналіз зв'язків між ними є важливою задачою кібербезпеки, зокрема у сфері мережевої безпеки.  Основою для дослідження є публічно доступні дані користувачів, які довзолять нам проаналізувати ознаки підозрілих акаунтів та побудвати соціальний граф мережі.

**Актуальність роботи**

Актуальність теми обумовлена поточоним військовим станом, у рамках якого активно розповсюджені методи маніпуляції, дезінформації та ІПСО в інтернет просторі[1].  Більшість соціальних мереж тримають свої розробки в таємниці, що не дає змоги звичайним користувачам, або ж спеціалізованим компаніям використовувати їх. Запропонований у роботі метод надає відкритий та адаптивний інструмент, що вирішує цю проблему.

**Мета дослідження** – розробити програмне забезпечення для автоматизованого аналізу акаунтів користувачів у соціальних мережах; вивчення методів візуалізації мережевих зв’язів.

*Об’єктом дослідження* виступають підозрілі акаунти в соціальних мережах

*Предмет дослідження* виступають мережеві зв’язки між користувачами та характерні ознаки притаманні підозрілим акаунам.

**Завдання дослідження:**

1. Провести аналіз акаунтів та виявити ознаки, що можуть свідчити про підозрілу активність користувача;
2. Дослідити існуючі підходи з вивлення підозрілих акаунтів;
3. Розробити власний метод на основі аналізу існуючих підходів з метою позбутися їх недоліків та описати алгоритм його роботи;
4. Розробити алгоритм аналізу даних;
5. Реалізувати програмне забезпечення, протестувати його та проаналізувати результати;
6. Зробити висновки;

**Методи дослідження:**

Аналіз літературних джерел, вивчення теорії графів, вивчення методів машинного навчання, моделювання соціальної активності.

**Практичне значення одержаних результатів:**

Результати дослідження можуть бути використані в системах моніторингу соціальних мереж. Розроблений прототип програми може бути інтегрований в сучасні інструменти кібербезпеки, аналітики даних, маркетингу тощо. Методика може бути використана державними структурами, або приватними компаніями для здійснення інформаційної безпеки.

**Апробація результатів роботи**

Результати роботи доповідалися на XXIII Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики» 2025 року

**Публікації**

Робота була опублікова у збірнику матеріалів XXIII Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики» 2025 року[1].

# 1 СОЦІАЛЬНІ МЕРЕЖІ ТА ПІДОЗРІЛІ АКАУНТИ

## 1.1 Соціальні мережі

У сучасному цифровому світі соціальні мережі стали не лише інструментом комунікації, а й важливою складовою інформаційного простору. Їх значення зросло настільки, що вони впливають на суспільну думку, економіку, політику, а також використовуються як платформи для інформаційно-психологічних операцій. В умовах збройного конфлікту, гібридних загроз та масових кампаній дезінформації, аналіз структури соціальних мереж і поведінки користувачів у них набуває критичного значення для забезпечення кібербезпеки.

Соціальні мережі можуть бути використані зловмисниками для створення бот-мереж, фейкових акаунтів, маніпулювання громадською думкою та поширення шкідливої інформації. Саме тому дослідження їхньої природи, типів акаунтів та зв’язків між ними є надзвичайно актуальним у контексті виявлення підозрілих облікових записів та підвищення рівня захисту інформаційного середовища.

Соціальні мережі – це онлайн-платформи, де користувачі можуть ділитися інформацією та спілкуватися з віртуальними спільнотами за допомогою тексту, відео, фотографій та іншого контенту[3].

Основною метою  соціальних мереж: надати користувачам зручний функціонал для обміну текстовими повідомленнями та медіаконтентом (фото, відео, музика, тощо), можливість створювати онлайн групи по інтересам та просто пошук нових знайомих/друзів. Зі своєї ж сторони, взаємодіючи з платформою, користувач переглядає рекламні оголошення. Тобто чим більше часу людина проводить у соціальній мережі, тим більше прибутку отримає компанія. Саме тому компаніям вигідно постійно покращувати свою платформус для того, щоб утримати увагу користувач на довше.

Соціальні мережі стали невід'ємною часиною життя сучасної людини. Згідно статистики DataReportal – Global Digital Insights[3] за 2024 рік понад 5.04 мільярдів людей, що становить 62.3% населення,  активно користуються соціальними платформами.

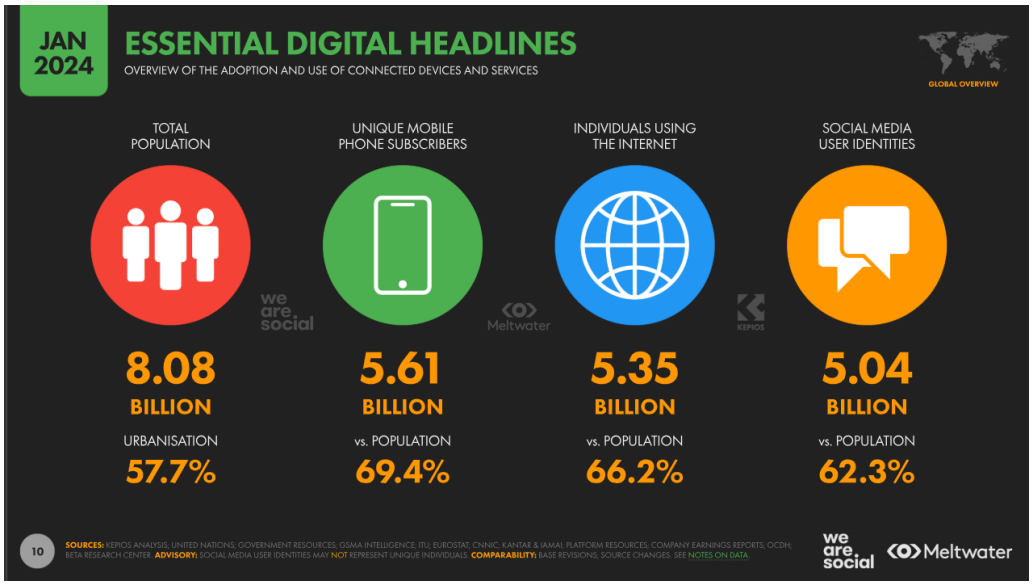


Рисунок 1.1 – Порівняння всього населення землі з інтернет активність

Згідно статистики станом на квітень 2025 у світі існує сім соціальних мереж, кожна з яких заявляє про 1 мільярд користувачів, або більше[5]. У цьому можемо переконатися, переглянувши топ 10 платформ за кількістю активних користувачів:

1. Facebook: 3,07 мільярда користувачів активних користувачів щомісяця
2. YouTube: Потенційне охоплення реклами становить 2,54 мільярда(а)
3. WhatsApp: 2 мільярди активних користувачів щомісяця
4. Instagram: 2 мільярди активних користувачів щомісяця
5. TikTok: Реклама потенційно охоплює 1,84 мільярда дорослих старше 18 років
6. WeChat: 1,39 мільярда активних користувачів щомісяця
7. Telegram: 1 мільярд активних користувачів щомісяця
8. Messenger: Потенційне охоплення реклами становить 965 мільйонів
9. Snapchat: 850 мільйонів активних користувачів щомісяця
10. Douyin: 770 мільйонів активних користувачів щомісяця

У цей же час база користувачів соціальних мереж не стоїть на одному місці, а щороку лиш збільшується на мільйони нових облікових записів. Ми можемо впевнитися у цьому переглянувши рисунок 1.2.

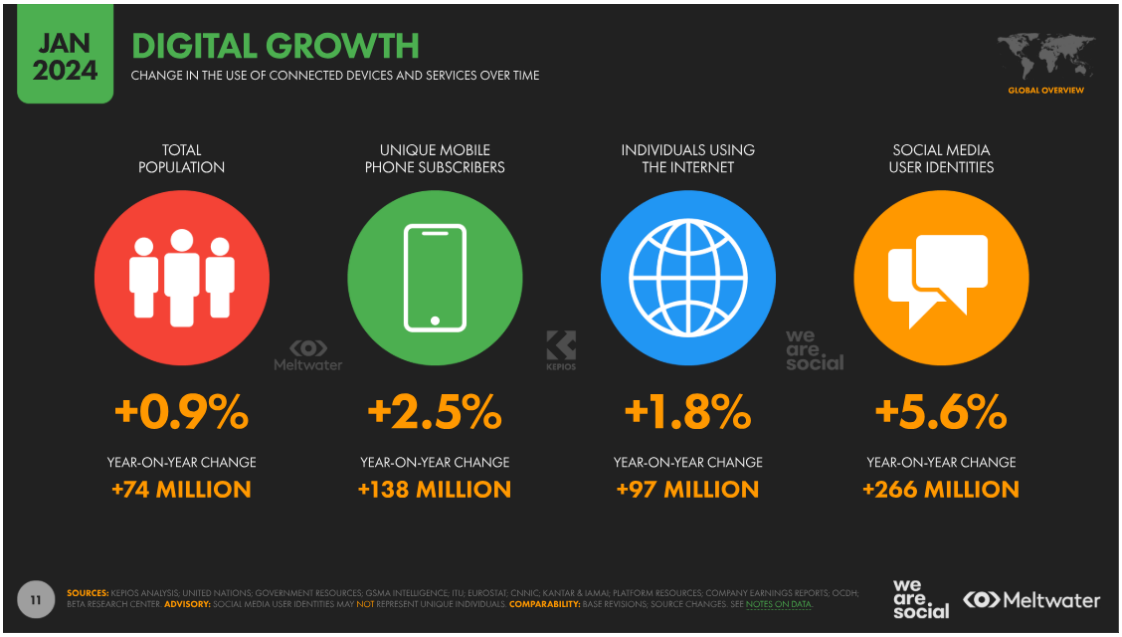


Рисунок 1.2 – Статистика приросту нових користувачів

Зростання популярності зумовлене активним розвитком цифрових технологій, легкістю комунікації (для взаємодії з колегами/друзями/родичами), можливістю самовираження та шансом отримувати прибуток за свою творчість.

Соціальні мережі це загальний термін, що описує всі типи різних соціальних платформ, але хотілось би виділити основні з них:

1. Загального призначення (унвіерсальні): До таких плафтформ відносять, всім відомі, Facebook, Twitter. Ці платформи орієнтовані на широку аудиторію. Вони поєднують у собі безліч функцій: спілкування, піблікації, новини, групи за інтересами фото/відео контент тощо. Кожен користувач зможе знайти щось для себе;
2. Професійні соціальні мережі: Цей тип орієнтований на ділове середовище. Основна мета – допомоготи користувачам, які готові налагоджувати

професійні зв’язки, шукають роботу чи співробітників. Яскравим представником є LinkedIn;

3. Мережі для обміну мультимедіа: Ці платформи мають більшу спрямованість на обмін звуко/відео контенту між користувачами. Платформи мають багато функціоналу зав’язаному на цьому: стрічки контенту, фільтри та вмонтовані відеоредактори. Відомі приклади: Instagram, Snapchat, TikTok;
4. Форуми та платформи Q/A: Сервіси спрямовані на тематичні обговорення, що дозволяє обговорювати улюблені речі. Користувачі можуть ставити запитання, відповідати на них, вести дискусії. Популярним на заході сервісом є Reddit;
5. Месенджери: Соціальні платформи, що зосереджені на особистому спілкування з людьми чи групами. Популярні у нас Telegram та Viber тому приклад;

Знання класифікації типів соціальних мереж є важливим для нашого дослідження, адже кожен тип платформи формує унікальну модель взаємодії між користувачами. Це безпосередньо впливає на формування соціальних графів, структуру мережевих зв’язків та визначення ознак підозрілої активності, що є ключовими елементами запропонованого підходу до виявлення підозрілих акаунтів.

### 1.2 Визначення різних типів акаунтів

Згідно з авторами статті[6], у якій дослідили групу з понад 100 000 користувачів платформи Twitter та класифікували їхні ролі на основі співвідношення кількості підписників до кількості послідовників, користувачі Twitter поділяються на три групи: “телеведучі” (в оригіналі broadcasters) , знайомі та зловмисники. Трохи детальніше про ці групи:

- “Телеведучі”: Користувачі, що  мають набагато більшу кількість підписників, ніж вони самі підписані. Багато з цих користувачів представляють відомих осіб, інтернет-магазини, великі організації, які використовують соціальні

мережі для просування своїх продуктів і безпосередньої взаємодії зі своєю цільовою аудиторією;

- Знайомі: Група користувачів, яких називають знайомими, схильна демонструвати взаємність у своїх стосунках, що є загальною характеристикою онлайн-соціальних мереж. Загалом, це реальні користувачі, які зв'язуються з друзями та родиною або ж стежать за людьми з груп розсилки;
- Зловмисники: Спільною рисою користувачів, що входять до третьої унікальної групи, є те, що вони підписані на значно більшу кількість людей, ніж мають підписників. Така поведінка типова для спамерів або людей, які активно пропагують свої переконання. Хоча це одна з найпідозріліших груп, вона не виключає можливості існування справжніх користувачів з багатьма інтересами чи вподобаннями.

Названі вище групи можуть бути представлені з допомогою діаграми розсіювання на рисунку 1.3. Відповідно бачимо три групи телеведучі, знайомі зловмисники в порядку зліва на право.

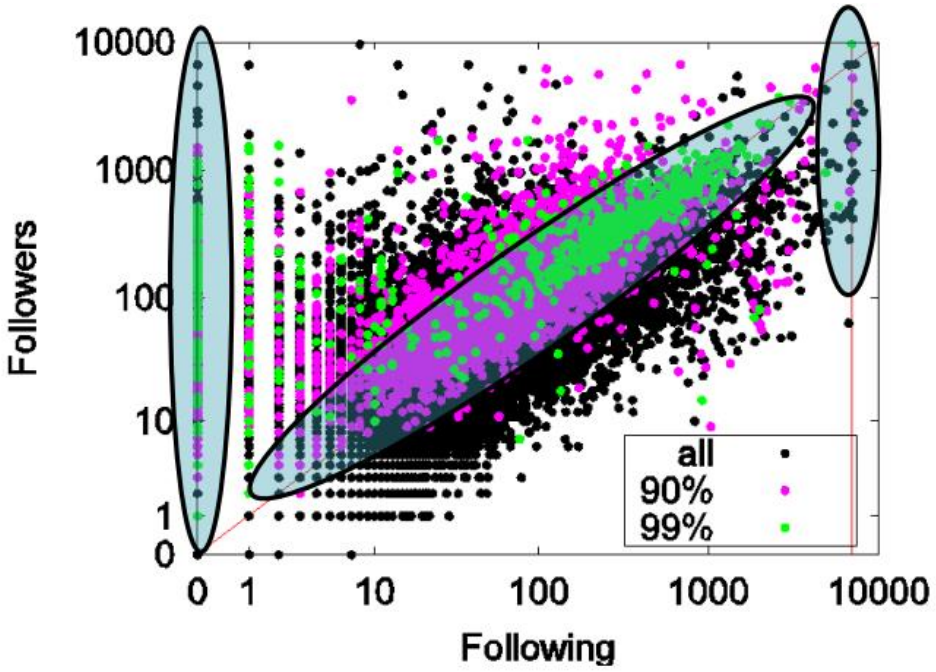


Рисунок 1.3 – Діаграма зв’язків з користувачами

Запропонована класифікація користувачів являє собою корисну базу знань про типові облікові записи, що зустрічаються на просторах соціальних мереж. Однак така класифікація нам не підходить, оскільки ми хочемо сформувати повне розуміння поведінки підозрілих акаунтів. До того ж, ця класифікація заснована лише на одній характеристиці облікового запису, що не може бути достатнім для подальшого аналізу.

