

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»

ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Кафедра інформаційної безпеки

«На правах рукопису»

«До захисту допущено»

УДК 004.056

В.о. завідувача кафедри

\_\_\_\_\_ М.В.Грайворонський

“ \_\_\_\_ ” \_\_\_\_\_ 2019 р.

**Магістерська дисертація**

**на здобуття ступеня магістра**

зі спеціальності: 125 Кібербезпека

на тему: Механізм виявлення інформаційних впливів між агентами в соціальних мережах

Виконав (-ла): студент (-ка) 2 курсу, групи ФБ-81мп

(шифр групи)

Левченко Артем Ігорович \_\_\_\_\_

(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник Ткач Володимир Миколайович \_\_\_\_\_

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Рецензент \_\_\_\_\_

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській дисертації  
немає запозичень з праць інших авторів без  
відповідних посилань.

Студент \_\_\_\_\_

(підпис)

Київ – 2019 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**  
**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**

**імені ІГОРЯ СІКОРСЬКОГО»**  
**ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ**

Кафедра інформаційної безпеки

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою  
Спеціальність (освітньо-професійна програма) – 125 Кібербезпека («Системи, технології та математичні методи кібербезпеки»)

**ЗАТВЕРДЖУЮ**

В.о. завідувача кафедри

\_\_\_\_\_ М.В.Грайворонський  
(підпис)

«\_\_\_» \_\_\_\_\_ 2019 р.

**ЗАВДАННЯ**

**на магістерську дисертацію студенту**

Левченко Артем Ігорович

(прізвище, ім'я, по батькові)

1. Тема дисертації: Механізм виявлення інформаційних впливів між агентами в соціальних мережах

\_\_\_\_\_  
науковий керівник дисертації: Ткач Володимир Миколайович,  
к.е.н, доцент кафедри інформаційної безпеки,

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

затверджені наказом по університету від «15» листопада 2019 р. № 3927-с

2. Термін подання студентом дисертації 10.12.2019 р.

3. Об'єкт дослідження: сучасні соціальні мережі, їх учасники, вплив між учасниками

4. Вихідні дані: власна модель моніторингу впливів між користувачами соціальних мереж, її програмна реалізація

5. Перелік завдань, які потрібно розробити: розробити модель моніторингу впливів між учасниками соціальної мережі, запропонувати її програмну реалізацію

6. Орієнтовний перелік ілюстративного матеріалу \_\_\_\_\_

\_\_\_\_\_

7. Орієнтовний перелік публікацій \_\_\_\_\_

\_\_\_\_\_

8. Дата видачі завдання \_\_\_\_\_

#### Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка

Студент \_\_\_\_\_  
(підпис)

Левченко А. І.  
(ініціали, прізвище)

Науковий керівник дисертації \_\_\_\_\_  
(підпис)

Ткач В. М.  
(ініціали, прізвище)

## РЕФЕРАТ

Робота обсягом 97 сторінок містить 18 ілюстрацій, 22 таблиці та 20 літературних посилань.

Метою даної кваліфікаційної роботи є розробка механізму виявлення впливів між агентами в соціальних мережах, на його підставі розробка моделі, яка описує взаємозв'язок між користувачами соціальної мережі, впливи між ними і нарешті програмна реалізація системи моніторингу впливів в соціальних мережах.

Об'єктом дослідження є сучасні соціальні мережі, їх учасники.

Предметом дослідження є вразливість учасників соціальних мереж до сторонніх впливів направлених на зміну думки учасника або маніпуляцію цим учасником.

Результати роботи викладені у вигляді моделі впливів між учасниками соціальних мереж та її програмної реалізації.

Результати роботи можуть бути використані для розробки нових або впровадження в існуючі системи моніторингу впливів між учасниками соціальних мереж.

Ключові слова: вплив в соціальних мережах, модель впливів в соціальних мережах, інформаційна безпека в соціальних мережах

## РЕФЕРАТ

Работа объёмом 97 страниц содержит 18 иллюстраций, 22 таблицы и 20 литературных ссылок.

Целью данной квалификационной работы является разработка механизма выявления влияний между агентами в социальных сетях, на его основании разработка модели, описывающей взаимосвязь между пользователями социальной сети, влияния влияний между ними и наконец программная реализация системы мониторинга влияний в социальных сетях.

Объектом исследования являются современные социальные сети, их участники.

Предметом исследования является уязвимость участников социальных сетей к посторонним влияниям направленных на изменение мнения участника или манипуляцию этим участником.

Результаты работы выложены в виде модели влияния между участниками социальной сети и ее программной реализации.

Результаты работы могут быть использованы для разработки новых или для внедрения в существующие системы мониторинга влияний между участниками социальных сетей.

Ключевые слова: влияние в социальных сетях, модель воздействий в социальных сетях, информационная безопасность в социальных сетях

## ABSTRACT

The 65-page work contains 97 illustrations, 22 tables and 20 literary references.

The purpose of this qualification work is to develop a mechanism for identifying influences between agents in social networks, on its basis, to develop a model describing the relationship between users of a social network, the influence between them, and finally to develop the software implementation of a system of monitoring influences in social networks.

The object of the research are modern social networks, their participants.

The subject of the research is the vulnerability of social network participants to third-party influences aimed at changing the participant's opinion or manipulating that participant.

The results of the work are presented in the form of a model of influence between the participants of the social network and its program implementation.

The results of the work can be used to develop new ones or to introduce into existing systems of monitoring of influence between participants of social networks.

Keywords: influence in social networks, model of influences in social networks, information security in social networks

## ЗМІСТ

Зміст.....	1
Перелік умовних позначень, символів, одиниць, скорочень і термінівс .....	9
Вступ.....	10
1 Вплив в соціальних мережах .....	12
1.1 Вплив.....	12
1.2 Інформаційна безпека і довіра. Астротурфінг.....	13
1.3 Соціальні мережі. Історія розвитку. Соціометрія.....	16
1.3 Інформація в соціальних мережах .....	19
1.3.1 Інформація, яку можна отримати в відкритому доступі .....	19
1.3.2 Інформація, яку не можна отримати в відкритому доступі .....	22
Висновки до розділу 1 .....	22
2 Теорія впливу в соціальних мережах. Моделі впливу та індекси впливовості ...	24
2.1 Моделі впливу .....	24
2.1.1 Моделі з порогамі.....	25
2.1.2 Моделі незалежних каскадів .....	26
2.1.3 Моделі на основі клітинних автоматів.....	28
2.1.4 Ланцюги Маркова.....	30
2.2 Індекси впливовості.....	32
Висновки до розділу 2 .....	34
3 Розробка моделі впливів в соціальній мережі .....	36
3.1 Розробка моделі .....	36
3.2 Програмна реалізація .....	42
3.3 Збір даних та схема бази .....	43
3.3.1 Колекція «posts» .....	44
3.3.2 Колекція «reactors» .....	45
3.3.2 Колекція «comments» .....	45
3.3.3 Колекція «reshares».....	46
3.4 Аналіз і візуалізація даних.....	46
3.5 Експеримент по відслідковуванню змін думок агентів.....	49
3.6 Виявлення агентів з найбільшим впливом .....	54

Висновки до розділу 3 .....	56
4 Розробка стартап-проекту .....	57
4.3 Опис ідеї стартап-проекту .....	57
4.4 Технологічний аудит ідеї проекту .....	60
4.5 Аналіз ринкових можливостей запуску стартап-проекту .....	61
4.6 Розробка ринкової стратегії проекту .....	71
4.7 Розробка маркетингової програми стартап-проекту.....	74
Висновки до розділу 4 .....	77
Висновки .....	79
Список джерел посилання.....	80
Додаток А Код програмної реалізації .....	82



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

Астротурфінг (англ. astroturfing) — це створення штучної громадської думки за допомогою гласних чи негласних заходів, форм та методів впливу зацікавленими іноземними спеціальними службами, окремими організаціями, групами та особами, що використовують програмне забезпечення або наймають представників засобів масової інформації, блогерів, інтернет-коментаторів, спеціалістів з метою витіснення думки реальних людей і створення враження, наче велика кількість людей вимагає чогось конкретного або виступає проти чого-небудь.

Аналіз тональності тексту (сентимент-аналіз, англ. Sentiment analysis, англ. Opinion mining[1]) — клас методів контент-аналізу в комп'ютерній лінгвістиці, призначений для автоматизованого виявлення в текстах емоційно забарвленої лексики і емоційної оцінки авторів (думок) по відношенню до об'єктів, мова про які йде в тексті.

ПЕОМ – персональна електронна обчислювальна машина;

ПК – персональний комп'ютер;

XML – eXtensible Markup Language — розширювана мова розмітки;

ПЗ – Програмне забезпечення;

СУБД (СУБД, СКБД англ. Database Management System, DBMS) — набір взаємопов'язаних даних (база даних) і програм для доступу до цих даних.[1] Надає можливості створення, збереження, оновлення та пошуку інформації в базах даних з контролем доступу до даних.