В іншому дослідженні[7], яка надає нам більш глибокий аналіз користувачів пропонує таку класифікацію: люди, боти, кіборги. Для кращого розуміння теми, коротко охарактеризувавши кожен тип:

- Людина: Користувач профіль якого містить автентичний, змістовний та конкретний контент. Конкретність означає, що вміст твіту представлений відносно простими словами з усвідомленням, наприклад, відповідь на запитання є релевантною та безпосередньо стосується його теми;
- Боти: Користувачі, які демонструють брак людського інтелекту або оригінального контенту, відносяться до групи ботів; нескінченно пересилають чужі пости та публікація; надмірна автоматизація; пересилання шкідливих URL; агресивна поведінка;

  Основними задачами таких ботів, являє собою:
  - Поширення неетичної та неправдивої інформації;
  - Поширення зовнішніх URL-посилань, зацікавлені привабливим текстом, користувачі натискаючи на них можуть бути перенаправленими на шкідливі сайти;
  - Випадковим чином додаються в друзі до інших користувачів, прагнучи охопити широку аудиторію;

Зловмисники використовують шкідливих ботів для надсилання спам- та фішингових повідомлень, поширення шкідливого програмного забезпечення,

розміщення каналів командування та управління (C&C) та запуску інших незаконних операцій.

- Кіборги: Згідно з визначенням цієї групи, до кіборгів можна віднести користувачів, у яких можна знайти ознаки як людини так і бота;

  Наприклад, типовий обліковий запис кіборга може містити різні типи твітів. Багато з них матимуть контент з людським інтелектом та оригінальністю, тоді як решта будуть опубліковані автоматично.

Дана класифікація була проведена у мережі Twitter на даних 500 тисяч користувачів та 40 мільйонів твітів. У межах аналізу враховані різноманітні аспекти: аналіз вмісту твітів, перегляд URL-адрес, інформацію про пристрої, інформацію профілю користувача, кількість підписників та друзів.

Такий мульти підхід до аналізу дозволяє нам підвищити точність класифікувати типів користувачів. Комбінування різних типів даних забезпечує надійність класифікації та може бути корисним при класифікації ненадійних акаунтів, що і є метою нашої роботи.

### 1.3 Підозрілі акаунти: означення, їх типи, шкода від них

Підозрілі акаунти це облікові записи, які викликають підозру, щодо їхньої мережевої активності. Це можуть бути новостворені, зазвичай не верифіковані, чи взламані акаунти. Підозрілі аканунти можуть мімікрувати під облікові записи інших з метою видачі себе за іншу людину. Такі акаунти зазвичай використовують для шахрайських дій, з метою поширення фейків, спаму.

У соціальних мережах можна виділити такі типи підозрілих акаунтів, а саме[8]:

- Бот-акаунти: Автоматизовані профілі, керовані програмними скриптами або ботнет-мережами, що здатні імітувати поведінку реальних користувачів;

Їх функціональність варіюється від безпечної (автоматичне інформування) до шкідливої, зокрема для поширення спаму чи пропаганди. Поведінка ботів часто налаштована таким чином, щоб ускладнити їхнє виявлення шляхом симуляції людської активності;

- Фейкові акаунти: Це облікові записи, створені з використанням неправдивої або вигаданої інформації щодо особистості. Часто такі акаунти супроводжуються штучно згенерованими зображеннями обличчя та вигаданими біографічними даними;

Метою фейкових акаунтів зазвичай є маніпулювання громадською думкою, соціальна інженерія, або шахрайство;

- Клоновані акаунти: Це профілі, що створюються шляхом копіювання особистих даних (ім’я, фото, опис) реального користувача для подальшого використання у шахрайських схемах або для компрометації репутації оригінального акаунта;
- Троль-акаунти: Акаунти, створені з метою навмисного провокування, ескалації конфліктів або деструктивної взаємодії у віртуальних спільнотах. Поведінка тролів спрямована на розпалювання ворожнечі, дискредитацію інституцій чи осіб, або саботування дискусій;

Узагальнення та систематизація типів підозрілих акаунтів є необхідним етапом для подальшого формалізованого аналізу мережевої активності користувачів. Різні типи облікових записів — від ботів до тролів і клонованих профілів — демонструють специфічні поведінкові патерни, що можуть бути використані як маркери аномальної активності.

Встановлення чіткої типології дозволяє більш цілеспрямовано підходити до побудови ознак, що ляжуть в основу моделей машинного навчання та експертних систем.

### 1.4 Ознаки підозрілих акаунтів

Найважливішим кроком для створення власної моделі прийняття рішень є визначення набору ознак, що будуть визначати підозрілий акаунт чи ні. Давайте детально переглянемо основні ознаки та характеристики, що можуть свідчити про підозрілий активність акаунта.

До основних ознак та шаблонів поведінки можна віднести[8, 10]:

- Нереалістично вигідні пропозиції – якщо щось здається надто гарним, щоб бути правдою, ймовірно, це шахрайство;
- Попросили надіслати гроші або дані – шахраї часто вдають із себе родича, банк, службовця чи коханого, щоб викликати страх або жалість;
- Невідомий або новостворений профіль – запит надходить від акаунта, якого ви не знаєте, або створеного зовсім нещодавно;
- Неповний або шаблонний профіль – мінімум особистої інформації, відсутність фото або використання стокових зображень, мала кількість публікацій;
- Підозрілі або фішингові посилання – незнайомі URL-адреси в особистих повідомленнях, які ведуть на підозрілі або фальшиві сайти;
- Небажане повідомлення з терміновим тоном – заклики до швидких дій, емоційний або тривожний стиль спілкування;
- Пропозиції швидкого заробітку або легкого працевлаштування – вакансії без попередньої заявки, «легкі гроші» або інвестиції з обіцяним високим прибутком;
- Фальшиві акаунти знаменитостей або публічних осіб – профілі, які видають себе за відомих людей для приваблення уваги та поширення шахрайського контенту;
- Низький рівень залучення – мало вподобань, коментарів, підозріле співвідношення між кількістю підписників і активністю в акаунті;

- Орфографічні та граматичні помилки – пости з грубими мовними помилками, що свідчить про відсутність професійного підходу;
- Неможливо перевірити особу чи компанію – відсутність присутності на інших платформах, офіційного сайту або відгуків користувачів;
- Немає контактних даних – відсутність електронної адреси, номера телефону, адреси або інших способів зв’язку;
- Використання погроз або залякування – вимоги передати особисті дані або гроші під тиском, нібито від державних органів чи знайомих;

Зазначені характеристики можуть стати в пригоді не лише звичайним користувачам для самостійного виявлення, але й слугувати основою для побудови нашої автоматизованої моделі. Треба відмітити, що не можливо дослідити чи дістати всі з зазначених ознак автоматизовано. Тому потрібно обирати тільки ті ознаки, що ми зможемо зібрати та об’єктивно проаналізувати.

Таблиця 1.1 містить перелік ознак та їх короткий опис. Ці ознаки доступні для збору (у більшості платформ) і можуть бути використані для аналізу акаунту[11, 11]. Деякі ознаки дозволяють робити висновки напряму, як от юзернейм чи фото, інші ж ознаки можна використовувати у комбінації з іншими.

Таблиця 1.1 – Таблиця ознак профілю користувача та їх опис

| № | Обрана ознака | Опис ознаки |
|---|---|---|
| 1 | Юзернейм | Ім'я, що написано в акаунті користувача |
| 2 | Біографія (Bio) | Короткий опис, написаний користувачем про себе |
| 3 | Фото профілю | Одна з головних функцій соціальних мереж; вона дозволяє легше розпізнати людину |

Кінець таблиці 1.1

| | | |
|---|---|---|
| 4 | Додаткова інфомація | Графа, яка зазвичай прописана самою соціальною мережою і дає змогу заповнити її користувачу за бажанням. Наприклад: досвід роботи, місце проживання, школа, університет тощо |
| 5 | Кількість підписників | Кількість інших акаунтів підписаних на нашого користувача |
| 6 | Кількість підписок | Кількість акаунтів на, які підписався наш користувач |
| 7 | Кількість постів | Кількість постів, ознака, яка дозволить нам визначити рівень активності користувача |
| 8 | Кількість лайків | Ознака, що вказує на кількість профілів, яким сподобався контент, створений користувачем |
| 9 | Текст постів | Ознака, що дозволить нам аналізувати текст |

Узагальнення наведених ознак дає змогу сформувати основу для подальшого формалізованого представлення поведінки підозрілих акаунтів. Визначені характеристики виступають важливими вхідними параметрами для побудови алгоритмів виявлення, що дозволяє системно підходити до аналізу користувацької активності у соціальних мережах. У подальших розділах ці ознаки буде адаптовано до формату, придатного для автоматизованої обробки та класифікації.

## 1.5 Поняття шкідливої інформації

Шкідлива інформація — це інформація, що спричиняє негативний вплив на психічний стан, поведінку, здоров’я особи, а також може порушити функціонування технічних або інформаційних систем[13]. До неї відносять як деструктивний

контент, так і інформацію, що використовується у злочинних чи маніпулятивних цілях.

Згідно із Законом України «Про інформацію», шкідливою може вважатися інформація, що порушує права та свободи людини, зазіхає на честь і гідність, містить заклики до насильства чи сприяє дискримінації [14].

Класифікація шкідливої інформації

- Психологічно шкідлива інформація: сцени насильства, жорстокості, суїциду.
- Маніпулятивна інформація: фейки, пропаганда, дезінформація.
- Інформація, що порушує закон: матеріали, пов’язані з тероризмом, екстремізмом, педофілією.
- Шкідливий код: віруси, трояни, фішингові повідомлення.
- Небажаний або шкідливий контент для дітей: порнографія, мова ворожнечі, азартні ігри.

Джерела шкідливої інформації

- Соціальні мережі та месенджери.
- Сайти з низькою модерацією або анонімними публікаціями.
- Рекламні мережі (у вигляді шкідливих банерів).
- E-mail розсилки (спам, фішинг).
- Даркнет-ресурси.

Наслідки впливу шкідливої інформації

- Формування викривленої картини світу (особливо у дітей).
- Підвищення рівня тривожності, агресії, депресії.
- Поширення паніки (наприклад, у випадку фейкових новин).
- Участь у небезпечних челенджах (наприклад, "синій кит" тощо).
- Зараження пристроїв вірусами, витік персональних даних.

Поняття шкідливої інформації допомагає краще зрозуміти, у чому саме полягає загроза від підозрілих акаунтів і чому важливо їх своєчасно виявляти. Такий контент може не лише дезінформувати або маніпулювати аудиторією, а й завдавати психологічної шкоди, провокувати соціальну напругу чи технічні ризики. Надалі ці аспекти будуть враховані при розробці критеріїв оцінки акаунтів, зокрема під час визначення рівня їхньої потенційної небезпеки для інформаційного середовища.

# ВИСНОВОК ДО РОЗДІЛУ 1

У першому розділі було розглянуто ключові теоретичні аспекти, пов’язані з природою та функціонуванням соціальних мереж. Також було розглянуто їхню роль у сучасному цифровому суспільстві. Було визначено основні типи соціальних платформ, що дозволило краще зрозуміти контексти, в яких виникають підозрілі акаунти та проявляється нетипова користувацька поведінка. Також було розглянуто поняття шкідливої інформації та її різновиди, що має значення для контекстуального аналізу діяльності таких акаунтів.

Особливу увагу приділено класифікації типів акаунтів, зокрема підозрілих, таких як боти, фейкові профілі, тролі та клоновані облікові записи. Наведено дві актуальні типології, запропоновані дослідниками, що базуються на реальних спостереженнях за великомасштабними даними. Це створює основу для подальшої побудови моделі підозрілої поведінки.

Важливою частиною дослідження стало виділення ознак, за якими можна ідентифікувати підозрілий акаунт. Ці ознаки охоплюють як структурні характеристики профілю, так і поведінкові шаблони, що дозволяє сформувати багатовимірне уявлення про потенційно аномальну активність.

Зібрані теоретичні відомості надалі будуть використані для формування формалізованих критеріїв підозрілості, які стануть основою для побудови алгоритму виявлення підозрілих акаунтів.

# 2 ВСТАНОВЛЕННЯ СТУПЕНЯ ПІДОЗРІЛОСТІ КОРИСТУВАЧА

## 2.1 Методи виявлення фейкових акаунтів: існуючі рішення та власне бачення

### 2.1.1 Існуючі методи встановлення підозрілості акаунту

У сучасних соціальних мережах дедалі більше актуальності набуває проблема підозрілих або фейкових акаунтів. Про потенційні загрози від взаємодії з такими обліковими записами вже було обговорено раніше, тож далі розглянемо методи їх виявлення. Існує кілька оснвоних підходів, які використовуються платформами для автоматичного та напівавтоматичного виявлення підозрілих акаунтів.[15] З метою кращого виявлення на провідних платформах впроваджують комбінування декількох підходів. Основні з них наведено нижче:

1. Поведінковий аналіз

Основою даного методу є моніторинг активності користувача. До освноних параметрів, що враховують, належать:

- Частота публікацій;
- Середній інтервал між діями користувача;
- Активність користувача за годину та цілй день;
- Взаємодія з іншими користувачами;

На основі зібраної інформації створюються поведінковий профіль, який порівнюється з типовою поведінкою реального користувача.

2. Аналіз соціального графа

Даний метод передбачає побудову графа в якому: вузли – це користувачі, ребра – їхні взаємовідносини (друг, підписник). Далі аналізується структура навколо кожного вузла:

- Кількість підписників/друзів та співвідношення між ними;
- Наявність/відсутність взаємних зв’язків;
- Участь у щільно пов’язаних групах;
- Різні метрики на побудованому графу;

Соціальні графи дозволяють виявити акаунти, які не інтегровані у звичайну мережу структуру, що допомагає легко виявляти мережі ботів

3. Контентний аналіз

Метод спрямований на аналіз текстів постів, описів, біо, хештегів на предмет виявлення даних ознак:

- Часто повторювані фрази в постах;
- Повторюваність чи шаблонність повідомлень;
- Граматична якість тексту;
- Наявність особистої інформації (фото, біографія тощо);

Відсутність особистого чи унікального контенту, або часте використання автоматично згенерованих текстів можуть свідчити про фейковий характер акаунту та його підозрілу активність.

4. Технічні характеристики

Оскільки соціальні мережі мають доступ до даних пристою та мережі, то цілком реально використовувати такі ознаки:

- IP-адреса, геолокація;
- Інформація про пристрій та user-agent;

- Частота використання однієї IP-адреси;
- Різка зміна IP-адреси та/або пристрою;

Даний метод особливо ефективний для виявлення масово створених акаунтів, дає змогу виявити використання автоматизованих програм.

Кожен із вищезгаданих методів виявлення підозрілих акаунтів має свої переваги та обмеження. У практиці соціальних платформ найчастіше використовується гібридний, який об’єднує різні підходи. Це дозволяє компенсувати слабкі сторони окремих методів та досягти більшої точності виявлення. Для порівняння способів я перечислив переваги та недоліки кожного з перечислених методів у таблиці 2.1.

Таблиця 2.1 – Порівняння переваг та недоліків перечислених методів

| Метод | Переваги | Недоліки |
|---|---|---|
| Поведінковий аналіз | (+) Дозволяє виявити автоматизовану активність<br><br>(+) Висока точність при наявності тривалої історії лій | (-) Можливі False Positive спрацювання при нетиповій поведінці справжнього користувача<br><br>(-) Потребує великий обсяг даних про користувача за певний період часу |
| Аналіз соціального графа | (+) Добре працює для пошуку бот-мереж | (-) Метод потребує повного графа, що є обчислювально затратним<br><br>(-) Погано працює при малих обсягах даних |
| Контентний аналіз | (+) Дозволяє виявити спам, пропаганду, дезінформацію<br><br>(+) Ефективний у поєнанні з NLP моделями | (-) Важко адаптувати до багатомовного середовища (різні мови, сленг, діалект, граматичні помилки тощо)<br><br>(-) Вимагає значних ресурсів для постійної обробки тексту |
| Технічні характеристики | (+) Ефективно виявляє технічно скоординовані атаки<br><br>(+) Швидке спрацювання без глибокого аналізу | (-) Можна обійти при правильно налаштованому проксі<br><br>(-) Можуть бути False Positive спрацювання при використанні спільних ресурсів в одному будинку/офісі |

### 2.1.2 Способи використання методів виявлення підозрілих акаунтів у провідних соціальних мережах

Розглянемо, які з описаних вище методів, використовуються на пракатиці в реальних соціальних мережах. Кожна компанія адаптує методики до особливостей контенту, структури мережі та очікуваної поведінки користувачів. Цей підхід дозволяє максимально ефективно виялвляти аномальні акаунти, ботів та акаунти з небезпечним контентом. Ось які підходи використовують відомі компанії:

1. Facebook
    - Поведінковий аналіз;
    - Аналіз соціального графа;
    - Контентний аналіз: Facebook активно фільтрує контент, що порушує політику платформи.[15] До такого контенту належить мова ворожнечі, дезінформація, порнографічні матеріали тощо;
    - Технічна перевірка: Обов'язкова верифікація акаунту через номер телефону;
    - Краудсорсинг (Crowdsourced Reporting): Механізм скарг від інших користувачів, який дозволяє реагувати на підозрілу поведінку, що не була виявлена автоматизованими засобами;
2. TikTok
    - Біометричний аналіз: Система може аналізувати обличчя, міміку, мову тіла та відео для перевірки справності користувача;
    - Аналіз шаблонів переглядів: Використовуються методи машинного навчання для виявлення неприродних шаблонів взаємодії з контентом (наприклад, масові перегляди з бот-акаунтів);
    - Поведінковий аналіз: Виявлення одночасних дій великої кількості акаунтів, що характерно для організованих бот-мереж;

- Комбінована модерація: Контент оцінюється як автоматизованими системами, так і командою модераторів;

3. LinkedIn
   - Контентний аналіз: Проводиться аналіз повноти профілю – порожні або мінімально заповнені облікові записи часто маркуються як підозрілі;
   - Поведінковий аналіз: Виявляються користувачі, які безконтрольно надсилають запити на зв’язок, що не є типовою поведінкою для цієї платформи;
   - Зворотний пошук зображення: Аватарки користувачів перевіряються на об'єкт використання чужих фотографій з Інтернету;

Наведені приклади свідчать про широке використання методів поведінкового, графового, контентного та технічного аналізу у практиці реальних компаній. В той же час, помітно, що кожна компанія запроваджує індивідуальні методи, адаптовані під специфіку для конкретної мережі, що дозволяє проводити більш якісний аналіз.

### 2.1.3 Огляд розробленого методу виявлення підозрілих користувачів

Серед наведених методів було обрано комбіновану методику, яка є технічно можливою до реалізації та успішного тестування. У запропонованій моделі присутні ознаки таких методів:

- Контентний аналіз: Автоматизовано проводиться аналіз особистої інформації користувача – ім’я, опис акаунта, фото профіля, наявність додаткової інформації тощо; також аналізуються тексти публікацій на предмет виявлення певних патернів чи шаблонів;

- Соціальний граф: Проводиться побудова неповного соціального графа з отриманими коофіцієнтами підозрілості, що дає змогу здійснити людську оцінку результатів (аналог модерації);

Тепер варто проаналізувати розроблений підхід до виявлення підозрілих користувачів. Основні переваги та недоліки було перечислено у таблиці 2.2.

Таблиця 2.2 – Плюси та мінуси авторського методу

| Метод | Переваги | Недоліки |
|---|---|---|
| Авторський підхід | (+) Можливо гнучко адаптувати модель для різних наборів даних<br><br>(+) Прозорість при прийнятті рішень та легка інтерпритація результатів | (-) Потреба в ретельному налаштуванні на початковому етапі<br><br>(-) Відсутність реакції на нові патерни злочинців |

Основою розробленого методу є використання інформація, що може зібрати кожна людина, однак програма автоматизує аналіз, спираючись, як на експертну думку так і на алгоритми машинного навчання. Це дозволяє об’єктивно визначити ваги та врахувати їхню узгодженість при прийнятті рішень.

Варто розглянути недоліки, перший недолік, пов’язаний із потребою підготовки та налаштування може спричинити незручності лише на етапі запуску програми, що не є критичним в довготривалій перспективі. Другий недолік стосується вразливості будь-якого методу: захист починає діяти лише після виявлення загрози. Це типовий виклик у кібербезпеці, а не недолік обраного методу. Зважаючи на аналіз недоліків, вважаємо їх незначними або ж такими, на які немає змоги вплинути.

У той же час перевага у гнукій адатації моделі допоможе легко інтергрувати її в більшість соціальних мереж.

## 2.2 Алгоритм встановлення ступеню підозрілості користувача

Задача алгоритму полягає в аналізі профіля користувача певної соціальної мережі з метою визначення ступеня його підозрілості. Детальний опис алгоритму:

1. Отримання інформації про користувача. Збираємо інформацію про профіль користувача та пости (лайки, коментарі, збережені тощо.) Заносимо інформацію до нашої бази даних.
2. Розрахунок ступеню підозрілості проводиться підсумовуванням критеріїв, при цьому кожному критерію призначається вага, що вказує на важливість певного критерія при прийнятті рішення. Ваги визначаються методом попарних порівнянь.

   Метод попарних порівнянь: метод використовується для встановлення ранжування об’єктів, порівнюючи їх попарно за певним критерієм. Замість того, щоб оцінювати всі варіанти одночасно, респондент порівнює їх по два і визначає, який з них важливіший[17].

   Спочатку проводиться формування вибірки критеріїв X = {$x_1$, $x_2$, $x_3$, …, $x_n$}, де n – кількість критеріїв, далі йде оцінка кожного критерія відносно іншої.

   При оцінці критеріїв будемо спиратися на шкалу Сааті, а саме[18, с. 8]:

   - 1 - рівна важливість критеріїв;
   - 3 - помірна перевага одного критерію над іншим;
   - 5- суттєва перевага одного критерю над іншим;
   - 7- значна перевага одного критерію над іншим;
   - 9- дуже сильна перевага одного критерію над іншим

   Після побудови матриці попарних порівнянь знаходимо суму по кожному ряду $R_i$, $i=1..n$, оцінка важливості, одержана в порівнянні з іншим критерієм та нормуємо по формулі:

$$w_i = \frac{R_i}{\sum R_i} \quad (2.1)$$

Таким чином, щоб виконувалось:

$$\sum w_i = 1 \quad (2.2)$$

У моєму випадку маємо 6 критеріїв:

- Ім'я (І);
- Біографія(Bio) (Б);
- Фото профілю (Ф);
- Наявність додаткової інформації (Н);
- Відношення Друг/Підписник (В);
- Схожість постів (С);

Таблиця 2.3 – Матриця попарних порівнянь з обчисленими вагами критеріїв

| Критерії | І | Б | Ф | Н | В | С | Сума ряду | Вага ознаки |
|---|---|---|---|---|---|---|---|---|
| І | 1 | 5 | 1/2 | 2 | 1/3 | 1/5 | 9,03 | 0,14087 |
| Б | 1/5 | 1 | 1/2 | 1/4 | 1/2 | 2 | 4,45 | 0,06939 |
| Ф | 2 | 2 | 1 | 1/2 | 1/4 | 1/6 | 5,92 | 0,09227 |
| Н | 1/2 | 4 | 2 | 1 | 4 | 1/7 | 11,64 | 0,18156 |
| В | 4 | 2 | 4 | 1/4 | 1 | 1/3 | 10,58 | 0,16504 |
| С | 5 | 1/2 | 6 | 7 | 3 | 1 | 22,50 | 0,35087 |
| Перевірка | | | | | | Сума | 64,13 | 1 |

3. Перевірка результатів. Оскільки розроблений спосіб базується на оцінці експертів, а експерт у нас тільки один, то потрібна додаткова перевірка правильності експертних тверджень.

Цю перевірку буде проведено з допомогою методів машинного навчання.

Спочатку перетворюємо текстові значення в числові, та нормалізуємо використовуючи Min-max нормалізацію[19], отримуючи стовбчики з даними в проміжку [0; 1].

Далі навчаємо модель, яка найкраще показує себе на тестовій вибірці та застосовуємо її на наших даних, у нашому випадку це Gradient Boosting. Отримуємо feature importance та основі цього формуємо вагові коофіцієнти.

У нашому випадку результат продемонстровано на рисунку 2.1. Числовий еквівалент отриманих результатів це:

- Біографія (Б): 0.054
- Фото профілю (Ф): 0.2623
- Наявність додаткової інформації (Н): 0.1542
- Відношення Друг/Підписник (В): 0.5295

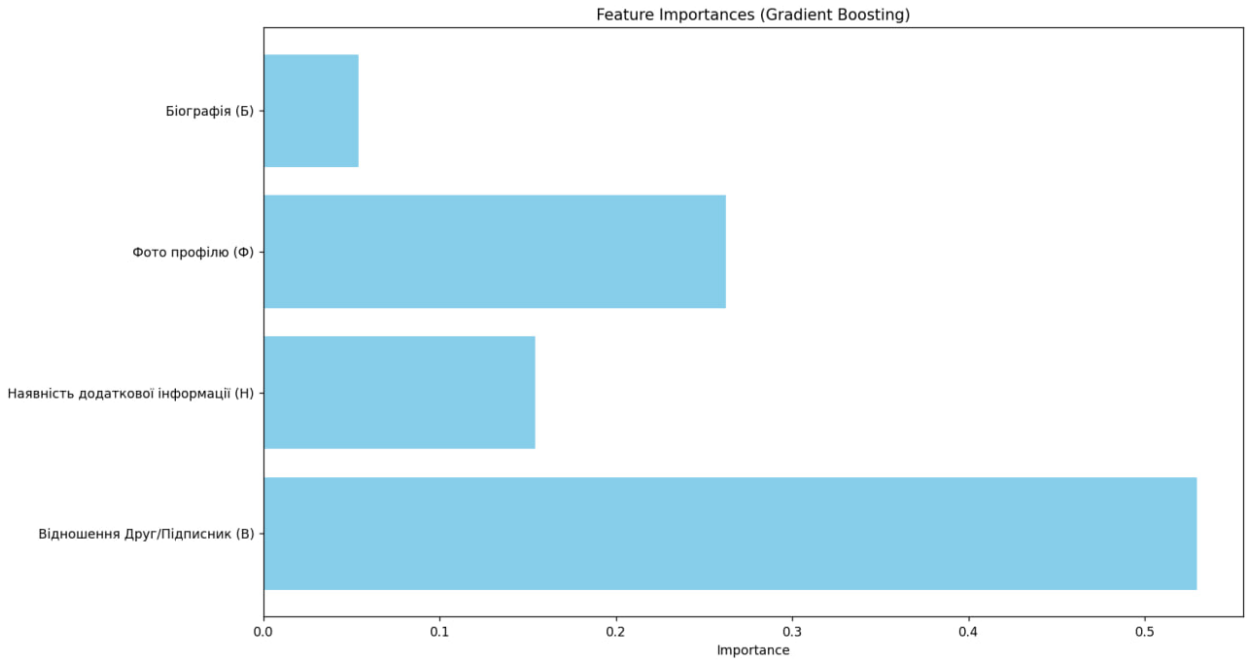


Рисунок 2.1 – Важливість критеріїв згідно інтерпретації Gradient Boosting

Як бачимо, під час визначенні ваг було використано не всі ознаки, оскільки аналіз деяких з них (ім’я; схожість постів) є складним і потребує більш тонкого підходу, що ускладнює модель для прямого аналізу цих ознак. Тому в нашому випадку обійдемося без цих ознак.

4. Узгодження ваг. В рамках роботи було здійснено оцінку важливості ознак двома способами: використовуючи експертну думку (метод попарних порівнянь) та із застосуванням алгоритмів машинного навчання. Для подальшого їх використання потрібно узгодити ваги з двох методів, щоб отримати фінальні результати ваг.

Для цього застосуємо метод агрегування зважених оцінок[20]. Спочатку нормалізуємо обидва набори вагових коофіцієнтів та визначаємо співвідношення довіри $\alpha$ для кожного методу, де:

- $\alpha$ – довіра до есперта, у нашому випадку 0.4;
- $(1 - \alpha)$ – довіра до методів машинного навчання, у нашому випадку 0.6;

Далі обчислюємо узгоджені ваги з допомогою формули 2.3

$$w_i^{final} = \alpha * w_i^{pcm} + (1 - \alpha) * w_i^{mlm} \quad (2.3)$$

де:

- $w_i^{final}$ – узгоджені ваги
- $w_i^{pcm}$ – ваги матриці попарних порівнянь
- $w_i^{mlm}$ – ваги методу машинного навчання

Як бачимо в результаті обчислень у нас виникло “перенакопичення ваг”, тобто умова 2.2 не виконується. Так сталося через поєднання частових даних, щоб продовжити працювати з вагами нам потрібно перенормалізувати обчислені значення. В результаті отримуємо фінальні ваги. У таблиці 2.4 можемо бачити проміжні та фінальні результати.