HTTP – Hypertext Transfer Protocol.

API – прикладний програмний інтерфейс (інтерфейс програмування застосунків, інтерфейс прикладного програмування) (англ. Application Programming Interface)

## ВСТУП

Багато сучасних дослідників вважають одним з найбільших феноменом Всесвітньої павутини утворення віртуальних спільнот. Ці спільноти мають свої специфічні регулятори соціальної поведінки, особливі форми спілкування, інтереси, цінності і мову, які обумовлені приналежністю учасників до певних реальних груп. Володіючи рядом загальних для соціальної групи ознак, віртуальне співтовариство відрізняється своїми особливими соціально психологічними характеристиками, які відносно недавно стали предметом окремих досліджень.

Одним з розглядуваних аспектів функціонування віртуальних спільнот є залученість в так звані інформаційні війни, в яких дані спільноти стають об'єктом не тільки міжгрупового, а й міждержавного інформаційного протистояння. В наш час практично будь-який конфлікт знаходить своє відображення в мережі Інтернет, і часто від образу в цьому «дзеркалі» залежить кінцевий результат протистояння конкуруючих сторін. Активне залучення в віртуальне середовище багатомільйонної аудиторії, формування громадської думки і маніпулювання нею дозволяє глобально впливати на процеси сторін, що конкурують.

Актуальність роботи зумовлена тим, що соціальні мережі стали головним майданчиком по поширенню інформаційного впливу, що в свою чергу створює загрозу маніпуляції людською свідомістю. В сучасному світі служби соціальних мережі стають одним з ключових чинників динаміки розвитку соціальних відносин. Немає сумнівів, що вплив цих мереж на протікання процесів у суспільстві постійно зростає. З розвитком соціальних мереж кожна людина отримала ще більше доступу та можливість стати джерелом поширення інформації, стати джерелом впливу не тільки в колі своїх друзів та знайомих, а в безкрайньому просторі Інтернету.

В епосі соціальних мереж, в якій ми зараз опинилися, дуже важливо встигати слідкувати за потоками інформації, адже вони описують події які відбулися, або провокують події які можуть відбутися.

Аналіз впливів в соціальних мережах дає можливість спостерігати перебіг будь-яких, а особливо соціально-громадських подій в реальному середовищі. Для проведення такого аналізу та спостережень необхідний кваліфікований персонал та програмні засоби для автоматизації процесу. Як не дивно в Україні не так багато компаній почало займатись розробками та дослідженнями в цьому напрямку, а отже є шанси стати розробниками-піонерами таких систем.

Метою роботи є розробка, реалізація та впровадження методу виявлення впливів між агентами в соціальних мережах.

Завданням роботи є реалізація системи, яка була б здатна відслідковувати динаміку впливів в соціальних мережах. Для досягнення цієї мети треба, по-перше сформулювати вимоги до компонентів системи, по-друге розробити алгоритм за яким буде працювати система.

Об'єктом дослідження є сучасні соціальні мережі, їх учасники, вплив між учасниками.

Предметом дослідження є вразливість учасників соціальних мереж до сторонніх впливів направлених на зміну думки учасника або маніпуляцію цим учасником.

Наукова новизна роботи полягає в пропозиції власної моделі виявлення впливів між користувачами соціальних мереж.

Практичне значення полягає в тому, що результати роботи можуть бути використані для розробки нових або впровадження в існуючі системи моніторингу впливів між учасниками соціальних мереж.

# 1 ВПЛИВ В СОЦІАЛЬНИХ МЕРЕЖАХ

## 1.1 Вплив

Вплив - процес і результат зміни агентом (суб'єктом впливу) поведінки іншого агента (індивідуального або колективного об'єкта впливу), його установок, намірів, уявлень і оцінок (а також дій, що ґрунтуються на них) в ході взаємодії з ним [1]. Вплив – здатність впливати на чийсь уявлення або дії [2].

Розрізняють спрямований і неспрямований вплив [1]. Спрямований (цілеспрямований) вплив - це вплив, що використовує в якості механізмів впливу на іншого суб'єкта переконання і навіювання. При цьому суб'єкт впливу ставить перед собою завдання домогтися певних результатів (наприклад, вибору певних дій) від об'єкта впливу. Неспрямований (нецілеспрямований) вплив – це вплив, при якому індивід не ставить перед собою завдання домогтися певних результатів від об'єкта впливу.

Як показують спостереження психологів [3], в соціальній мережі агенти часто не мають достатньої для прийняття рішень інформації або не можуть самостійно обробити її, тому їх рішення можуть ґрунтуватися на спостережуваних ними рішеннях або уявленнях інших агентів (соціальний вплив).

Соціальний вплив реалізується в двох процесах: комунікації (в ході спілкування, обміну досвідом та інформацією, обговорення тих чи інших питань з авторитетними для агента сусідами він приходить до певних уявлень, установок, думок) і порівняння (у пошуках соціальної ідентичності та соціального схвалення агент приймає уявлення і дії, очікувані від нього іншими агентами в даній ситуації; агент задається питанням «що б зробив інший агент (еталон для порівняння), будь він у моїй ситуації?» і, порівнюючи себе з ним, визначає свою адекватність і грає відповідну роль. Порівняння можна пояснити і пошуком стратегічної переваги:

порівнюючи себе з іншими агентами, які займають ті ж позиції в соціальній системі, агент може ввести або прийняти нововведення, які зроблять його більш привабливим в якості об'єкту відносин). Необхідно відзначити, що при комунікативному підході до впливу агенти можуть прийти до схожих уявлень, але не обов'язково до подібної поведінки. При порівнянні агент зазвичай непрямим чином копіює поведінку. Очевидно, поведінка агента визначається не тільки уявленнями, а й обмеженнями, з якими він стикається. Тому агенти зі схожими уявленнями можуть вести себе по-різному, і навпаки, агенти з різними уявленнями можуть вести себе однаково.

## **1.2 Інформаційна безпека і довіра. Астротурфінг**

Відповідно до словника з соціальної психології під редакцією М. Ю. Кондратьєва, довіра - це специфічне ставлення суб'єкта до певних об'єктів, пов'язаних з ситуативною, актуальною значущістю і апріорної надійністю (безпекою) об'єкта для суб'єкта [1].

У системах, побудованих на довірі, існує ризик бути обманутим суб'єктом, з яким об'єкт вступив в довірчі відносини. Отже, чим вище ймовірність передачі даних третім особам, тим вище ризик початку відносин з даним об'єктом - тим менше до них довіри.

Масштабні соціальні мережі стали надзвичайно популярні завдяки стрімкому розвитку мережевих і мобільних технологій [1, 2], проте ця популярність неминуче притягує зловмисників, які прагнуть використовувати велику призначену для користувача аудиторію для поширення шкідливих програм, крадіжки персональних даних або маніпулювання громадською думкою. Допмагаючи підтримувати зв'язок з близькими, друзями і колегами, соціальні мережі одночасно створюють ризики безпеки. Деякі соціальні мережі, наприклад Facebook, LinkedIn і Google+, вимагають, щоб користувачі вказували про себе реальні відомості, і

можуть зберігати великий обсяг персональної інформації, в тому числі дані по кредитних картах.

Для залучення цільової аудиторії і маніпуляції нею все більш широко застосовується астротурфінг (astroturfing) - штучне формування громадської думки або емуляція громадської підтримки шляхом масштабного коментування в середовищі інтернет-комунікацій.

Застосовується для витіснення думки реальних людей на веб-форумах або соціальних мережах, для організації підроблених кампаній в Інтернеті, які створюють враження, що велика кількість людей вимагають чогось конкретного або виступають проти чого-небудь.

Астротурфінг властиві певні інструменти:

- створення бригад для розміщення повідомлень в соціальних мережах, чатах, форумах, новинах, для просування або, навпаки, для осміювання (нерідко за допомогою троллінгу) певних ідей;

- поширення в Мережі ідей, які будуть підтримані і ретранслювати більшістю користувачів;

- «серфінг» - технічна «накрутка» лічильника переглядів тієї чи іншої інформації для підняття її в рейтингу.

Астротурфінг може використовувати спеціально найнятих людей, які на форумах, в блогах і соціальних мережах пишуть замовні пости і коментарі з метою створення враження спонтанної масової підтримки або осуду заданих реалій (товару, продукту, сервісу, події, точки зору та ін.).

Так, ще в 2008 році, за даними британської газети The Guardian, в Китаї налічувалося 300 000 астротурферів, завданням яких була організація протистояння негативної інформації про КНР, що з'явилася, зокрема, після оприлюднення фактів масового отруєння дітей китайськими молочними продуктами. Підвищення ефективності астротурфінгу було реалізовано на основі

розробки мережі ботів - спеціальних програм, що виконують автоматично або за заданим розкладом будь-які дії через ті ж інтерфейси, що і звичайний користувач.