Таблиця 2.4 – Фінальні ваги, отримані за методом агрегування зважених оцінок

| Критерії | Ваги МПП | Ваги ММН | Ваги узгоджені | Фінальні ваги |
|---|---|---|---|---|
| І | 0,1409 | 0 | 0,141 | 0,1088 |
| Б | 0,0694 | 0,0540 | 0,060 | 0,0465 |
| Ф | 0,0923 | 0,2623 | 0,194 | 0,1500 |
| Н | 0,1816 | 0,1542 | 0,165 | 0,1275 |
| В | 0,1650 | 0,5295 | 0,384 | 0,2963 |
| С | 0,3509 | 0 | 0,351 | 0,2709 |
| Сума: | 1 | 1 | 1,295 | 1 |

5. Проводимо оцінку акаунту згідно обраних ознак. У результаті маємо отримати множину, C = {$c_1$, $c_2$, $c_3$, …, $c_n$}, де $c_n \in [0; 1]$, де:
    - 0 – дана ознака не стосується даного акаунту;
    - 1 – дана ознака стосується даного акаунту;

    Власне оцінка буде детально обговорена в розділі 3, а відповідні критерії оцінки будуть наочно продемонстровані в підрозділі, присвяченому реалізації методології.

6. Для обчислення інтергрального показника підозрілості скористаємось формулою:

$$f = w_1 * c_1 + w_2 * c_2 + \ \dots \ + w_n * c_n \qquad (2.4)$$

    де:

    - $w_i$ – ваговий коофіцієнт обраного критерія
    - $c_i$ – оцінка акаунта згідно обраного критерія
    - $n$ – кількість критеріїв

7. Для інтерпретація числових результатів вводимо якісну шкалу категорій, яка розбиває діапазон значень на п'ять рівнів підозрілості:
    - Низький, $0 \leq score < 0.2$;

- Нижче середнього, $0.2 \leq score < 0.4$;
- Середній, $0.4 \leq score < 0.6$;
- Вище середнього, $0.6 \leq score < 0.8$;
- Високий, $0.8 \leq score \leq 1$;

8. Візуалізую граф зв’язків

З метою забезпечення інформативності при побудові графа будемо спиратися на такі правила:

- Візуалізація графа буде інтерактивною
- Колір ребер буде залежати від типу зв’язку (зелений – друзі, помаранчевий – підписник)
- Розмір вузла залежить від кількості підписників
- Колір вузла залежить від отриманого рівня підозрілості (зелений, жовто-зелений, жовтий, помаранчевий, червоний – відповідно до вказаних рівнів підозрілості)
- Про деяких користувачів відсутня інформація, тому колір їх вузла буде сірим.

## 2.3 Соціальний граф мережі

Соціальний граф — це граф, вузли якого представлені соціальними об'єктами, такими як профілі користувача з різними атрибутами (наприклад: ім'я, день народження, рідне місто, тощо), співтовариства, медіа-контент, тощо, а ребра — соціальними зв'язками між ними[21].

Формально соціальний граф визначається як:

$$G = (V, E), \tag{2.5}$$

де:

- V – множина вершин (вузлів), що відповідає користувачам або іншим об’єктам мережі;
- E – множина ребер (зв’язків), які представляють соціальну взаємодію: дружбу, підписку, лайки, коментарі тощо;

Залежності від особливостей обраної соціальної мережі та характеристик взаємодії між користувачами можемо стикнутися з різними типами графа:

1. Орієнтовний граф[22]

У орієнтованому графі кожне ребро має напрямям – від одного вузла до іншого. Це означає, що взамодія може бути асиметричною: користувач А може впливати на користувача В, але це не означає, що В діє у відповідь.

Такий тип графа широко використовується для моделювання сучасних соціальних мереж, таких як Twitter, Instagram, Telegram, де переважають односторонні зв’язки.

У нашій роботі ми будемо працювати саме з орієнтовними графами, оскільки вони краще відображають сучасні соціальні мережі.

2. Неорієнтований граф

У неорієнтованому графі зв’язки не мають напряму, тому взаємодія вважається симетричною: якщо користувач А пов’язаний із користувачем В, то й В пов’язаний з А. Це характрено, для моделей, де взаємоія можлива лише за взаємною згодою. Прикладами таких соціальних мереж є:

- Facebook, який притримується класичної моделі “Друзів” (хоча інші взаємодії можуть бути асиметричними)
- LinkedIn (зв’язки першого рівня – лише за згодою обох сторін)
- Окремі, менш популярні форуми, або мережі

Як було сказано раніше, нашу увагу буде зосереджено на соціальних мережах, що моделюються з допомогою орієнтованого графа, тож доцільно буде розглянути типи соціальних зв'язків, які можуть бути представлені у такій структурі.

Структуру зв'язків у межах орієнтовного графа можна проілюструвати на прикладі, зображеному на рис. 2.2. Можемо виокремити три основні типи зв'язку між вузлами:

1. Підписники – користувачі, які стежать за оновленнями іншого користувача

Користувач $u$ є підписником користувача $v$, якщо існує ребро з напрямком:

$$(u, v) \in E \text{ і } (v,\ u) \notin E$$

У нашому випадку користувач С є підписником користувача В

2. Друзі – користувачі, які взаємно підписалися на оновлення одне одного

Користувачі $u$ та $v$ є друзями, якщо існує ребро з напрямком:

$$(u, v) \in E \text{ і } (v,\ u) \in E$$

У нашому випадку користувач А та користувач В є друзями

3. Незнайомці – користувачі, у яких відсутній зв'язок

Користувачі $u$ та $v$ є незнайомцями, якщо жодне ребро між ними не існує:

$$(u, v) \notin E \text{ і } (v,\ u) \notin E$$

У нашому випадку користувач D є незнайомцем для користувачів А, В, С

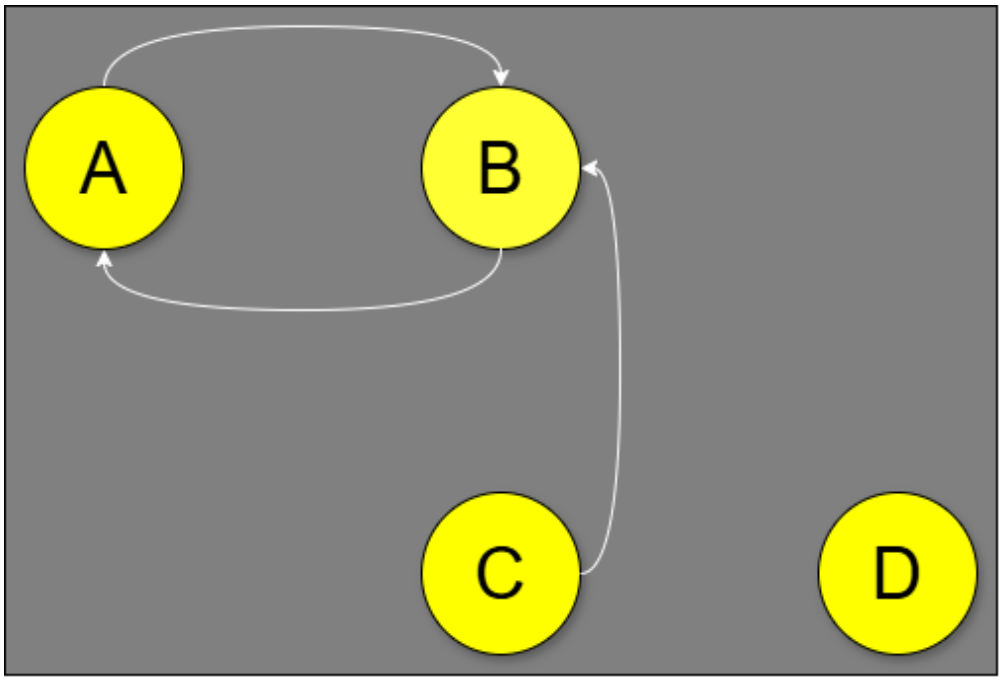


Рисунок 2.2 – Соціальний граф орієнтованої мережі

## 2.4 Алгоритм встановлення соціальних зв’язків

Задача алгоритму полягає у візуалізації соціальних зв’язків, обраного нами користувача. Алгоритм можна описати так:

1. Ініціаліазція графа – створюю порожній інтерактивний граф;
2. Задаємо початковий вузол – задаємо початкового користувача, з якого продовжимо побудову графа;
3. Задаємо зв’язки між користувачами – для кожного друга та підписника додаємо ребро;
4. Рекурсивна побудова – в залежності від початкових параметрів запуску, побудова завершується, або ж запускається додаткова побудова для кожного знайденого користувача;
5. Застосовуються візуальні параметри обговорені у алгоритмі вище;
6. Побудований граф повертається у вигляді інтерактивної .html сторінки;

В результаті отримуємо візуалізацію соціального графа, в центрі якого розташований обраний нами користувач. Така візуалізація дозволяє зручно проводити подальший аналіз зв’язків.

# ВИСНОВОК ДО РОЗДІЛУ 2

Під час роботи над другим розділом дипломної роботи було проведено огляд сучасних підходів до виявлення підозрілих акаунтів у соціальних пережах. Було розглянуто основні моделі, які використовуються у провідних платформах, а також проаналізовано їхні переваги та недоліки. На основі проведеного аналізу запропоновано власний комбінований метод виявлення підозрілих акаунтів, що враховує як експертні оцінки, так і елементи машинного навчання.

Окрему увагу приділено побудові соціального графа: його структурі, типам взаємозв’язків між користувачами та принципам візуалізації. Визначено особливості орієнтованого та неорієнтованого графа та обґрунтовано вибір першого для моделювання сучасних соціальних мереж.

У рамках запропонованого підходу сформульовано два основних алгоритми: алгоритм визначення ступеня підозрілості користувача та алгоритм побудови соціальних зв’язків. Обидва рішення лягли в основу програмної реалізації, що буде розглянута у наступному розділі.

# 3 РОЗРОБКА ПЗ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

## 3.1 Програмне забезпечення

Програма була виконана на мові програмування Python 3.11.4. Для реалізації коду були використані такий список бібліотек:

- Re – бібліотека для роботи з регулярними виразами; допомагає шукати, змінювати та перевіряти текстові шаблони;
- Pandas – найпопулярніша бібліотека для роботи з датафреймами;
- Numpy – бібліотека для ефективної та зручної обробки числових масивів;
- Matplotlib – бібліотека для візуалізації різних графіків, діаграм, гістограм тощо;
- Sklearn – бібліотека для машинного навчання. В ході роботи буде використано декілька різних модулів[23, 24];
- Pyvis – бібліотека для створення інтерактивних графів, заснована на бібліотеці NetworkX. Виступає нашим основним методом візуалізації соціальних графів;
- Ast – бібліотека для аналізу python коду як структури даних (синтаксичного дерева);

## 3.2 Процес розробки програмного забезпечення

Для проведення аналізу було використано датасет, який містить інформацію про 6609 користувачів соціальної мережі. Датасет містить мітки класу, що вказують на підозрілість акаунту, отже, ми зможемо перевірити розроблений нами метод. Основні ознаки користувачів у датасеті представлені наступними атрибутами:

- Id: Унікальний ідентифікатор користувача;
- Name: Ім’я користувача;
- Bio: Опис профіля;

- photo_url: Фото профілю;
- has_extra_info: Ідентифікатор наявності додаткової інформації;
- friends: Перелік друзів акаунта;
- followers: Перелік підписників акаунта;
- is_verified: Статус верифікації акаунту (при розробці методології даний параметр було виключено, оскільки існують соціальні мережі, де можна купляти статус “ verified ”);
- class: Мітка класу, що вказує підозрілий акаунт чи ні, де 0 – це не підозрілий, 1 - підозрілий

Окрім цього, до датасету було прикріплено додатковий датасет, що містить інформацію з постами цих користувачів за певний проміжок часу. Загалом датасет містить інформацію про 17001 пост. Основними атрибутами цього датасету є:

- user_id: Унікальний ідентифікатор користувача, якому належить даний пост
- text: Текс поста

Датасет містить і інші стовбці, але через недостачу даних ними буде знехтувано при використанні розробленої моделі.

- likes: Кількість лайків
- comments_count: Кількість коментарів
- shared: Кількість поширень
- timestamp: Дата опублікування поста

### 3.2.1 Підготовка даних та визначення вагових коофіцієнтів

В методолії було вказано про додаткове визначення ваг з допомогою методів машинного навчання. Реалізація цього в коді виконана двома функціями, а саме: scaling_modified_data(), model_chose()  та getting_weights(). Нижче наведені стислий опис реалізації функцій.

Оскільки не можливо працювати з сирим датасетом для коректної роботи обраних алгоритмів, то спочатку потрібно здійснити попередню обробку датасету. Для цього завантажуєм датасет *users.csv* та обробляєм його отримуючи потрібні параметри:[25] *id, has_bio, has_photo, has_extra_info, friends_followers_ratio*.

Після формування датафрейму із зазначених параметрів проводимо нормалізацію для усунення маштабних відмінностей між ознаками. Зберігаємо отриманий датасет як: *prepared_features_scaled.csv*. Паралельно створюємо додатковий датасет *users_modified.csv* з вже обчисленим стовбцем *friends_followers_ratio* для зручності у подальшому аналізі.

З допомогою функції model_chose (), маючи всі нормалізовані дані, проводимо відбір найкращої моделі:

- Тестуємо декілька різних моделей, оцінюємо їх метрики та обираємо найкращий варіант:

Таблиця 3.1 – Результати тестування різних моделей

| Модель | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| Gradient Boosting | 0,7358 | 0,5750 | 0,3581 | 0,4414 |
| k-NN | 0,7105 | 0,5053 | 0,3270 | 0,3971 |
| Random Forest | 0,7070 | 0,4961 | 0,3305 | 0,3967 |
| LightGBM | 0,7317 | 0,5772 | 0,2976 | 0,3927 |
| XGBoost | 0,7171 | 0,5248 | 0,3114 | 0,3909 |
| Naïve Bayes | 0,7181 | 0,5282 | 0,3080 | 0,3891 |
| MLP | 0,7302 | 0,5729 | 0,2924 | 0,3872 |
| Decision Tree | 0,7110 | 0,5073 | 0,3010 | 0,3779 |
| Cat Boost | 0,7342 | 0,6000 | 0,2647 | 0,3673 |
| SVM | 0,7327 | 0,5960 | 0,2578 | 0,3599 |
| Logistic Regression | 0,7307 | 0,5902 | 0,2491 | 0,3504 |

З таблиці 3.1 можемо бачити, що моделі мають однакому точність в межах сотої похибки, отже, потрібно зробити вибір на основі іншого параметра. У нашому випадку f1-score в моделі Gradient Boosting значно кращий у порівннянні з іншими моделями, тож обираємо його.

В ф-ції getting_weights() проводимо навчанням обраної моделі та обираємо найкращі параметри для неї:

- Ініціалізуємо модель Gradient Boosting[26], задаємо *random_state* для повторюваності результатів;

```
# === Модель ===
gb = GradientBoostingClassifier(random_state=42)
```

Рисунок 3.1 – Ініціалізація моделі

- Обираємо параметри для перебору для вибору найкращих параметрів;

```
# === Параметри для перебору ===
param_grid = {
    'n_estimators': [50, 100, 200, 500],
    'learning_rate': [0.01, 0.1, 0.2],
    'max_depth': [3, 5, 7],
    'subsample': [0.7, 1.0]
}
```

Рисунок 3.2 – Параметри моделі для перебору

- Запускаємо GridSearchCV для автоматизованого перебору, ключовим параметром виступає f1-score;

В ході цього алгоритму отримуємо найкращі параметри, які дають максимальний f1-score: 0.424

```
Найкращі параметри:
{'learning_rate': 0.2, 'max_depth': 3, 'n_estimators': 100, 'subsample': 0.7}
```

Рисунок 3.3 – Найкращі параметри моделі

Згідно отриманої Confusion matrix, отриманої при застосуванні моделі тестовій вибірці, можемо зробити висновки, що модель має труднощі з визначенням класу 1, бо він часто хибно передбачувався, як клас 0.