Сьогодні астротурфінг використовується не тільки корпоративними лобістами, вірусними маркетологами або політтехнологами, але також і спецслужбами для маніпулювання учасниками соціальних мереж і цілими мережевими співтовариствами, які налічують вже сотні мільйонів людей. У зв'язку з новими завданнями, які ставляться перед астротурфінгом, і новими можливостями організаторів астротурфінгових проектів він переходить на більш високий технологічний рівень, в якому інтегруються найпотужніші програмні засоби з людським потенціалом. Згідно з американським політичним блогу Daily Kos, астротурфінгові компанії зараз мають у своєму розпорядженні інформаційними технологіями для управління персональними акаунтами (обліковими записами), які надають мультиплікаційного ефекту в відношенні зусиль кожного астротурфера, створюючи враження, що присутня велика підтримка того, що корпорація або уряд намагаються зробити. Це програмне забезпечення автоматично генерує всі оточення, яким повинен володіти справжня людина в Інтернеті: ім'я, адреса e-mail, веб-сторінки, активність в соціальних мережах. Фальшиві акаунти можуть підтримуватися і оновлюватися за допомогою автоматичної передруку постів (записів в блогах) або посилань на контент, опублікований на інших мережних майданчиках, щоб посилити враження, що власники цього акаунта реальні і активні. Астротурферам можуть присвоюватися попередньо «зістарені» акаунти для створення ним неправдивих мережних історій. Подібне цьому програмне забезпечення потенційно здатне знищити Інтернет як майданчик для конструктивного обміну думками. Воно ставить під загрозу саме поняття онлайн-демократії. Всі перераховані вище елементи є додатком більш глобальних інструментів. Йдеться про різні методологічних проектах, результатом роботи яких є народження нових ідеологій, смислів і цілей суспільства. Надалі ці ідеї підхоплюються і тиражуються широким колом громадськості (без істотної технічної підтримки через астротурфінг). Однак розробки ідеологій - доля

глобальних гравців, оскільки є ресурсоемними. Вони використовуються в геополітичних проектах або ж транснаціональними корпораціями.

### 1.3 Соціальні мережі. Історія розвитку. Соціометрія

Під соціальною мережею на якісному рівні розуміється соціальна структура, що складається з безлічі агентів (суб'єктів - індивідуальних чи колективних, наприклад, індивідів, сімей, груп, організацій) і певного на ньому безлічі відносин (сукупності зв'язків між агентами, наприклад, знайомства, дружби, співпраці, комунікації). Формально соціальна мережа являє собою граф  $G(N, E)$ , в якому  $N = \{1, 2, \dots, n\}$  – безліч вершин (агентів) і  $E$  - безліч ребер, що відображають взаємодію агентів.

Соціальні мережі сприяють, по-перше, організації соціальних комунікацій між людьми і, по-друге, - реалізації їх базових соціальних потреб. Можна виділити дві пересічних трактування соціальної мережі - як соціальної структури і її специфічної інтернет-продажу.

Техніка соціометрії (опису соціальних груп в термінах теорії графів) була вперше запропонована і розвинена в роботах Дж. Морено. Термін «соціальна мережа» був введений в 1954 році соціологом Джеймсом Барнсом в [19], але масове поширення (не тільки серед вчених-соціологів) отримав з початку 2000-х років з розвитком відповідних інтернет-технологій. В даний час, як справедливо відзначається в [20, 21], відчувається гострий дефіцит систематичного викладу методів і алгоритмів мережевого аналізу, придатних для сучасних прикладних досліджень.

Узагальнюючи причини привабливості соціальних мереж, можна виділити наступні надані ними користувачам можливості:

- отримання інформації (в тому числі виявлення ресурсів) від інших членів соціальної мережі;
- верифікація ідей через участь у взаємодіях в соціальній мережі;



- соціальна вигода від контактів (причетність, самоідентифікація, соціальне ототожнення, соціальне прийняття і ін.);
- рекреація (відпочинок, проведення часу).

Вже давно прослідковувався потенціал комп'ютерних мереж для полегшення соціальної взаємодії. Спроби підтримки соціальних мереж за допомогою комп'ютерно-опосередкованої комунікації були зроблені багатьма онлайн-сервісами, які з'явилися на ранніх етапах розвитку, в тому числі Usenet, ARPANET, LISTSERV, і дошки оголошень (BBS). Багато прототипів з функціональністю соціальних мереж також були присутні в онлайн-сервісах, таких як America Online, Prodigy, CompuServe, ChatNet та The WELL. На першому етапі появи соціальних мереж вони були у вигляді узагальнених онлайн співтовариств, таких як Theglobe (1995), Geocities (1994) і Tripod.com (1995). Багато з цих ранніх співтовариств були зосереджені на групуванні людей разом, аби взаємодіяти один з одним через чати, і закликали користувачів ділитися особистою інформацією та ідеями за допомогою особистих веб-сторінок, надаючи прості у використанні інструменти публікації та безкоштовний або недорогий веб-простір. Деякі співтовариства, наприклад як Classmates.com - обрали інший підхід, просто пов'язавши людей один з одним за допомогою їх адрес електронної пошти. Наприкінці 1990-х, профілі користувачів стали центральним елементом соціальних мереж, що дозволяє користувачам складати списки «друзів» і шукати інших користувачів зі схожими та спільними інтересами. Нові методи соціальних мереж були розроблені до кінця 1990-х років, і багато сайтів почали надавати більш складні функції та підходи для користувачів, щоб знаходити та керувати списками друзів. Нове покоління соціальних мереж почало процвітати з появою SixDegrees.com в 1997 році, а потім Makeoutclub в 2000 році, Hub Culture та Friendster в 2002 році, і незабаром став частиною інтернет-мейнстріму. За Friendster, через рік по слідували MySpace і LinkedIn, і врешті-решт Bebo. Friendster стала дуже популярною на Тихоокеанських островах. Orkut став першою соціальною мережою в Бразилії і також швидко популяризувався в Індії.

Свідченням становлення соціальних мереж як невід'ємної частини Інтернету став той факт що в 2005 на MySpace було зафіксовано більше переглядів сторінок ніж у Google . Facebook , запущений в 2004 році, на початку 2009 року став найбільшою соціальною мережею. Він був вперше представлений як мережа для спілкування Гарвардського університету.

До України мода на соціальні мережі прийшла двома роками пізніше – в 2006-му, з появою Однокласників і Вконтакте.

Соціальні інтернет-мережі дозволяють запропонувати своїм користувачам якісно новий рівень представлення інформації, коли на одній сторінці зводяться дані різного типу і з різних джерел з можливістю для користувача вибору форми їх подання (текстова, облікова, таблична, графічна); Ви завжди маєте простий в сприйнятті, але збагачений (інтерактивними картами, відеоматеріалами та т. п.) репортаж з можливістю ведення пошуку, отримання додаткових довідкових даних, фільтрації контенту і висловлення власної думки. Все це створює переконливі ілюзії об'єктивності наданої інформації та необмеженої свободи одержувача в оперуванні вихідними даними.

Наявний в найбільш популярних соціальних мережах інструментарій по публікації інформації і організації спільнот, а саме: можливість створення груп, своєї порядку денного, публікації тексту, фото-, аудіо- або відео-файлів, а також, з огляду на зростаючу частку мобільного ринку соціальних програм, оперативність , з якої можна надати дані величезній кількості людей, - все це дає свободу в повній мірі застосовувати методи ведення інформаційних кампаній в соціальних мережах.

У сучасному інформаційному середовищі в умовах дефіциту уваги висунення на перший план тієї чи іншої тематики повідомлення є найсильнішим засобом маніпуляції цільовою аудиторією. Конкретні позиції читача по відношенню до певних питань, які піднімаються по конкретній темі часто мають зовсім мале значення в порівнянні з самим фактом спрямованості аудиторії саме до цієї теми і відповідно ігнорування в даний момент будь-якої іншої тематики.

### **1.3 Інформація в соціальних мережах**

Як вже стало зрозуміло, з такою популярністю мереж, та ростом кількості користувачів вже не є можливим та раціональним затрачувати людський ресурс для поглядання персональних сторінок в мережі, з метою пошуку певної інформації, яка може нести загрозу. А отже настав час проводити автоматизацію цього процесу. Не дивно що вже зараз велика кількість різних державних структур, крупних корпорацій, невеличких приватних фірм й навіть деякі команди ентузіастів зрозуміли всю важливість ролі соціальних мереж у формуванні інформаційного поля як окремо взятої людини так і суспільства в цілому й розробляють найрізноманітніші інструменти для ефективного дослідження та аналізу мереж з точки зору їх впливу на формування громадської думки. Але, як згадувалось раніше, в Україні цей процес пішов не так швидко.