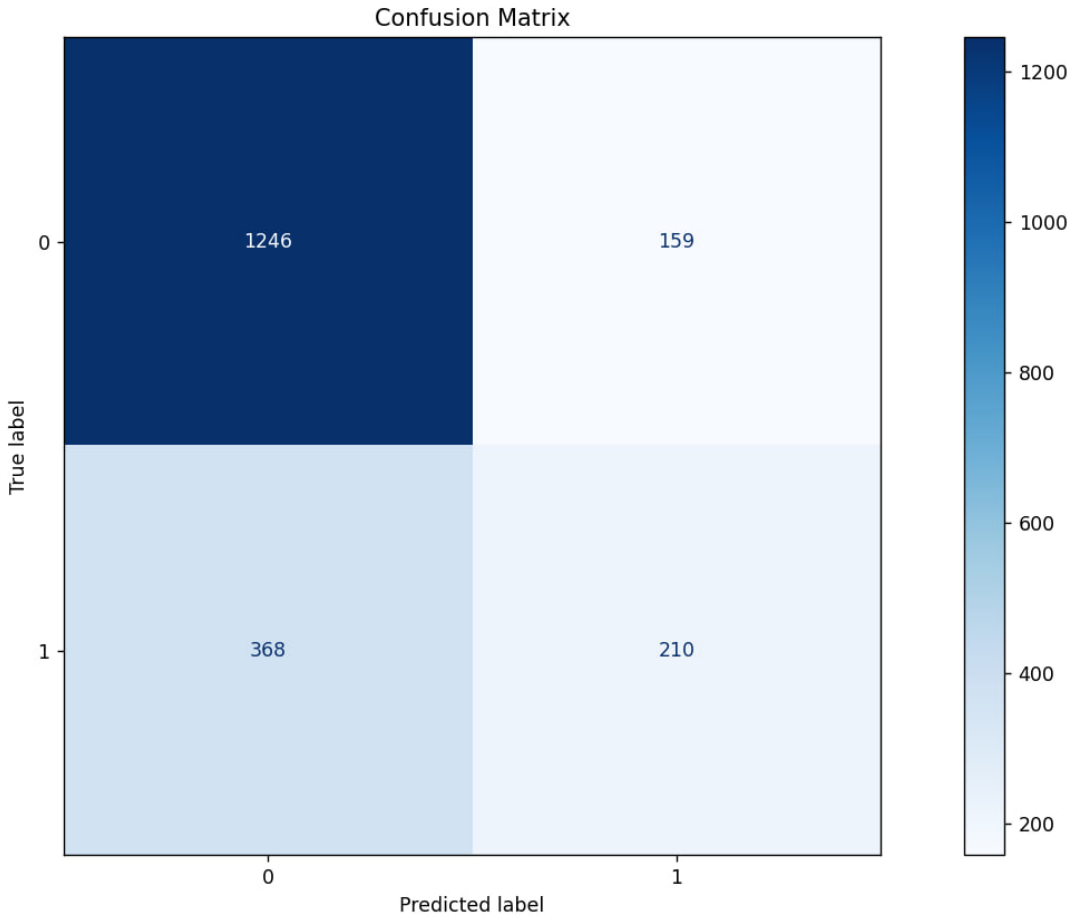


Рисунок 3.4 – Confusion matrix натренованої моделі

В результаті отримуємо ваги важливості та будуємо гістограму, яку ми вже спостерігали на рисунку 2.1.

### 3.2.2 Аналіз відношення друг/підписник

Одним із ключови індикаторів при оцінці рівня підозрілості акаунта у соціальній мережі є атрибут *friends_followers_ratio*, який відповідає за відношення кількості друзів користувача до кількості його підписників. Цей показник потрібно правильно оцінити та проаналізувати. Для цього в нашому коді використовується функція friends_followers_ratio_analise(). В масив ratios записуємо всі значення з стовбця відношення та передаємо його для обчислення метрик. Можемо бачити реалізацію на рисунку 3.4.

```
# === Перетворення дані на числовий тип та видалимо NaN ===
df['friends_followers_ratio'] = pd.to_numeric(df['friends_followers_ratio'], errors='coerce')
ratios = df['friends_followers_ratio'].dropna()

# === Обчислення основних метрик ===
print("Основна статистика:")
print(ratios.describe())

# === Обчислення квантілі ===
print("\nКвантілі:")
print(ratios.quantile([0.1, 0.25, 0.5, 0.75, 0.9]))
```

Рисунок 3.5 – Формування масиву ratios та обчислення статистичних метрик

У результати виконання функції отримані такі ключові характеристики:

```
Основна статистика:
count    6609.000000
mean        3.770236
std         4.402750
min         0.000000
25%         0.500000
50%         2.200000
75%         5.333333
max        25.000000

Квантілі:
0.10     0.000000
0.25     0.500000
0.50     2.200000
0.75     5.333333
0.90    10.000000
```

Рисунок 3.6 – Результати виконання ф-ції friends_followers_ratio_analise().

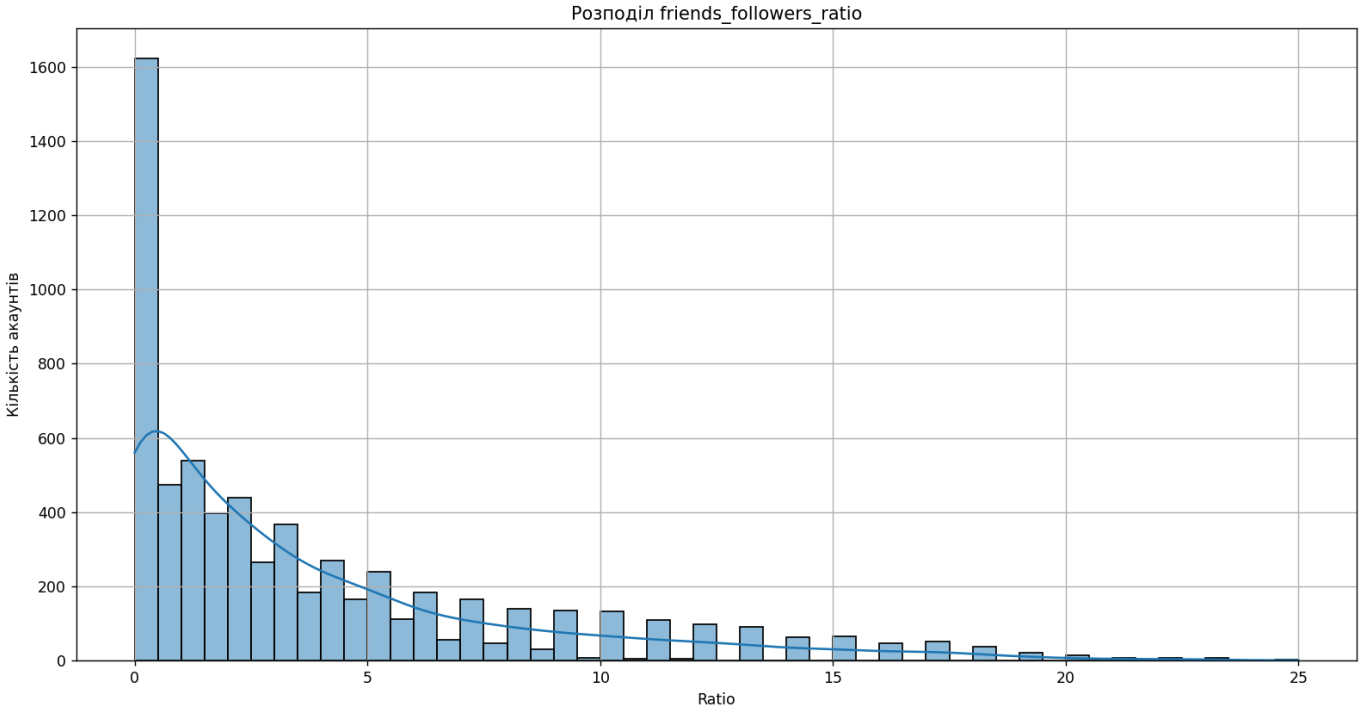


Рисунок 3.7 – Гістограма розподілу ratio та кількість користувачів

Тепер варто інтерпретувати результати:

- Середнє значення: 3.77 – загальна тенденція, вказує на те, що більшість користувачів мають більше друзів, ніж підписників;
- Медіана: 2.2 – означає, що половина користувачів мають приблизно таке співвідношення;
- 10% перцентіль: 0 – вказує на те, що 10% не мають друзів взагалі, або ж дуже велику кількість фоловерів;
- 25% перцентіль: 0.53 – чверть акаунтів мають співвідношення, менше або дорівнює 0.53;
- 90% перцентіль: 10 – 10% мають ratio більше ніж 10, що доволі підозріло;
- Максимум: 25 – крайнє значення ratio, може вказувати на поведінку бота;

Та зробити висновки:

- Значення ratio в межах 1–2.5 можна вважати типовими для звичайних користувачів соціальної мережі.
- Значення нижче 0.5 може свідчити про накручування підписників або про пасивну взаємодію з іншими користувачами (відсутність ініціативи у додаванні друзів).
- Значення вище 5 розглядається як потенційно аномальне та може вказувати на автоматизовану або спам-орієнтовану поведінку.
- Ratio понад 10 є високим ризиком та, ймовірно, вказує на ботоподібний акаунт, що здійснює масову розсилку запитів або спроби штучного нарощування мережі контактів.

Дані висновки знадобляться нам в подальшому при оцінці акаунту.

### 3.2.3 Визначення показника підозрілості та його інтерпретація

Згідно метології після отримання вагових коофіцієнтів для кожного критерія нам потрібно провести оцінку акаунта користувача. У цьому нам допоможе функція score_count(). Вона проводить оцінку акаунта на основі інформації з профіля користувача та обчислює інтегральний показник підозрілості. Оцінка акаунта проводиться згідно таких міркувань:

- Ім'я користувача: Один з основних атрибутів профілю, оскільки більшість користувачів довго думають над його ретельним вибором, іноді вдаючись до символізму. З цієї причина імена, що складаються лише цифри є підозрілими та отримують оцінку 1.

  У деяких випадках користувачі можуть випадково забути чи не придумати нічого оригінального і тоді їм надається шаблонне ім’я типу "user12345", де 12345 певний унікальний набір чисел, але в той же час акаунти,

які генеруються автоматизовано теж ймовірно матимуть шаблонні імені. Такі імена оцінюються цей в 0.5.

- Біографія (Bio): Bio це короткий опис акаунта чи самої людини. Тут може бути потенційно, що завгодно, але коли біо взагалі порожнє, то це наштовхує на певні підозри, хоча ймовірні і випадки коли людина просто нічого не хоче писати про сете, тому такі випадки оцінюєм в 0.5.

  Також хочеться відмити, що підозріло коли біо містить лише посилання, яке оцінюємо в 1. Можна подумати, що це посилання на якийсь інший блог чи ресурс, але більшість соцмереж забезпечують додаткові поля спеціально для цього в додатковій інформації, тому немає сенсу писати їх в біо, окрім, що для зловмисницьких дій.
- Фотографія профілю: Фото це можливість людині для самовираження, люди схильні ставити себе на аватарку або ж якісь фото з якими вони себе асоціюють, тому доводі підозріло коли воно геть відстунє, такі випадки оцінюємо в 1. У той же час фото, яке можна знайти по першому запиту є цілком можливоми для вибору але від цього не менш підозрілими, тому оцінка 0.5.
- Наявність додаткової інформації: Це гарний показник зацікавленості людини в своєму профілі. Чим більше даних людина вказала (наприклад, місто, освіта, місце роботи), тим більш активно вона веде себе у соцмережі.

  Доцільно було б проводити додаткову перевірку цього параметра, але в нашому датасеті більш точної інформації ніж наявність/відсутність немає, тому працюємо з цим. У випадку коли інформація є присвоюємо оцінку 0, за відсутності — оцінку 1.

- Відношення друг/підписник: Спираючись на результати аналізу, який було проведено в минулому підозділі маємо таку оцінку:
  - Ratio $\leq 0.1$ — оцінка 1;

- $0.1 < Ratio \leq 0.5$ — оцінка 0.5;
- $0.5 < Ratio \leq 5$ — оцінка 0;
- $5 < Ratio \leq 10$ — оцінка 0.5;
- $Ratio > 10$ — оцінка 0.1;

- Схожість постів: Високий ступінь схожості (понад 80%) може вказувати на автоматизоване поширення інформації або спам-активність. У такому випадку акаунти отримують оцінку — 1.

За розробленою шкалою оцінки, акаунт вважається 100% підозрілим із інтегральним значенням 1, лише у випадку, коли кожен з критеріїв отримує максимальне значення оцінки в 1. Такий підхід дозволяє здійснювати градацію рівня підозрілості та враховувати неоднохначність кожного з факторів.

Інтерпретація результатів проходить в функції results_interpretation() згідно заданої якісної шкали.

### 3.2.4 Візуалізація соціального графа

Візуалізації соціального графа проходить з допомогою коду в файлі “Social graph.py” з допомогою освновної ф-ції add_user_and_connections(). Вона працює на основі бібліотеки pyvis.network. При запуску ф-ції вказується 2 основних параметри:

- User_id: вказує користувача для якого буде побудовано соціальний граф мережі;
- Max_depth: визначає наскільки глибоко буде показана соціальна мережа (тільки друзі; друзі друзів і так далі в глибину);

Інші візуальні рішення, що було впроваджено, вже вмонтовані в код та не потребують додаткового налаштування.

### 3.3 Аналіз отриманих результатів

Після застосування всього функціоналу було тримано результати, які представлені на рисунку 3.7.