Результати дослідів вказують на значний вплив великих соціальних груп всередині мереж на формування поточної громадської думки. Буквально за хвилини від моменту створення посту в групі кількість людей, що діляться нею зі своїми друзями, знайомими може зрости до кількох сотень або навіть тисяч, а в подальшому й до близько десяти тисяч людей. Звісно ці цифри сильно залежать від змісту посту, його інтересу для публіки, моменту появи, можливо провокативності. Крім того такі щосекундні «новини» забуваються настільки ж швидко як й набувають свою популярність, але все ж таки ці масштаби та темпи вражають. Й у випадку достатньо тривалого, масового й концентрованого просування однієї теми засобами таких груп у мережах це може призвести до досить суттєвих наслідків.

#### **1.3.1 Інформація, яку можна отримати в відкритому доступі**

Кожен користувач соціальних мереж, які існують сьогодні, має можливість закрити інформацію про себе від сторонніх користувачів у рамках можливостей,

що надає йому сервіс соціальної мережі. Але, як би дивно це не було, дуже багато людей не приділяють потрібної уваги питанню про безпеку інформації про себе.

Багато людей не розуміють, що інформація, розміщена ними в соціальних мережах, може бути знайдена і використана ким завгодно, не обов'язково з благими намірами. Інформацію про учасників соціальних мереж можуть знайти їх роботодавці, родичі, збирачі боргів, злочинці і так далі. Судові пристави іноді використовують соціальні мережі, щоб знайти неплатників або отримати відомості про їхнє майно.

На прикладі соціальної мережі «Facebook», якщо не стоїть обмежень, то у відкритому доступі можна отримати:

1. Загальна інформація
  - 1.1 ПІБ
  - 1.2 Дата народження та вік
  - 1.3 Сімейний стан
  - 1.4 Список родичів
2. Контакти користувача
  - 2.1 Країна
  - 2.2 Місце народження
  - 2.3 Місце проживання
  - 2.4 Мобільний телефон
  - 2.5 Домашній( додатковий) телефон
  - 2.6 Скайп
  - 2.7 Посилання на особистий сайт
  - 2.8 Посилання на сторінки в інших соціальних мережах
3. Інтереси користувача (від улюбленої книги, до виду діяльності)
4. Освіта користувача
  - 4.1 Місце та роки отримання шкільної освіти
  - 4.2 Місце та роки отримання вищої освіти
5. Кар'єра
  - 5.1 Місце та роки роботи

- 5.2 Посада
- 6. Служба
  - 6.1 Країна
  - 6.2 Військова частина
- 7. Життєва позиція
  - 7.1 Політичні переконання
  - 7.2 Світогляд
  - 7.3 Що для користувача є головним в житті
  - 7.4 Що для користувача є головним в людях
  - 7.5 Ставлення до паління/алкоголю
- 8. Фотографії та відеозаписи користувача
- 9. Друзі користувача
- 10. Групи та сторінки з підписки

І це лишень інформація, яку користувач сам про себе заповнює. Ще більше інформації можна отримати з груп, та зі сторінок друзів:

- 1. Діалоги та суперечки з іншими людьми
- 2. Записи на чужих сторінках
- 3. «Мені сподобалось» до записів

Найбільш інформативною є, так звана, «стіна» користувача. З неї можна отримати таку інформацію:

- 1. Записи самого користувача
- 2. Репости записів інших користувачів або груп
- 3. Записи на «стіні» від інших користувачів
- 4. Коментарі до записів
- 5. «Мені сподобалось» на стіні

А ця можливість несе набагато більше інформації про людину. За цими записами можна судити за характером людини, за його інтересами та зміною поведінки.

### **1.3.2 Інформація, яку не можна отримати в відкритому доступі**

Звичайно є велика кількість користувачів які розуміють що викладати стільки інформації про себе у відкритий доступ є, як мінімум, не розумно. А отже вони використовують можливості аби закрити інформацію від сторонніх користувачів. Або взагалі не заповнюють частину анкети.

Також не представляється можливим без доступу до бази даних мережі добути інформацію про перегляди користувачем публікацій, крім випадків, коли користувач власноруч публічно не відреагував на публікацію в будь-який передбачений розробниками спосіб (реакція, коментар, тощо). Досить суб'єктивною і приблизною може виявитися інформація щодо перебування користувача в режимі «онлайн», адже щоб ефективно збирати подібні данні необхідно моніторити в «режимі реального часу» кожного учасника мережі може бути технічно нездійсненною задачею, а дані зібрані в ретроспективі не достатньо для точного підрахунку.

Найбільш конфіденційна інформація – переписки користувача, звичайно, лишається закритою.

## **Висновки до розділу 1**

В цьому розділі були висвітленні поняття впливу в соціальних мережах, самої соціальної мережі, астротурфінгу як інструментом для маніпуляції громадською думкою в соціальних мережах. Піднято питання про загрозу яку може нести астротурфінг для рядових користувачів та для Інтернету цілому.

Вже пройшло багато часу з моменту як соціальні мережі стали одним з домінуючих інструментів для спілкування, обміну та отримання інформації користувачами. Але водночас з цим з'явилась небезпека зловживанням цим інструментом комунікації шляхом маніпуляції людською довірою до наданої інформації. Постає потреба у виявленні цих маніпуляцій, тобто задача у виявленні

та представленні у зручному для обробки вигляді впливів між учасниками соціальною мережі, з метою недопуску зловживання людською довірою. Ця задача вирішується шляхом модулювання соціальної мережі, поведінки її користувачів, виявленню впливів між ними. В наступному розділі буде розглянуто існуючі популярні моделі для виявлення впливів, сформовано критерії і властивості, якими повинна володіти мережа, щоб ефективно відображати впливи в соціальних мережах.

## 2 ТЕОРІЯ ВПЛИВУ В СОЦІАЛЬНИХ МЕРЕЖАХ. МОДЕЛІ ВПЛИВУ ТА ІНДЕКСИ ВПЛИВОВОСТІ

### 2.1 Моделі впливу

Соціальна мережа відіграє велику роль в поширенні інформації, ідей і впливів між її членами. При моделюванні соціальних мереж, взаємовпливу їх членів (агентів), динаміки їх думок та ін. виникає необхідність врахування чинників (ефектів), що мають місце в реальних соціальних мережах. В цілому, в реальному світі соціальні мережі мають такі характеристики і властивості, які обумовлені характеристиками і потребами агентів (що впливають і піддаються впливу), характером їх взаємодії, а також властивостями самої соціальної мережі:

- 1) наявність власних думок у агентів;
- 2) розподіл агентів по групах ( «за інтересами», зі схожими думками);
- 3) зміна думок під впливом інших членів соціальної мережі;
- 4) стійкість думки агента (різна ступінь схильності агентів до впливу);
- 5) формалізація індексів впливу (вплив кожного агента можна виміряти);
- 6) існування «лідерів думок» (агентів з максимальним «впливом»);
- 7) існування непрямого впливу (через ланцюжок спільних знайомих);
- 8) існування (зазвичай менш значущих) зовнішніх чинників впливу (реклама, маркетингові акції) і, відповідно, зовнішніх агентів (засоби масової інформації, виробники товарів та ін.);

Перераховані ефекти і властивості повинні знаходити відображення в моделях, які претендують на адекватний опис реальних соціальних мереж. Цей розділ присвячений аналітичному огляду вже існуючих моделей соціальних мереж та встановленням їх відповідності перерахованим вище властивостям.



### 2.1.1 Моделі з порогами

Ця модель разом із «Моделлю незалежних каскадів» відноситься до класу *дифузних* моделей і є найбільш популярними і дослідженими серед всіх моделей соціальних мереж.

Агент - вузол соціальної мережі (вершина графа) - може перебувати в активному і неактивному станах, причому можливий перехід тільки з неактивного стану в активне (зворотний перехід не допускається). Особливістю цієї моделі є властивість, що вірогідність кожного агента стати активним монотонно збільшується зі збільшенням числа активних сусідів агента. Таким чином, процес буде виглядати приблизно так, як це виглядає з точки зору початково неактивного вузла  $i$ : з плином часу активізуються дедалі більше сусідів  $i$ ; в якийсь момент це може призвести до активізації  $i$ , а рішення  $i$  може, в свою чергу, може в подальшому вплинути на рішення вузлів, з якими  $i$  зв'язаний.

Granovetter та Schelling були одними з перших, хто запропонував моделі, що фіксують такий процес; їх підхід базувався на використанні конкретних порогових значень. Багато моделей цього формату були досліджені (див., Наприклад, [6, 9]), але базова *Лінійна Порогова Модель* лежить в основі більшості подальших модифікацій. У цій моделі на вузол  $i$  впливає кожен сусід  $j$  відповідно до ваги  $w_{ij}$  так, що

$$\sum_{j \text{ активний вузол-сусід } i} w_{ij} \leq 1 \quad (2.1)$$

Потім динаміка процесу протікає наступним чином. Кожен вузол  $i$  вибирає поріг  $\theta_i$  рівномірно випадково з інтервалу  $[0, 1]$ ; це являє собою зважену частку сусідів  $i$ , які повинні стати активними для того, щоб  $i$  став активним. Враховуючи випадковий вибір порогових значень та початковий набір активних вузлів  $A_0$  (при

всіх інших вузлах неактивних), процес дифузії детерміновано розгортається на дискретні етапи: на етапі  $t$  всі вузли, які були активними на етапі  $t - 1$ , залишаються активними, тоді ми активуємо будь-який вузол  $i$ , для якого загальна вага його активних сусідів становить не менше  $\theta_i$ :

$$\sum_{j \text{ активний вузол-сусід } i} w_{ij} \geq \theta_i \quad (2.2)$$

Таким чином, пороги  $\theta_i$  інтуїтивно представляють різні приховані тенденції вузлів сприймати нововведення, коли це роблять їхні сусіди; той факт, що вони вибрані випадковим чином, призначений для моделювання нашого незнання їх значень - ми насправді усереднюємо можливі порогові значення для всіх вузлів.

Тобто, якщо в даній моделі агент  $i$  зазнає впливу  $w_{ij}$  кожного свого  $j$ -го сусіда в мережі так, що виконується умова  $\sum_{j \text{ активний вузол-сусід } i} w_{ij} \leq 1$ , то  $i$  стає активним в залежності від обраного ним порогу  $\theta_i \in [0; 1]$ , тобто маємо умову активації (2.2).

У статті [6] пропонується розширення моделі на основі введення нелінійних граничних функцій; в роботі [7] каскадні (лавиноподібні) ефекти вивчаються з точки зору топології мережі.

### 2.1.2 Моделі незалежних каскадів

Концептуально найпростіша модель цього типу - це те моделі незалежних каскадів (*Independent Cascade Model*) досліджена в контексті маркетингу Голденбергом, Лібаєм та Мюллером [8]. Моделі незалежних належать до категорії моделей так званих «систем взаємодіючих частинок» (*Interacting Particle Systems*). Вузол мережі (агент) визначається аналогічно вищеописаної моделі. Ми знову

починаємо з початкового набору активних вузлів  $A_0$ , і процес поділяється на дискретні етапи відповідно до наступного рандомізованого правила. Коли вузол  $i$  вперше стає активним на кроці  $t$ , йому надається єдиний шанс активувати кожного поточно неактивного сусіда  $j$ ; він досягає успіху з імовірністю  $p_{ij}$  - параметр системи, який не залежить від історії попередніх змін стану. Коли агент  $i$  стає активним в певний момент часу [8], він отримує шанс активувати на наступному (і тільки на наступному) кроці кожного зі своїх сусідів  $j$  з імовірністю  $p_{ji}$  (причому  $j$  можуть намагатися незалежно активувати і інші агенти). Якщо активація пройшла успішно, то з наступного кроку  $t + 1$  даний вузол вважається активним. Незалежно від цього всі вузли, активні починаючи з кроку  $t$ , більше не мають можливості брати участь в процесі активації сусідніх вузлів. Процес триває до тих пір, поки зберігається хоча б одна можлива спроба активації.

У реальних соціальних мережах оцінити ймовірності  $p_{ij}$  досить складно, оскільки занадто багато факторів впливає на кожне конкретне рішення користувача про публікацію нового повідомлення або про поширення повідомлення когось із друзів. Більшість з цих факторів приховані від очей дослідника і відносяться до невидимої активності користувача в соціальній мережі. Дійсно, ми не знаємо, і навряд чи можемо визначити на основі відкритих даних, який обсяг часу користувач витрачає на перегляд сторінки, особистих повідомлень, коментарів та ін., тоді як будь-яке з цих джерел може привести, по-перше, до того, що користувач познайомиться з цікавим для нас повідомленням, і, по-друге, прийме рішення про його поширення.

У статті [9] пропонується узагальнення лінійної попрогової моделі і моделі незалежних каскадів і показується їх еквівалентність.

### 2.1.3 Моделі на основі клітинних автоматів

Для опису процесів поширення інформації в соціальній мережі останню можна розглядати як складну адаптивну систему, що складається з великої кількості агентів, взаємодія між якими призводить до масштабної, колективної поведінки, яку важко передбачити і аналізувати. Для моделювання та аналізу таких складних систем іноді використовуються клітинні автомати. Клітинний автомат (див., наприклад, [10]) складається з набору об'єктів (в даному випадку агентів), які зазвичай утворюють регулярну решітку.

Стан окремо взятого агента в кожен дискретний момент часу характеризується деякою змінною. Стани об'єкта синхронно змінюються через дискретні інтервали часу відповідно до незмінних локальних ймовірносних правил, які можуть залежати від стану змінних, що описують найближчих сусідніх агентів в околиці даного агента, а також можливо, від стану самого агента.

В роботі [8] моделюється ефект «з вуст в уста» в поширенні інформації в соціальних мережах. Кожен агент у великій мережі відноситься до однієї персональної мережі, агенти в якій пов'язані сильними (стабільними і постійними) зв'язками. Агент також має слабкі зв'язки з агентами з інших персональних мереж (про слабкі і сильні зв'язки див. [5]). Імовірність того, що проінформований агент вплине за допомогою сильного зв'язку на неінформованого агента (тобто останній стане проінформованим) в даний період часу дорівнює  $\beta_s$ , а по слабкій -  $\beta_w$  ( $\beta_s > \beta_w$ ). Також неінформовані агенти в даний момент часу з ймовірністю  $\alpha$  (яка менше імовірності, що досягається за допомогою ефекту «з вуст в уста») стають поінформованими завдяки рекламі та іншим маркетинговим прийомам.

Отже, в момент часу  $t$  неінформований агент, який має  $m$  сильних зв'язків з проінформованими агентами з його персональної мережі та  $j$  слабких зв'язків з проінформованими агентами з інших персональних мереж, стане поінформованим з ймовірністю:

$$p(t) = (1 - (1 - \alpha)(1 - \beta_w)^j(1 - \beta_s)^m) \quad (2.2)$$

В [8] пропонується використовувати імовірнісний клітинний автомат за наступним алгоритмом.

1. Спочатку всі агенти не проінформовані (значення 0).

2. У початковий момент часу агенти стають проінформованими завдяки рекламі, оскільки поширення інформації способом «з вуст в уста» вимагає наявності інформованих агентів. Для кожного агента датчиком випадкових чисел генерується випадкове число  $U$  ( $0 < U < 1$ ), яке порівнюється з ймовірністю  $p(t)$  реалізації інформованості. Якщо  $U < p(t)$ , то агент стане проінформованим (значення 1).

3. У наступні моменти часу підключається ефект «з вуст в уста» (сильні і слабкі зв'язки). Знову-таки, якщо  $U < p(t)$ , то агент стане поінформованим (значення 1).

4. Процес повторюється, поки 95% агентів не стануть поінформованими.

Для імітаційного експерименту в [8] задавалися наступні параметри: розмір кожної персональної мережі, число слабких зв'язків для кожного агента, ймовірність  $\beta_s$ , ймовірність  $\beta_w$  і  $\alpha$ . Як виявилось, хоча ймовірність поширення по слабким зв'язкам нижче, але вплив слабких зв'язків на швидкість поширення інформації таке ж, як і сильних зв'язків. У початковій фазі («early informed») більший вплив в інформуванні агентів має реклама (див. вище, в подальшому її роль незначна), в наступній фазі («middle informed») інформація поширюється в персональних мережах завдяки сильним зв'язкам; в міру того, як поінформованих агентів в таких мережах стає більше, ефект сильних зв'язків послаблюється, і зростає роль слабких зв'язків в активації нових мереж. При збільшенні розміру персональної мережі роль сильних зв'язків збільшується, а слабких - зменшується. При збільшенні кількості слабких зв'язків ефект від сильних зв'язків знижується, а

від слабких - збільшується. При посиленні реклами ефект від сильних зв'язків трохи збільшується, а від слабких - зменшується.