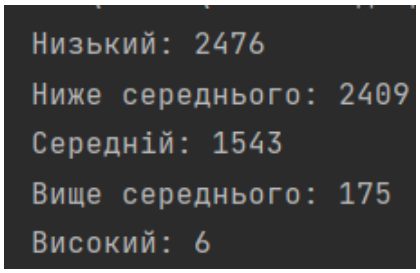


Рисунок 3.8 — Результат інтерпретації, обчисленого показника підозрілості

Також було проведено перевірку нашої моделі згідно отриманих значень score. Всі акаунти з середнім та вищим ризиком вважаються підозрілими та отримують клас 1, інші ж — клас 0. Після звірки значень отримуємо результати:

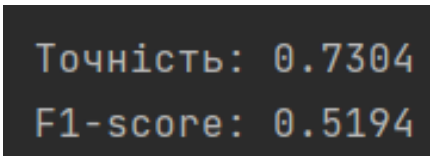


Рисунок 3.9 — Результат інтерпретації, обчисленого показника підозрілості

Була побудувана та проаналізована Confusion matrix, на жаль, у нас все ще присутні випадки коли відбувається неправильне передбачення, але краще ніж тільки при використанні моделі Gradient Boosting. Пояснити такі відхилення можна декількома факторами:

- Акаунти з аномальною поведінкою: користувачі можуть мати ознаки підозрілої активності, але бути чистими;
- Незбалансованість класів: модель гірше вчиться виявляти клас 1 через його не велику частоту появи;

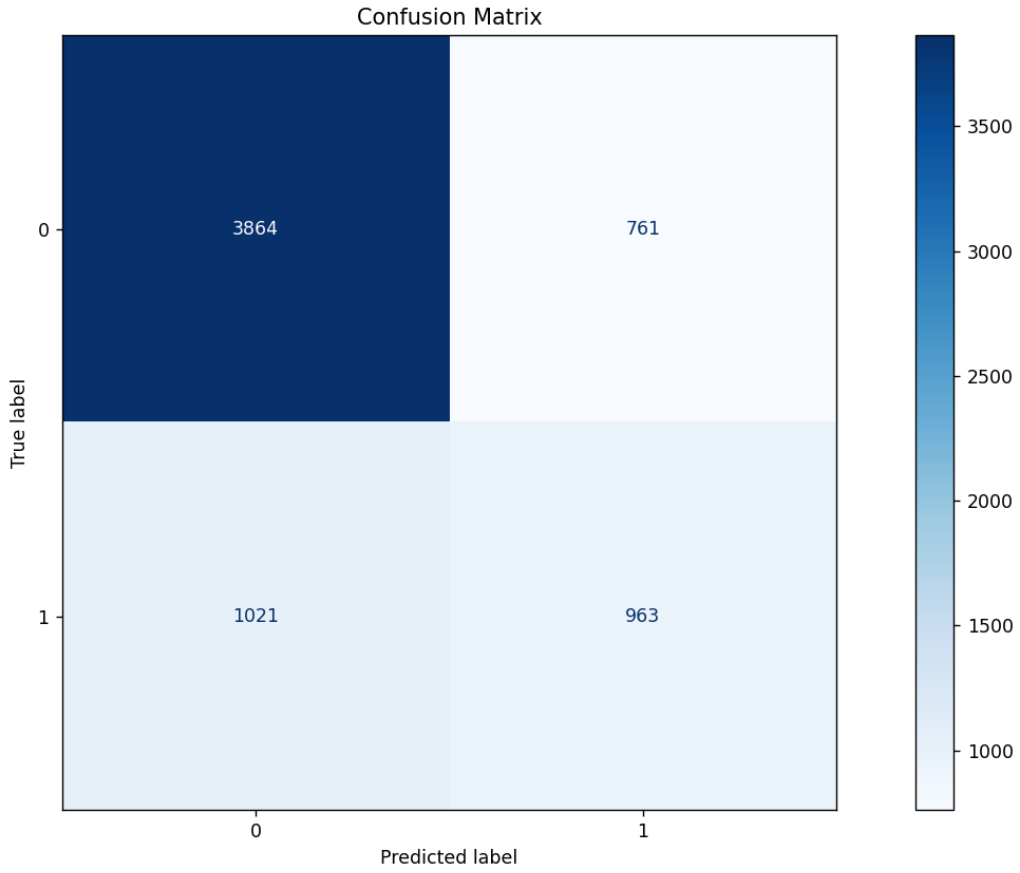


Рисунок 3.10 — Confusion matrix власної моделі

В кінці було проведено статистичний аналіз отриманих значень показника підозрілості (score), результати якого можемо бачити на рисунку 3.11.

| Основна статистика: | |
|---|---|
| count | 6609.000000 |
| mean | 0.278996 |
| std | 0.168450 |
| min | 0.000000 |
| 25% | 0.150000 |
| 50% | 0.277500 |
| 75% | 0.409550 |
| max | 0.922350 |

| Квантілі: | |
|---|---|
| 0.2 | 0.13205 |
| 0.4 | 0.20440 |
| 0.6 | 0.31955 |
| 0.8 | 0.44630 |
| 1.0 | 0.92235 |

Рисунок 3.11 — Статистичний аналіз обчислених значень score

На основі проведено аналізу можемо зробити висновки щодо нашої групи користувачів:

- Середнє значення: 0.279 — це вказує на те, що переважна більшість користувачів потряпляє до категорії “Нижче середнього” за рівнем підозрілості;
- Медіана: 0.258 — майже збігається із середнім;
- Максимум: 0.9223 — жоден акаунт із нашої вибірка не наблизився до значення 1, тобто система не виявила абсолютно підозрілих акаунтів;
- Мінімум: 0 — тобто є користувачі, які за всіма критеріями не мають ознак підозрілої поведінки;

В ході аналізу було запропоновано використовувати динамічні парамаметри для інтерпретації результатів, а власне квантілі. Після тестування отримуємо точність — 0.698 та f1-score — 0.5706. Точність впала, але f1-score виріс, то ж вважаємо це за успіх.

Результати на Confusion matrix теж змінились. Можемо сказати, що наша модель стала краще передбачати клас 1, але в школу 0 класу. Модель стала більш консервативної, але в рамках кіберзахисту це може бути навіть перевагою.

Оскільки метою дослідження є захист користувачів від потенційно небезпечних акаунтів, то є доцільним застосування консервативного підходу з жорстокішими критеріями відбору, що зменшує ризик пропуску дійсно підозрілих користувачів.

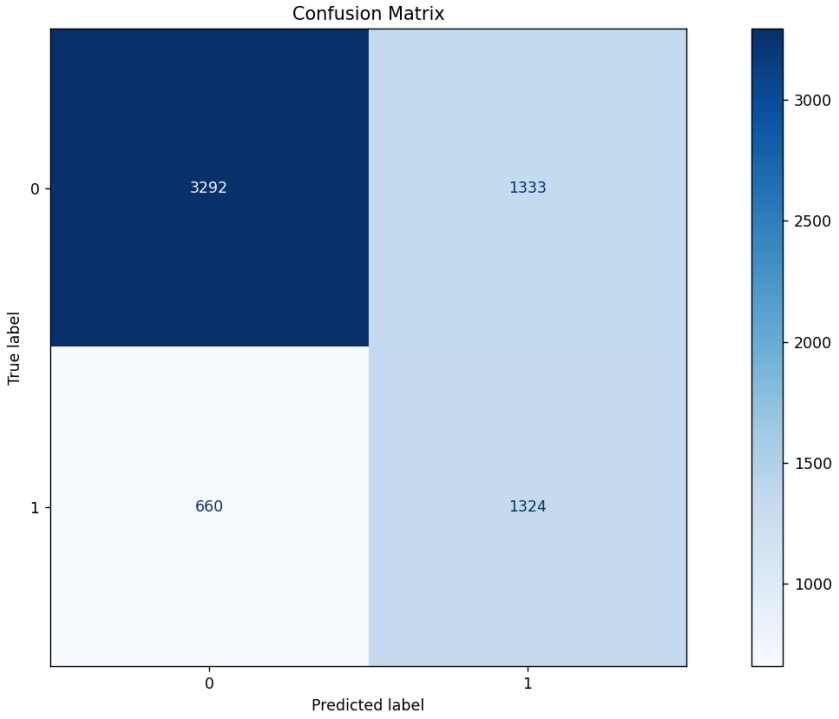


Рисунок 3.12 — Confusion matrix власної моделі після аналізу

Крім того, аналіз розподілу показав наявність лівої асиметрії, що означає домінування низьких значень score. Це можна побачити на рисунку 3.13. Гістограма представляє кількісний розподіл користувачів за рівнем підозрілості.

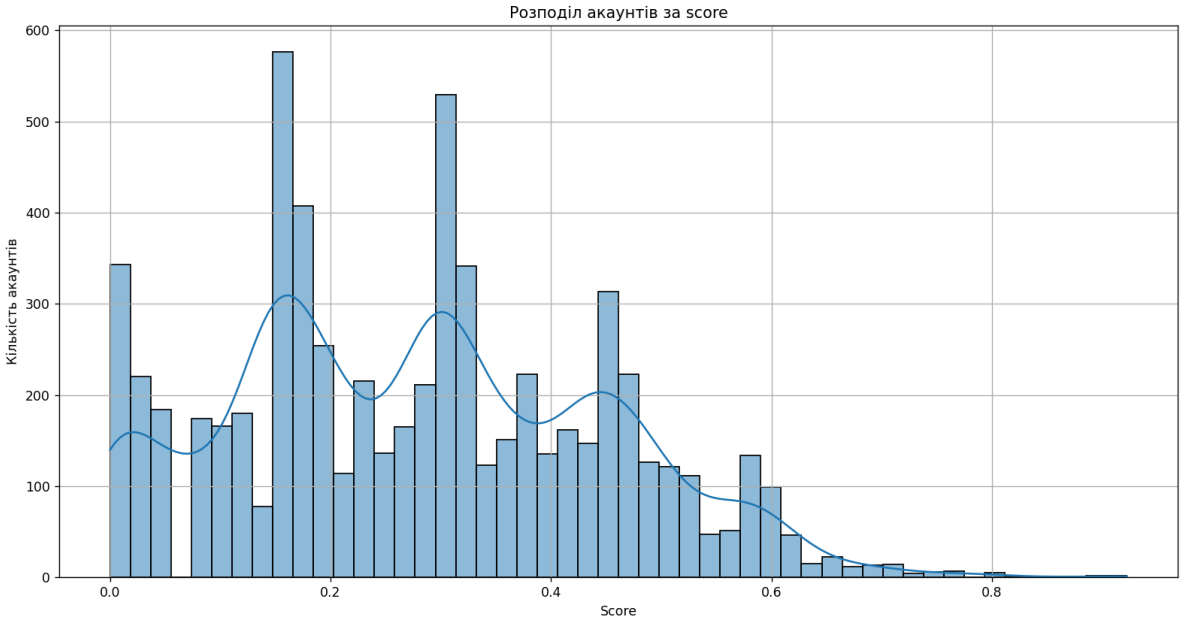


Рисунок 3.13 — Розподіл акаунтів за визначеним показником підозрілості

На рисунку 3.14 бачимо приклад візуалізації соціального графа для обраного користувача. На рисунку показано локальну мережу користувача: його друзі та підписники (глибина 0 у нашому способі побудови).

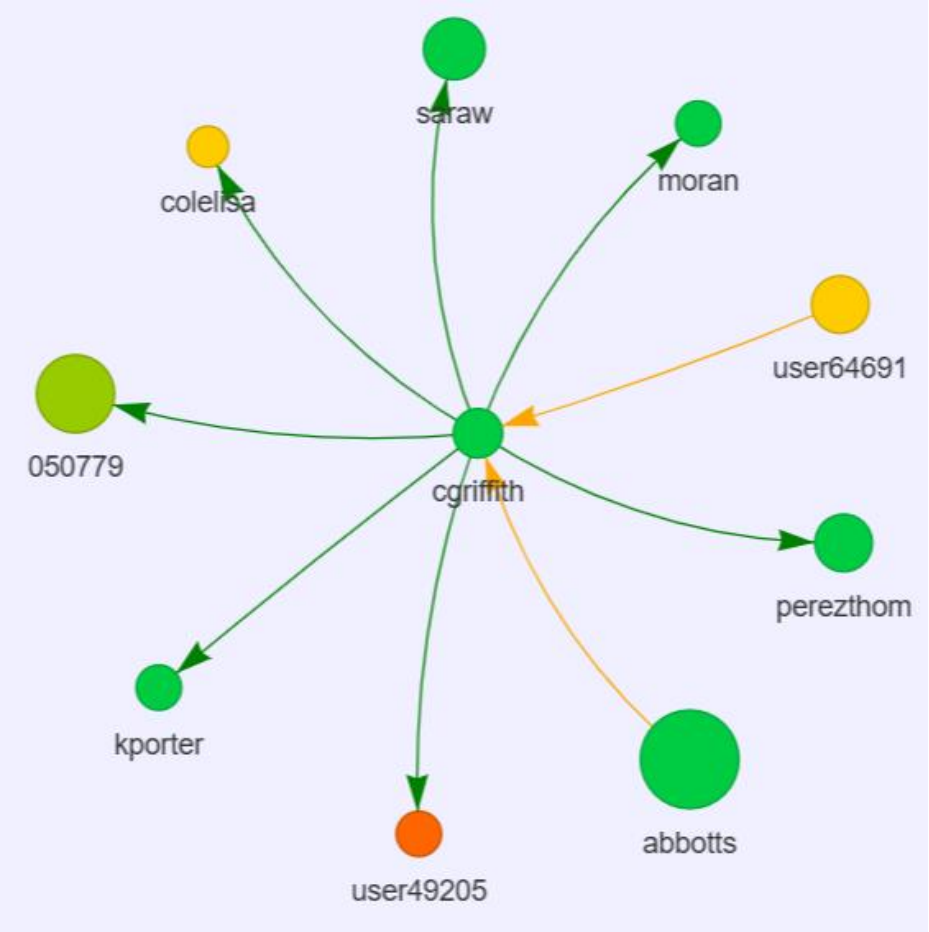


Рисунок 3.14 — Соціальний граф користувача *cgriffith* (глибина 0)

На рисунку 3.14 зображено розширений соціальний граф. На графі можемо бачити ширший погляд на соціальну мережу цього користувача. Рохглянемо потенційні загрози, що можемо побачити на соціальному графі