#### 2.1.4 Ланцюги Маркова

У статті [13] представлена модель ланцюгів Маркова, в якій вивчається вплив в команді (групі агентів). Запропонована модель є динамічною байєсовою мережею (Dynamic Bayesian Network - DBN) з дворівневою структурою: рівнем індивідів (моделюються дії кожного агента) і рівнем групи (моделюються дії групи в цілому). Всього є  $N$  агентів;  $i$ -й агент в момент часу  $t$  знаходиться в стані  $S_t^i$ , ймовірність якого  $P(S_t^i | S_{t-1}^i S_{t-1}^G)$  залежить від попереднього стану агента і стану команди, і виконує дію  $O_t^i$  з умовною ймовірністю  $P(O_t^i | S_t^i)$ . Команда в кожен момент часу  $t$  знаходиться в деякому стані  $S_t^G$ , ймовірність якого  $P(S_t^G | S_t^1, \dots, S_t^N)$  залежить від станів всіх агентів. Таким чином, для  $N$  агентів ймовірність того, що в певний момент часу  $T$  вони будуть знаходитися в сукупному стані  $S$  і зроблять сукупна дію  $O$ , дорівнює

$$P(S, O) = \prod_{i=1}^N P(S_1^i) \prod_{i=1}^N \prod_{t=1}^T P(O_t^i | S_t^i) \prod_{t=1}^T P(S_t^G | S_t^1, \dots, S_t^N) \prod_{t=2}^T \prod_{i=1}^N P(S_t^i | S_{t-1}^i S_{t-1}^G)$$

Якщо ввести змінну  $Q$ , яка визначає стан групи, і припустити, що:

а) вона не залежить від станів інших агентів;

б) при значенні  $Q = i$  стан групи  $S_t^G$  залежить тільки від стану  $i$ -го агента  $S_t^i$

то  $P(S_t^G | S_t^1, \dots, S_t^N)$  можна записати як

$$\sum_{i=1}^N P(Q = i) P(S_t^G | S_t^i) = \sum_{i=1}^N \alpha_i P(S_t^G | S_t^i)$$

де  $\alpha_i$  - вплив  $i$ -го агента на стан групи.

Описана дворівнева модель впливу тісно пов'язана з рядом інших моделей: *Mixed-memory Markov Model (MMM)* [14], *Coupled Hidden Markov Models (CHMM)* [14], моделі впливу і дерева динамічних систем (*DST - Dynamical Systems Trees*) [15]. *MMM* декомпозує складну модель (наприклад, марковську модель  $k$ -го порядку) в такий спосіб:

$$P(S_t^i | S_{t-1}, \dots, S_{t-k}) = \sum_{i=1}^N \alpha_{ij} P(S_t^i | S_{t-1}^i)$$

В моделі *CHMM* моделюється взаємодія декількох ланцюгів Маркова прямим зв'язком поточного стану одного потоку з попередніми станами всіх інших потоків:

$$P(S_t^i | S_{t-1}^1, S_{t-2}^2, \dots, S_{t-1}^N)$$

Однак така модель є обчислювально складною, тому її спрощують наступним чином:

$$P(S_t^i | S_{t-1}^1, S_{t-2}^2, \dots, S_{t-k}^T) = \sum_{i=1}^N \alpha_{ij} P(S_t^i | S_{t-1}^i)$$

де  $\alpha_{ij}$  - вплив  $j$ -го агента на  $i$ -го. Запропонована модель розширює ці моделі, використовуючи змінну  $S_t^G$  рівня групи, яка дозволяє моделювати вплив між усіма агентами і командою  $P(S_t^G | S_t^1, \dots, S_t^T) = \sum_{i=1}^N \alpha_{ij} P(S_t^G | S_t^i)$  і додатково встановлює динаміку кожного агента від стану команди  $P(S_t^i | S_{t-1}^i S_{t-1}^G)$

Дерева динамічних систем мають структуру дерева, яка моделює інтерактивні процеси через приховані ланцюга Маркова. Є дві відмінності між *DST* і розглянутої вище моделлю [13]. По-перше, в *DST* батько має власну ланцюжок Маркова, в той час як в даній моделі поточний стан команди прямо не залежить від її попереднього стану (т. Е. Дія групи - це агреговане дію агентів). По-друге, в моделі [13] як команда впливає на агентів, так і агенти впливають на команду.

Автори [13] висувають гіпотезу, що запропонований ними підхід до багаторівневого впливу послужить засобом аналізу соціальної динаміки для виявлення шаблонів виникає групової поведінки.

На властивості марковських ланцюгів також спираються *FHG* моделі [16] соціального впливу.

## 2.2 Індекси впливовості

У соціальній мережі іноді важливо визначити впливовість агента, кількісною характеристикою якої є індекс «сили» або індекс впливовості агента. Індекси «сили» (decisional power indexes) і індекси впливу. В роботі [17] вивчається наступна модель впливу в соціальній мережі. Перед агентами групи стоїть необхідність прийняти або відкинути деякий пропозицію (наприклад, проголосувати за певного кандидата на виборах). Передбачається, що агент спочатку має схильність до деякого рішенням (так - «+1» чи ні - «-1»), однак через впливу інших агентів може прийняти інше підсумкове рішення.

Іноді для оцінки впливу (decisional power)  $k$ -го агента на рішення всієї групи використовується індекс Хёде-Баккера (HoedeBakker) [18]:

$$HB_k(B, Gd) = \frac{1}{2^{n-1}} \sum_{\{i | i_k = +1\}} gd(Bi)$$

де  $N$  – множина з  $n$  агентів;  $i \in I$  - вектор схильностей/намірів цих агентів; а  $B: I \rightarrow I$  - функція впливу, яка відображає вектор намірів в підсумковий вектор рішень;  $gd: B(I) \rightarrow \{+1, -1\}$  - функція групового рішення. Індекс Хёде-Баккера означає міру відповідності рішень групи намірам агента. Функція впливу може бути заснована на принципі більшості (коли агент приймає рішення на основі наміру, підтримуваного більшістю агентів), або може бути заснована згідно ключової ролі експертів (агенти надходять відповідно до відомого наміру визнаного «гуру» /експерта) та ін.

В роботі [17] запропоновано наступне узагальнення індексу Хёде-Баккера:



$$GHB_k(B, Gd) = \frac{1}{2^n} \left( \sum_{\{i|i_k = +1\}} gd(Bi) - \sum_{\{i|i_k = -1\}} gd(Bi) \right)$$

Визначимо «Успіх» (*success*), «Невдачу» (*failure*) і «Авторитетність» (вирішальну роль, *decisiveness*) агента. Якщо рішення групи збігається з наміром агента, то він успішний. Якщо припустити, що вектори намірів рівновірогідні, то узагальнений індекс Хёде-Баккера є «чистий» успіх, тобто він дорівнює «Авторитетність = Успіх - Невдача», де Авторитетність, Успіх і Невдача є ймовірності того, що агент відповідно буде відігравати вирішальну роль, буде успішним або невдалим. Якщо успішний агент визначається як агент, наміру якого відповідає рішення групи, і вектори намірів різновірогідні, то узагальнений індекс Хёде-Баккера збігається з абсолютним індексом Банцафа [18]:

$$PB_k(W) = \sum_{R:k \in R \in W, R\{k\} \not\subseteq W} \frac{1}{2^n} + \sum_{R:k \notin R \not\subseteq W, R \cup \{k\} \in W} \frac{1}{2^n}$$

де  $R$  - такий вектор, в якому агенти з множини  $R$  приймають рішення прийняти пропозицію (значення відповідної компоненти вектора дорівнює «+1»), а інші відхиляють пропозицію (значення дорівнює «-1»).  $W$  - множина векторів, які ведуть до прийняття пропозиції (+1) групою відповідно до деякого заданого правила. Недолік індексу Хёде-Баккера полягає в тому, що він не вимірює вплив між агентами (неясно який ефект функції впливу  $B(x)$ ), а показує тільки відношення між вихідним вектором намірів і підсумковим рішенням групи [18]. Тому автори [18] вводять сімейство індексів впливу для агентів і їх коаліцій, що має узагальнену форму:

$$d_\alpha(B, S \rightarrow j) = \frac{\sum_{i \in I_{S \rightarrow j}^*(B)} \alpha_i^{S \rightarrow j}}{\sum_{i \in I_{S \rightarrow j}} \alpha_i^{S \rightarrow j}} \in [0, 1]$$

Величина  $d_\alpha$  показує зважений індекс впливу множини агентів  $S$  на  $j$ -го агента, який не входить в цю множину;  $I_{S \rightarrow j}$  - безліч всіх векторів намірів, для кожного з яких існує потенційна можливість впливу коаліції  $S$  на зміну

підсумкового рішення агента  $j$  (тобто це такий вектор  $I$ , для якого наміри коаліції і агента протилежні);  $I_{S \rightarrow j}^*(B)$  - безліч всіх векторів намірів; для кожного вектора  $i$  підсумкове рішення  $j$ -го агента  $(Bi)_k$  відповідає наміру коаліції  $S$ . Оскільки на зміну рішення агента  $j$  можуть вплинути й інші агенти з  $N \setminus \{S \cup j\}$  (як і їх різні коаліції) з тим же наміром, що і агенти з  $S$ , то вводяться вагові коефіцієнти  $\alpha_i^{S \rightarrow j}$  для  $S$ .

Автори [17] додатково вводять такі поняття:

- послідовники (followers) - агенти, рішення яких завжди збігаються з намірами агентів з множини  $S$  в тих випадках, коли наміри агентів  $S$  збігаються (вони утворюють коаліцію);
- функція чистого впливу  $B(x)$  множини агентів  $S$  на  $T$  - коли агенти з  $T$  приймають рішення виходячи з намірів агентів безлічі  $S$  в тих випадках, коли наміри агентів  $S$  збігаються, а решта агентів приймають рішення відповідно до своїх початкових намірів;
- ядро функції впливу  $B(x)$  - безліч таких множин агентів  $S$ , підмножини яких не мають послідовників. Перераховані конструкції дозволяють кількісно оцінити впливовість агентів і їх груп.

## Висновки до розділу 2

В цьому розділі були сформовані загальні характеристики і властивості соціальної мережі, яким повинна відповідати математична модель що описує соціальну мережу, взаємозв'язок між її учасниками, відображає вплив між ними. Перераховані ефекти і властивості повинні знаходити відображення в моделях, які претендують на адекватний опис реальних соціальних мереж.

Також були розглянуті деякі з найпопулярніших вже існуючих моделей соціальних мереж, особливо ретельно зосереджена увага на тому як в цих моделях виділяється вплив між агентами і понять які його супроводжують.

Розроблені властивості та досвід вже існуючих практик розробки моделей допоможуть в наступному розділі розробити власну модель соціальної мережі, яка здатна виділяти і досліджувати вплив між її агентами.

### 3 РОЗРОБКА МОДЕЛІ ВПЛИВІВ В СОЦІАЛЬНІЙ МЕРЕЖІ

#### 3.1 Розробка моделі

Для розробки моделі необхідно визначити і описати структури мережі, які б правдоподібно відображали властивості соціальної мережі загалом, поведінку її учасників, їх взаємини між собою, описати правила за якими учасники мережі комунікують між собою та впливають один на одного, визначити і формалізувати цілі, яка модель повинна задовільняти.

Основною структурною одиницею моделі є *агент* - вузол соціальної мережі (вершина графа). Агент може перебувати в активному і неактивному станах, причому можливий перехід тільки з неактивного стану в активне (зворотний перехід не допускається). Решту структури мережі утворюють такі введені поняття як: *співтовариство* (множина агентів, які не зазнають впливу агентів поза ним), *група* (спільнота агентів, в якому кожен агент впливає або підпадає під вплив кожного іншого агента групи прямо чи опосередковано) і *послідовник* (агент, що не виявляє впливу ні на одну з груп). Припустимо, що в кожній групі хоча б один агент хоч скільки-небудь довіряє своїй думці. Як виявляється, тоді в кінцевому підсумку думки супутників визначаються думкою груп, а всередині груп думки агентів сходяться і рівні (загальні необхідні і / або достатні умови збіжності - правильність ланцюга Маркова ін. - можна знайти в [12], огляд і дослідження ролі структури комунікацій агентів - в [1]). У такій соціальній мережі видається цілком природним розгляд завдання виявлення інформаційного впливу (зміна думок невеликої групи ключових агентів в мережі таким чином, що в результаті поширення зміни думок формуються необхідні думки решти учасників мережі).

Модель інформаційного впливу має давати можливість досліджувати залежність поведінки суб'єкта від його інформованості та від інформаційних впливів яким він піддавався. Маючи модель інформаційного впливу, можна

ставити і вирішувати завдання інформаційного управління - якими мають бути інформаційні впливи (з точки зору керуючого суб'єкта), щоб добитися необхідного поведінки від керованого суб'єкта. І, нарешті, вмючи вирішувати завдання інформаційного управління, можна моделювати інформаційне протиборство - взаємодія декількох суб'єктів, що володіють незбіжними інтересами і здійснюють інформаційні впливи на один і той же керований суб'єкт.

Соціальна мережа представляється графом  $G(N, E)$ , в якому  $N = \{1, 2, \dots, n\}$  – множина вершин, агентів (користувачів мережі), а  $E$  – множина ребер, що відображають взаємодію агентів.

Агенти в мережі впливають один на одного, і *ступінь впливу* задається матрицею прямого впливу  $T$  розмірності  $n \times n$ , де  $t_{ij}$  відображає ступінь довіри  $i$ -го агента до  $j$ -го агента (або *індекс впливовості*  $j$ -го агента на  $i$ -го).

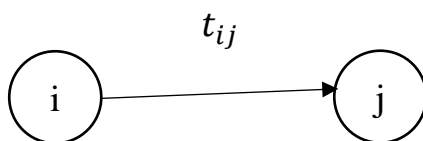


Рисунок 3.1 - Ступінь довіри  $i$ -го агента до  $j$ -го (вплив  $j$ -го на  $i$ -го)

*Ступінь довіри* (або іншими словами – *індекс впливовості*) пропонується розраховувати за наступною формулою:

$$t_{ij} = \frac{\alpha l_{ij} + \beta c_{ij} + \gamma r_{ij}}{N_j}, \quad (3.1)$$

де  $N_j$  – загальна кількість публікацій агента  $j$ ;

$l_{ij}$  – кількість реакцій агента  $i$  на публікації агента  $j$  (*likes, reactions*);

$c_{ij}$  – кількість коментарів агента  $i$  до публікацій агента  $j$ ;

$r_{ij}$  – кількість поширень агентом  $i$  публікацій агента  $j$ ;

$\alpha, \beta, \gamma$  – коефіцієнти значимості реакцій/коментарів/поширень відповідно.

$$t_{ij} \in [0; 1]$$

У кожного агента в початковий момент часу є своя індивідуальна думка з певного питання. Думку  $i$ -го агента з конкретного питання відображає число  $d_i$  таке, що  $d_i \in [-1; 1]$ . Тут числа -1 і 1 – граничні значення, що відповідно відповідають негативній і позитивній оцінці. Думку всіх агентів мережі тоді відображає вектор-стовбець думок  $d$  розмірності  $n$ . Агенти в соціальній мережі взаємодіють, обмінюючись думками. Цей обмін призводить до того, що думка кожного агента змінюється відповідно до думки агентів, яким даний агент довіряє:

$$d_i^{(k)} = \mu \sum_j t_{ij} d_j^{(k-1)}, \quad (3.2)$$

де індекс  $k$  означає момент часу,  $\mu \in \{-1, +1\}$  – тональна характеристика (негативне або позитивне).

У векторному записі перша зміна думки агентів дорівнює добутку матриці безпосередньої довіри на вектор початкових думок:  $d^{(1)} = td$ . Якщо обмін думками триває і далі, то вектор думок агентів стає рівним:  $d^{(2)} = t^2 d$ ,  $d^{(3)} = t^3 d$  і так далі. В кінцевому підсумку їх думки сходяться до результуючої (підсумкової) думки:  $D = \lim_{n \rightarrow \infty} d^{(n)}$ , тобто підсумкова думка дорівнює  $D = Td$ , де  $T = \lim_{n \rightarrow \infty} t^n$ . І тоді, по-перше, в кожній з груп мережі підсумкові думки агентів збігаються (цей результат відповідає висновкам соціальних психологів [24]); по-друге, підсумкові думки послідовників повністю визначаються думкою однієї або декількох груп [10]. Тоді рівняння  $D = Td$  відображає зв'язок початкової та результуючої думки агента щодо певного питання. Знаючи матрицю впливу, можна впливати на результуючу думку агентів шляхом додавання фактору керованого впливу. Формально керований вплив направлений на зміну думки агента можна представити у вигляді вектора керованих впливів  $u$ . Зміст цього вектора полягає в зміні початкової думки  $i$ -го агента з  $a_i$  на  $a_i + u_i$ . Тоді підсумкові думки  $B$  будуть визначатися таким рівнянням:

$$D_u = T(d + u), \quad (3.3)$$

тобто результуюча думка агента є сумою його власної результуючої думки і змін, викликаних керованим впливом.

Кожен агент мережі має свою індивідуальне ставлення до певного питання. Це ставлення можна представити у вигляді тональної характеристики: негативне, позитивне або нейтральне ставлення. С приводу конкретного питання усіх агентів мережі можна формально поділити так:

$$X \subseteq N, |x_i \in X, x_i \in [0.4, 1] \forall i = 1, \dots, n|, \quad (3.4)$$

$$Y \subseteq N, |y_j \in Y, y_j \in [-1, -0.4] \forall j = 1, \dots, m|, \quad (3.5)$$

$$Z \subseteq N, |z_k \in Z, z_k \in (-0.4, 0.4) \forall k = 1, \dots, z|, \quad (3.6)$$

$$||N|| = n + m + z, \quad (3.7)$$

Тут множина  $X$  – множина агентів, які позитивно ставляться до питання, множина  $Y$  – негативно і  $Z$  – нейтрально. З часом, під впливом інших агентів кожен агент може змінювати своє ставлення до розглядуваного питання, відповідно переходячи з однієї множини в іншу. Так, наприклад, агенти з «табору позитивних» можуть перейти до «табору негативних» під впливом інформаційного тиску з боку агентів, що дотримуються протилежної думки.

$$Y' = Y + F(X, \delta), \quad (3.8)$$

де  $\delta$  – керований вплив агентів з множини  $Y$  на агентів з множини  $X$ .