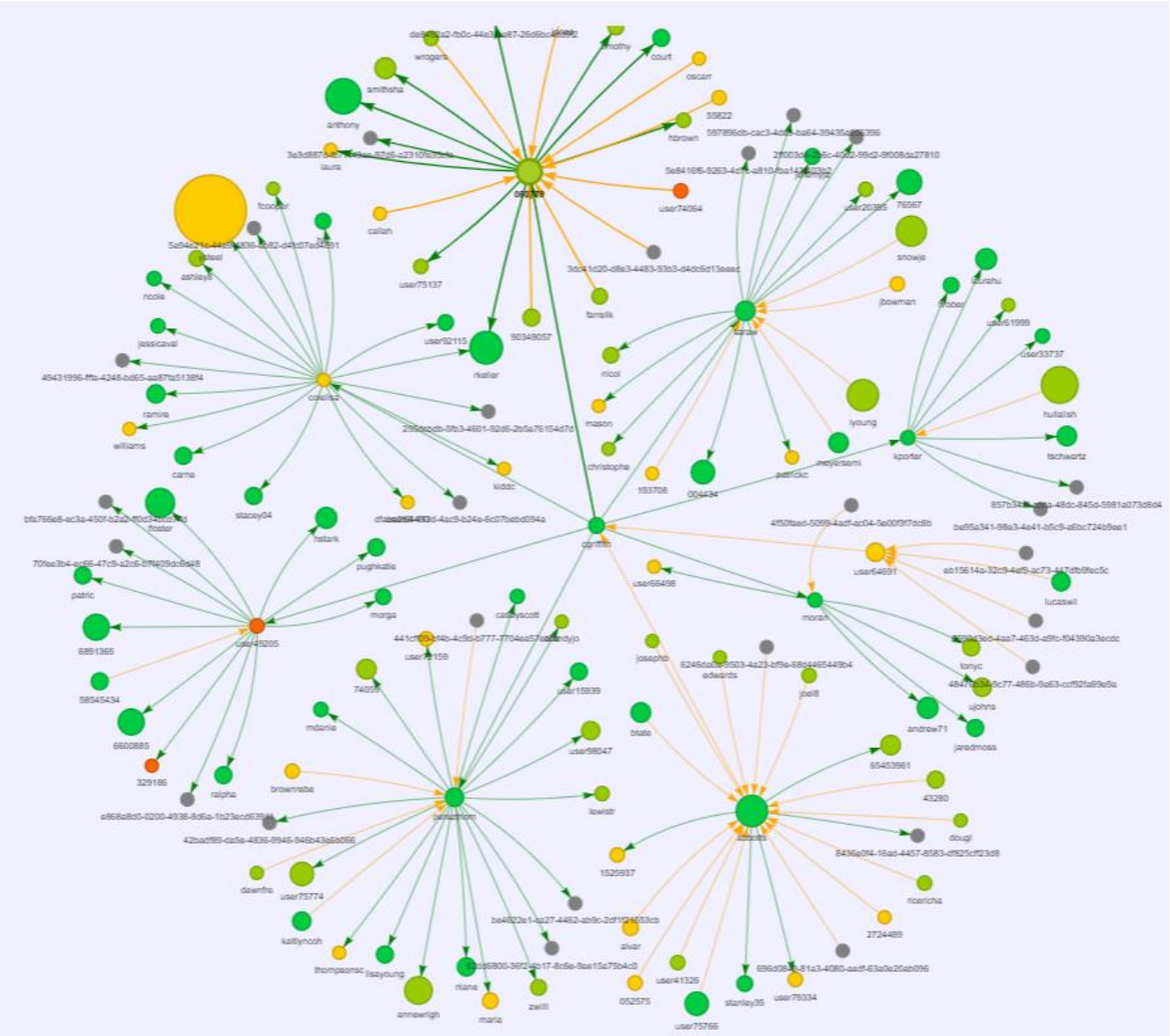


Рисунок 3.15 — Соціальний граф користувача *nicoleturn* (глибина 1)

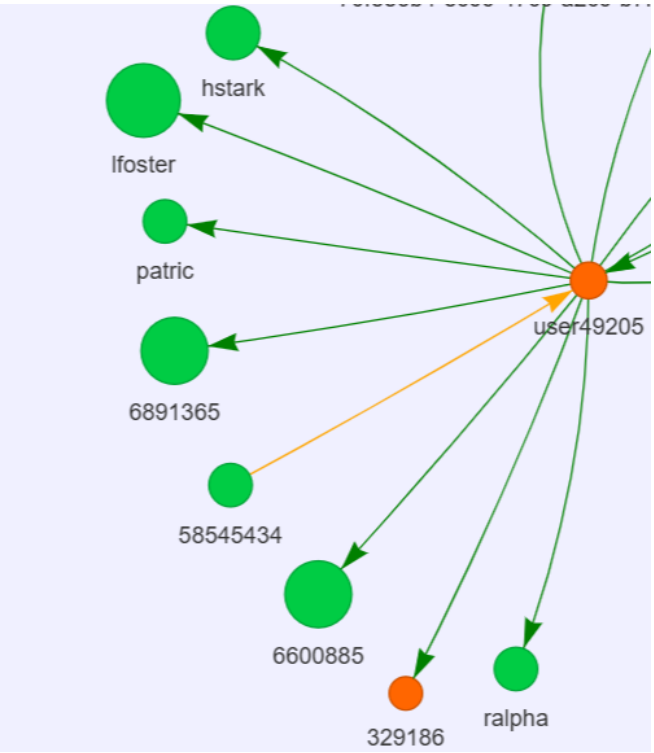


Рисунок 3.16 — Аналіз соціальних зв’язків

Бачимо зв’язок нашого користувача з підозрілим користувачем (червоний вузол). Підозріло, що в нього багато друзів з числовими іменами, одного ми змогли детектувати, решту ж ні. Таке можливо через те, що акаунти ймовірно щойно створені та не займаються підозрілою активністю (рисунок 3.15)

В іншому ж є акаунти середньої підозрілості, котрі пов’язані з його друзями, якщо такі користувачі вийдуть на контакт, варто запитати довірених друзів про надійність користувача.

# ВИСНОВОК ДО РОЗДІЛУ 3

У цьому розділі було реалізовано розроблене програмне забезпечення, яке включає функціонал для обробки даних, визначення вагових коефіцієнтів, розрахунку інтегрального показника підозрілості акаунта та побудови соціального графа з урахуванням типів зв’язків. Було здійснено повний цикл від підготовки даних до інтерактивної візуалізації результатів.

Програмне забезпечення протестовано на датасеті обсягом 6609 облікових записів. За результатами тестування модель показала хорошу точність класифікації (0.7304), але середнє значення f1-score (0.5706) вказує на наявність класового дисбалансу та можливі труднощі з виявленням підозрілих акаунтів. У порівнянні лише з ММН модель показала покращення f1-score на 34.6%, що підтверджує ефективність проведених удосконалень.

Запропонована методика дозволяє пояснювати за рахунок яких саме ознак була сформована відповідна оцінка. На відміну від деяких «чорних скриньок» у машинному навчанні, запропонований підхід базується на прозорих механізмах визначення ваг через поєднання експертної оцінки та моделі Random Forest, що також забезпечує гнучкість і адаптивність системи.

Крім того, візуалізація соціального графа з глибиною зв’язків, кольоровим кодуванням рівнів підозрілості та типів зв’язку значно спрощує аналіз складних взаємозв’язків і сприяє практичному застосуванню в рамках систем кібермоніторингу.

# ВИСНОВКИ

В рамках даної дипломної роботи було повністю реалізовано поставлену мету: розробити програмне забезпечення для виявлення підозрілих акаунтів у соціальних мережах шляхом аналізу мережевих зв’язків та побудови соціального графа.

Проведено теоретичне обґрунтування проблеми, здійснено аналіз існуючих методів, запропоновано власний підхід, що поєднує елементи контентного аналізу, графового моделювання та машинного навчання. Продемонстровані кількісні результати та їх інтерпретація з допомогою якісної шкали. Результати показують працездатність розробленої методики.

У порівнянні з існуючими рішеннями, що використовуються у провідних соціальних мережах (Facebook, TikTok, LinkedIn), запропонований підхід відрізняється відкритістю, адаптивністю та можливістю застосування без внутрішнього доступу до системи. Це робить його придатним для незалежного аудиту, досліджень і впровадження в корпоративних чи державних структурах. Програмне забезпечення орієнтоване на широке коло користувачів та легко масштабується.

Перспективним напрямком подальших досліджень є вдосконалення методів автоматичного збору даних з відкритих API, розширення набору ознак користувачів, а також інтеграція текстових моделей для глибшого контентного аналізу. Окремо варто розглянути можливість розширення методології на відеоконтент і взаємодію між платформами.

# ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ


1. EU DisinfoLab. Reports on disinformation campaigns. — URL: https://www.disinfo.eu/
2. Теоретичні і прикладні проблеми фізики, математики та інформатики: матеріали XXIII Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених (14 – 17 травня 2025 р., м. Київ, Україна) / Уклад.: Пономаренко С. М., Бех С. В., Степаненко В. М., Козленко О. В., Мирошникова І. Ю., Мікава П. В., Деркач О. Г. – Київ : КПІ ім. Ігоря Сікорського, Видавництво «Політехніка», 2025. – 474 с. ISBN 978-966-990-053-1
3. Kaplan A. M., Haenlein M. Users of the world, unite! The challenges and opportunities of Social Media // Business Horizons. – 2010. – Vol. 53, No. 1. – P. 59–68.
4. Datareportal. Digital 2024: Global Overview Report. — 01/31/2024. — URL: https://datareportal.com/reports/digital-2024-global-overview-report
5. Datareportal. Global social media Statistics— April 2025. — URL: https://datareportal.com/social-media-users
6. Krishnamurthy, B.; Gill, P.; Arlitt, M.F. A few chirps about twitter. In Proceedings of the WOSN ’08: Proceedings of the First Workshop on Online Social Networks, Seattle, WA, USA, 17–22 August 2008; pp. 19–24.
7. Chu, Z.; Gianvecchio, S.; Wang, H.; Jajodia, S. Who is Tweeting on Twitter: Human, Bot, or Cyborg? In Proceedings of the 26th Annual Computer Security Applications Conference, Austin, TX, USA, 6–10 December 2010; pp. 21–30
8. Zannettou S., Sirivianos M., Blackburn J., Kourtellis N. The Web of False Information: Rumors, Fake News, Hoaxes, Clickbait, and Various Other Shenanigans // J. Data and Information Quality (JDIQ). – 2019. – Vol. 11, No. 3. – P. 1–37.
9. Trend Micro. How to Spot Scams:  Red Flags in Social Media Requests. — 03/14/2025. —URL: https://helpcenter.trendmicro.com/en-us/article/tmka-09614?utm_source=scampage&utm_medium=featarticle

10. TechTarget. 10 social media scams and how to avoid them. — 07/24/2024. — URL: https://www.techtarget.com/whatis/feature/Social-media-scams-and-how-to-avoid-them
11. Beskow D. M., Carley K. M. Bot-hunter: A tiered approach to detecting & characterizing automated activity on Twitter // Proc. of ASONAM. – 2019.
12. Cresci S., Di Pietro R., Petrocchi M., Spognardi A., Tesconi M. Fame for sale: Efficient detection of fake Twitter followers // Decision Support Systems. – 2015. – Vol. 80. – P. 56–71.
13. Marwick A., Lewis R. Media Manipulation and Disinformation Online: Data & Society Report. – 2017. – URL: https://datasociety.net/pubs/oh/DataAndSociety_MediaManipulationAndDisinformationOnline.pdf
14. Закон України «Про інформацію» № 2657-XII від 02.10.1992. — URL: https://zakon.rada.gov.ua/laws/show/2657-12
15. ENISA Threat Landscape Report 2023. – URL: https://www.enisa.europa.eu/
16. Meta Transparency Center. Community Standards Enforcement Report. – URL: https://transparency.fb.com/enforcement/
17. Ткачова, О. (2015). МЕТОД САТІ ПРИ ПРИЙНЯТТІ УПРАВЛІНСЬКИХ РІШЕНЬ (PDF). ISSN 1814-1161. — URL: https://web.archive.org/web/20220325052718/http:/irbis-nbuv.gov.ua/cgi-bin/irbis_nbuv/cgiirbis_64.exe?C21COM=2&I21DBN=UJRN&P21DBN=UJRN&IMAGE_FILE_DOWNLOAD=1&Image_file_name=PDF%2Fdrep_2015_4_17.pdf
18. Стьопочкіна І., Кіфорчук К. Комплексні системи захисту інформації: проектування, впровадження, супровід. Методичний посібник до практичних занять. — 2024. — 38 с.
19. Han J., Kamber M., Pei J. Data Mining: Concepts and Techniques. – 3rd ed. – Morgan Kaufmann, 2011. – 744 p.

20. Т. Л. Сааті. "Прийняття рішень. Метод аналізу ієрархій". – К.: Основи, 2008. – 320 с.

21. Horowitz D., Kamvar S. D. The Anatomy of a Large-Scale Social Search Engine // Proc. of WWW'10, Raleigh, USA, 26–30 April 2010. – P. 431–440.

22. Бондаренко В. М., Бойко В. М., Місюра Ю. В. Теорія графів та її застосування. – К.: Видавництво КНЕУ, 2003. – 240 с.

23. Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M., Duchesnay É.
*Scikit-learn: Machine Learning in Python* // Journal of Machine Learning Research. – 2011. – Vol. 12. – P. 2825–2830.

24. Müller A., Guido S. Introduction to Machine Learning with Python. – O'Reilly Media, 2016.

25. Guyon, I., & Elisseeff, A. An Introduction to Variable and Feature Selection // Journal of Machine Learning Research. – 2003. – Vol. 3. – P. 1157–1182.

26. Friedman J. H. Greedy function approximation: a gradient boosting machine // The Annals of Statistics. – 2001. – Vol. 29, № 5. – P. 1189–1232. – DOI: 10.1214/aos/1013203451.

# ДОДАТОК А

“Model analize.py”

```
# === Підключення бібліотек ===
import ast
import warnings
import numpy as np
import pandas as pd
from sklearn.svm import SVC
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier
from catboost import CatBoostClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.preprocessing import MinMaxScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score


def remove_zeros(input_csv, output_csv, n_remove):
    """
    Видаляє випадкові n_remove рядків з класом 0 та зберігає результат.

    Параметри:
    - input_csv: шлях до вхідного CSV-файлу
    - output_csv: шлях для збереження обробленого CSV
    - n_remove: кількість випадкових рядків з класом 0 для видалення
    - id_column: якщо True, видаляє перший стовпець (id)
    """

    df = pd.read_csv(input_csv)

    zeros = df[df.iloc[:, -1] == 0]

    if len(zeros) < n_remove:
        raise ValueError(f"У файлі лише {len(zeros)} рядків з класом 0, а ти хочеш
видалити {n_remove}")

    drop_indices = np.random.choice(zeros.index, size=n_remove, replace=False)
    df = df.drop(drop_indices)

    df.to_csv(output_csv, index=False)
    print(f"Файл збережено у {output_csv}, видалено {n_remove} рядків з класом 0.")


def scaling_modified_data():

    """
        Функція завантажує users.csv. Обирає потрібні для аналізу ознаки, перетворює
їх в числовий тип даних та нормалізує.
```

```
        Повертає:
        Оброблений датасет users_modified.csv
        Оброблений датасет prepared_features_scaled.csv.
    """

    # === Завантаження даних ===
    users = pd.read_csv("users7030_1.csv")

    users_simple = pd.read_csv("users7030_1.csv")

    # === Обробка users.csv ===
    users["bio_len"] = users["bio"].fillna("").apply(len)
    users["has_photo"] = users["photo_url"].notna().astype(int)
    users["has_extra_info"] = users["has_extra_info"].astype(bool).astype(int)

    users["num_friends"] = users["friends"].apply(lambda x: len(ast.literal_eval(x))
if pd.notna(x) else 0)
    users["num_followers"] = users["followers"].apply(lambda x:
len(ast.literal_eval(x)) if pd.notna(x) else 0)

    # === Створення нових ознак ===
    users["has_bio"] = users["bio"].fillna("").apply(lambda x: int(len(x.strip()) >
0))

    users["friends_followers_ratio"] = users.apply(
        lambda row: row["num_friends"] / row["num_followers"] if row["num_followers"]
!= 0 else 0,
        axis=1
    )

    # === Створення допоміжного датасета ===
    friends_followers_ratio = users.apply(
        lambda row: row["num_friends"] / row["num_followers"] if row["num_followers"]
!= 0 else 0,
        axis=1
    )
    users_simple["friends_followers_ratio"] = friends_followers_ratio
    users_simple.to_csv("users7030_modified_1.csv", index=False)

    # === Формування фінального датафрейму ===
    features_custom = users[[
        "id", "has_bio", "has_photo", "has_extra_info",
        "friends_followers_ratio"
    ]]

    # === Нормалізація показників ===
    ids = features_custom["id"]
    data_to_scale = features_custom[["friends_followers_ratio"]]

    scaler = MinMaxScaler()
    scaled_data = scaler.fit_transform(data_to_scale)

    features_scaled = pd.DataFrame(scaled_data, columns=data_to_scale.columns)
    features_scaled.insert(0, "id", ids)
    features_scaled.insert(1, "has_bio", features_custom["has_bio"])
    features_scaled.insert(2, "has_photo", features_custom["has_photo"])
    features_scaled.insert(3, "has_extra_info", features_custom["has_extra_info"])
```

```
    features_scaled.insert(5, "class", users["class"])

    # === Збереження результатів ===
    features_scaled.to_csv("prepared_features_scaled_1.csv", index=False)

    print("=== Modified & Scaled Features ===")
    print(features_scaled.head())


def model_chose():
    """

        Функція завантажує prepared_features_scaled_1.csv. Навчає декілька моделей та
оцінює їх.

    """

    warnings.filterwarnings("ignore")

    # === Завантаження даних ===
    df = pd.read_csv("prepared_features_scaled_1.csv")

    df = df.drop(columns=['id'])

    # === Розділення на ознаки та мітки ===
    X = df.iloc[:, :-1]  # усі колонки, крім останньої
    y = df.iloc[:, -1].astype(int)  # остання колонка — мітка класу (0/1)

    # === Поділ на train/test ===
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.3, random_state=42
    )

    # === Список моделей ===
    models = {
        "Logistic Regression": LogisticRegression(),
        "k-NN": KNeighborsClassifier(),
        "Naive Bayes": GaussianNB(),
        "Decision Tree": DecisionTreeClassifier(),
        "Random Forest": RandomForestClassifier(),
        "Gradient Boosting": GradientBoostingClassifier(),
        "XGBoost": XGBClassifier(verbosity=0),
        "LightGBM": LGBMClassifier(verbose=-1),
        "CatBoost": CatBoostClassifier(verbose=0),
        "SVM": SVC(probability=True),
        "MLP": MLPClassifier(max_iter=500)
    }

    # === Оцінка моделей ===
    results = []

    for name, model in models.items():
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)

        acc = accuracy_score(y_test, y_pred)
        prec = precision_score(y_test, y_pred, zero_division=0)
        rec = recall_score(y_test, y_pred, zero_division=0)
```

```
        f1 = f1_score(y_test, y_pred, zero_division=0)

        results.append((name, acc, prec, rec, f1))

    # === Вивід результатів ===
    results_df = pd.DataFrame(
        results,
        columns=["Model", "Accuracy", "Precision", "Recall", "F1-score"]
    ).sort_values(by="F1-score", ascending=False)

    print("\n=== Результати класифікації ===\n")
    print(results_df.to_string(index=False))


#remove_zeros("users_1.csv", "users7030_1.csv", n_remove=1848)

#scaling_modified_data()

#model_chose()
```

“Getting weights.py”

```
# === Підключення бібліотек ===
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay


def getting_weights():

    """
        Функція завантажує prepared_features_scaled.csv. Навчаємо модель
GradientBoostingClassifier підбираючи найкращі параметри.
        В Результаті отримуємо вагові коофіцієнти критеріїв на основі отриманих
feature importance.

        Повертає:
        Консольний вивід списка критеріїв та їхню важливість.
    """

    # === Завантаження даних ===
    df = pd.read_csv("prepared_features_scaled_1.csv")

    df = df.drop(columns=['id'])

    X = df.iloc[:, :-1]
    y = df.iloc[:, -1]

    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.3, random_state=42
    )
```

```
# === Модель ===
gb = GradientBoostingClassifier(random_state=42)

# === Параметри для перебору ===
param_grid = {
    'n_estimators': [50, 100, 200, 500],
    'learning_rate': [0.01, 0.1, 0.2],
    'max_depth': [3, 5, 7],
    'subsample': [0.7, 1.0]
}

# === GridSearchCV===
grid_search = GridSearchCV(
    estimator=gb,
    param_grid=param_grid,
    scoring='f1',
    cv=5,
    n_jobs=-1,
    verbose=2
)

grid_search.fit(X_train, y_train)

print("Найкращі параметри:")
print(grid_search.best_params_)

print("\nКращий F1-score на валідації:")
print(grid_search.best_score_)

best_model = grid_search.best_estimator_


# === Прогноз на тестових даних ===
y_pred = best_model.predict(X_test)

# === Класифікаційний звіт ===
print("\n=== Звіт по класифікації ===")
print(classification_report(y_test, y_pred))

# === Confusion Matrix ===
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot(cmap='Blues')
plt.title("Confusion Matrix")
plt.show()

# === Важливість ознак ===
feature_names_map = {
    "has_bio": "Біографія (Б)",
    "has_photo": "Фото профілю (Ф)",
    "has_extra_info": "Наявність додаткової інформації (Н)",
    "friends_followers_ratio": "Відношення Друг/Підписник (В)"
}

importances = best_model.feature_importances_
features = X.columns
readable_features = [feature_names_map.get(f, f) for f in features]
```

```
    # === Текстовий вивід у консоль ===
    print("\n=== Важливість ознак ===")
    for feature, importance in zip(features, importances):
        readable_name = feature_names_map.get(feature, feature)
        print(f"{readable_name:<40}: {importance:.4f}")

    # === Графік ===
    plt.figure(figsize=(10, 6))
    plt.barh(readable_features, importances, color='skyblue')
    plt.xlabel("Importance")
    plt.title("Feature Importances (Gradient Boosting)")
    plt.gca().invert_yaxis()
    plt.tight_layout()
    plt.show()


def friends_followers_ratio_analise():

    """
        Функція обчислює метрики та будує гістограму.

        Повертає:
        Консольний вивід основної статистики та квантілів.
        Візуалізація гістограми.
    """

    # === Завантаження даних ===
    df = pd.read_csv('users7030_modified_1.csv')

    # === Перетворення дані на числовий тип та видалимо NaN ===
    df['friends_followers_ratio'] = pd.to_numeric(df['friends_followers_ratio'],
errors='coerce')
    ratios = df['friends_followers_ratio'].dropna()

    # === Обчислення основних метрик ===
    print("Основна статистика:")
    print(ratios.describe())

    # === Обчислення квантілів ===
    print("\nКвантілі:")
    print(ratios.quantile([0.1, 0.25, 0.5, 0.75, 0.9]))

    # === Побудуємо гістограму ===
    plt.figure(figsize=(10, 4))
    sns.histplot(ratios, bins=50, kde=True)
    plt.title('Розподіл friends_followers_ratio')
    plt.xlabel('Ratio')
    plt.ylabel('Кількість акаунтів')
    plt.grid(True)
    plt.show()


#getting_weights()

#friends_followers_ratio_analise()
```

“Score calculation.py”

```
from sklearn.metrics import ConfusionMatrixDisplay
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score, f1_score, confusion_matrix


def score_count():

    """
        Функція завантажує users_modified.csv та posts.csv. Обирає потрібні для
аналізу ознаки, перетворює їх в числовий тип даних та нормалізує.

        Повертає:
        Оброблений датасет users_score.csv з додатковою колонкою "score"
    """

    # === Завантаження даних ===
    df_users = pd.read_csv('users7030_modified_1.csv')
    df_posts = pd.read_csv('posts.csv')

    # === Створюємо словник: user_id -> список текстів постів ===
    user_posts = df_posts.groupby('user_id')['text'].apply(list).to_dict()

    # === Власна функція для обчислення score ===
    def calculate_score(row):

        score = 0

        # === 1. Ім'я ===
        name = str(row['name']).strip()
        if name.isdigit():
            score += 1 * 0.1088
        elif re.fullmatch(r'user\d{5,}', name):
            score += 0.5 * 0.1088
        else:
            score += 0

        # === 2. Bio ===
        bio = str(row['bio']).strip() if pd.notna(row['bio']) else ''
        if not bio:
            score += 0.5 * 0.0465
        elif re.fullmatch(r'https?://\S+', bio):
            score += 1 * 0.0465

        # === 3. Фото ===
        photo = str(row['photo_url']).strip() if pd.notna(row['photo_url']) else ''
        if not photo:
            score += 1 * 0.15
        elif 'i.pravatar.cc' in photo:
            score += 0.5 * 0.15

        # === 4. Додаткова інформація ===
        if not row.get('has_extra_info', False):
            score += 1 * 0.1275

        # === 5. Відношення кількість друзів/підписників ===
```

```
        ratio = float(row['friends_followers_ratio'])
        if ratio <= 0.1:
            score += 1.0 * 0.2963
        elif ratio <= 0.5:
            score += 0.5 * 0.2963
        elif ratio > 10:
            score += 1.0 * 0.2963
        elif ratio > 5:
            score += 0.5 * 0.2963

        # === 6. Подібність постів ===
        posts = user_posts.get(row['id'], [])
        if len(posts) > 1:
            try:
                tfidf = TfidfVectorizer().fit_transform(posts)
                sim_matrix = cosine_similarity(tfidf)
                n = len(posts)
                total_sim = (sim_matrix.sum() - n) / (n * (n - 1))

                if total_sim > 0.8:
                    score += 1 * 0.2709
            except:
                pass

        return score

    # === Додаємо колонку "score" в датасет ===
    df_users['score'] = df_users.apply(calculate_score, axis=1)

    # === Зберігаємо результат ===
    df_users.to_csv('users_score_1.csv', index=False)


def results_interpretation():

    # === Завантаження даних ===
    df_users = pd.read_csv('users_score_1.csv')

    # === Інтерпретуємо результати ===
    df_users['class_predicted'] = np.where(df_users['score'] < 0.4, 0, 1)

    # === Інтерпретуємо оцінки в групи ===
    scores = df_users['score'].values

    low = np.sum((scores >= 0.0) & (scores < 0.2))
    below_average = np.sum((scores >= 0.2) & (scores < 0.4))
    average = np.sum((scores >= 0.4) & (scores < 0.6))
    above_average = np.sum((scores >= 0.6) & (scores < 0.8))
    high = np.sum((scores >= 0.8) & (scores <= 1.0))

    # === Консольний вивід результатів ===
    print(f"Низький: {low}")
    print(f"Ниже середнього: {below_average}")
    print(f"Середній: {average}")
    print(f"Вище середнього: {above_average}")
    print(f"Високий: {high}")

    # === Обчислення точності ===
```

```
    accuracy = accuracy_score(df_users['class'], df_users['class_predicted'])
    print(f"\nТочність: {accuracy:.4f}")

    # === Обчислення F1-score ===
    f1 = f1_score(df_users['class'], df_users['class_predicted'])
    print(f"F1-score: {f1:.4f}")

    # === Матриця неточностей ===
    cm = confusion_matrix(df_users['class'], df_users['class_predicted'])

    disp = ConfusionMatrixDisplay(confusion_matrix=cm)
    disp.plot(cmap='Blues')
    plt.title("Confusion Matrix")
    plt.show()


def score_analise():

    """
            Функція обчислює метрики та будує гістограму.

            Повертає:
            Консольний вивід основної статистики та квантілів.
            Візуалізація гістограми.
        """

    # === Завантаження даних ===
    df = pd.read_csv('users_score_1.csv')

    # === Перетворення дані на числовий тип та видалимо NaN ===
    df['score'] = pd.to_numeric(df['score'], errors='coerce')
    ratios = df['score'].dropna()

    # === Обчислення основних метрик ===
    print("Основна статистика:")
    print(ratios.describe())

    # === Обчислення квантілів ===
    print("\nКвантілі:")
    print(ratios.quantile([0.2, 0.4, 0.6, 0.8, 1]))

    # === Побудуємо гістограму ===
    plt.figure(figsize=(10, 4))
    sns.histplot(ratios, bins=50, kde=True)
    plt.title('Розподіл акаунтів за score')
    plt.xlabel('Score')
    plt.ylabel('Кількість акаунтів')
    plt.grid(True)
    plt.show()


#score_count()

results_interpretation()

#score_analise()
```

“Social graph.py”

```
# === Підключення бібліотек ===
import ast
import pandas as pd
from pyvis.network import Network

# === Завантаження даних ===
df = pd.read_csv("users_score_1.csv")

# === Налаштування Network ===
net = Network(height='800px', width='100%', directed=True, notebook=False,
bgcolor='#f0f0ff')
net.force_atlas_2based(gravity=-50, central_gravity=0.02, spring_length=200)


# === Отримаємо колір в залежності від score ===
def get_color_by_score(score):

    if score < 0.2:
        return "#00cc44"  # зелений
    elif score < 0.4:
        return "#99cc00"  # жовто-зелений
    elif score < 0.6:
        return "#ffcc00"  # жовтий
    elif score < 0.8:
        return "#ff6600"  # помаранчевий
    else:
        return "#cc0000"  # червоний


# === Додавання вузла ===
def add_node_if_not_exists(user_id):

    if user_id in net.get_nodes():
        return

    user_row = df[df['id'] == user_id]
    if user_row.empty:
        net.add_node(user_id, label=str(user_id), color='gray', size=10)
        return

    user_row = user_row.iloc[0]
    name = user_row['name']
    score = float(user_row['score']) if pd.notna(user_row['score']) else 0
    followers = ast.literal_eval(user_row['followers']) if
pd.notna(user_row['followers']) else []
    size = 10 + len(followers)
    color = get_color_by_score(score)

    net.add_node(user_id, label=name, color=color, size=size)


# === Додаємо користувача + друзів/підписників + глибина ===
def add_user_and_connections(user_id, depth, max_depth, visited):

    if depth > max_depth or user_id in visited:
        return
```

```
    visited.add(user_id)

    add_node_if_not_exists(user_id)

    user_row = df[df['id'] == user_id]
    if user_row.empty:
        return
    user_row = user_row.iloc[0]

    friends = ast.literal_eval(user_row['friends']) if pd.notna(user_row['friends'])
else []
    for f in friends:
        add_node_if_not_exists(f)
        net.add_edge(user_id, f, color='green', title='friend')
        add_user_and_connections(f, depth+1, max_depth, visited)

    followers = ast.literal_eval(user_row['followers']) if
pd.notna(user_row['followers']) else []
    for f in followers:
        add_node_if_not_exists(f)
        net.add_edge(f, user_id, color='orange', title='follower')
        add_user_and_connections(f, depth+1, max_depth, visited)


# === Задаєм початкові параметри ===
start_user_id = "e4e7ae77-566f-485d-b494-89f30d06c7a3"
max_depth = 0

visited_nodes = set()
add_user_and_connections(start_user_id, depth=0, max_depth=max_depth,
visited=visited_nodes)

# === Виводимо результат та зберігаємо ===
net.show("social_graph_with_score.html", notebook=False)
net.write_html("graph/my_graph.html")
```