Такий умовний розподіл агентів по тональності їх думок дає корисний інструмент для відстеження зміни цих самих думок. Наприклад в ситуаціях, коли під впливом думок інших агентів піддослідний агент різко змінює свою власну думку з того чи іншого питання.

Наочно розглянемо ситуацію коли завдяки керованому інформаційному впливу певної групи агентів на конкретну жертву остання змінює свою думку на протилежну. На рисунку 3.2 схематично зображені дві групи агентів  $X$  та  $Y$ , кожна

з яких має загальну думку с приводу певного питання протилежну до конкуруючої групи. Група  $X$  має позитивну оцінку розглядуваного питання, група  $Y$  – негативну. У кожного агента наряду з числовим значенням думки (число всередині вузла зі знаком +/-) задано і ступені довіри до сусідніх агентів (стрілка від агента до агента із числовим значенням довіри).

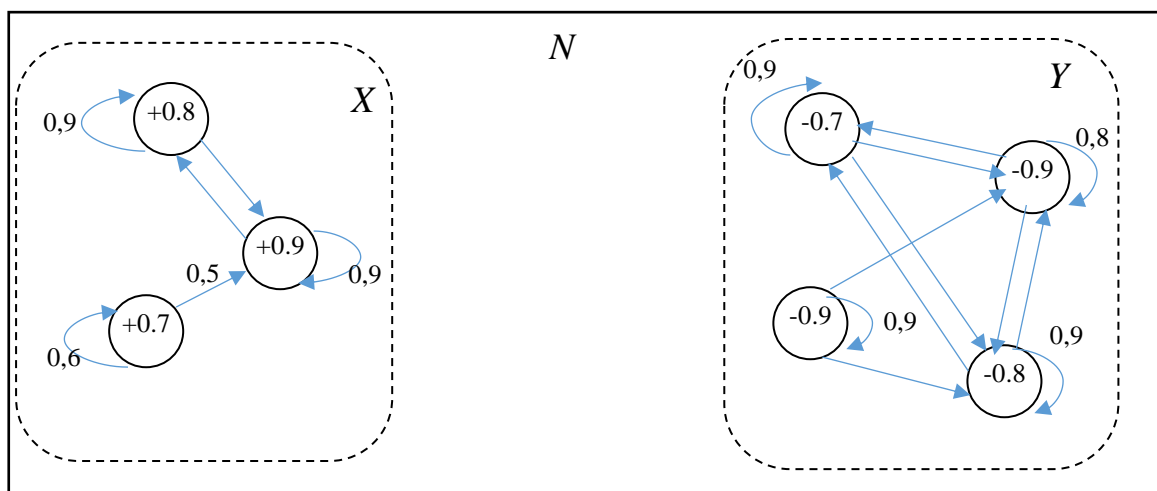
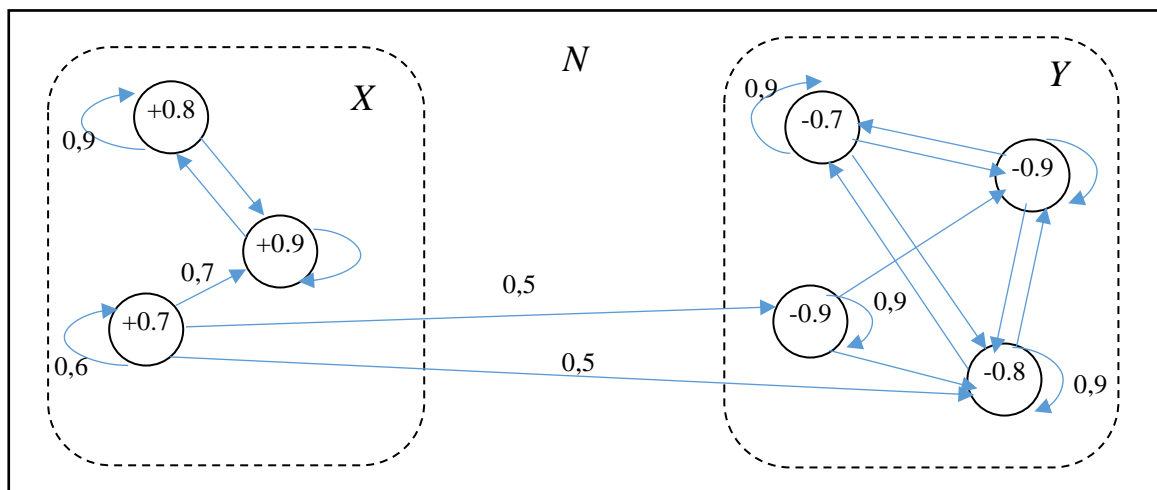


Рисунок 3.2 - Початковий розподіл думок в групах

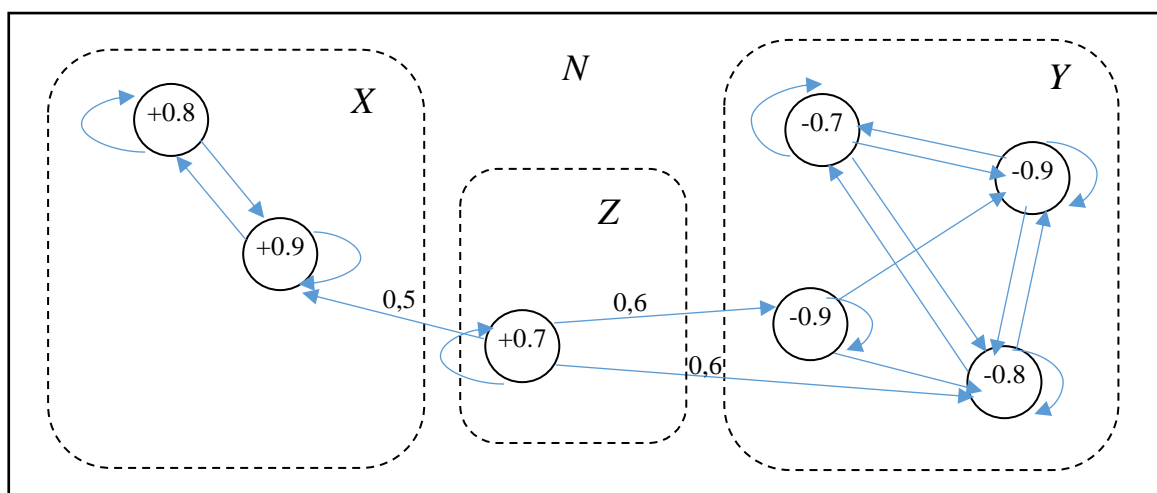
Бачимо, що у групі  $Y$  спостерігаються тісні зв'язки між агентами та високий рівень довіри один до одного. Також кожен агент має досить високий рівень «самодовіри», тобто агент глибоко впевнений в своїх переконаннях, та вважає себе самодостатнім носієм ідеї. Водночас, у групі  $X$  явно виділяється агент, який має досить слабкі зв'язки зі своїми однодумцями та найнижчий показник довіри до самого себе, тобто, очевидно, що агент знаходиться у стані вагання.

На наступному етапі (рисунок 3.3) на цього «найслабшого» агента групи  $X$  починають впливати агенти з групи  $Y$ .



Рисунок 3.3 - Поява керованого впливу  $Y$  на  $X$ 

На рисунку 3.4 бачимо, що розподіл значень довіри розглядуваного агента змінився: окрім вектора довіри до свого сусіда з групи  $X$  з'явилися ще два вектори до агентів з групи  $Y$ , причому сумарно числове значення довіри до агентів з групи  $Y$  почало переважати зв'язок з групою  $X$ . Як результат маємо перехід агента до «нейтральної» групи  $Z$ . Ступінь довіри до сусіда з групи  $X$  поступово зменшується, а довіра до агентів з групи  $Y$  навпаки зростає.

Рисунок 3.4 - Перехід агента до групи  $Z$ 

На кінцевому етапі, зображеному на рисунку 3.5 зв'язок агента з групою  $X$  остаточно зникає, зв'язки з агентами з групи  $Y$  остаточно закріплюються. Виявляється, що перехід агента з однієї ідеологічної групи до іншої є поступовим

процесом, який вимагає часових затрат, наявності проміжних станів та постійного інформаційного впливу з боку конкуруючої групи.

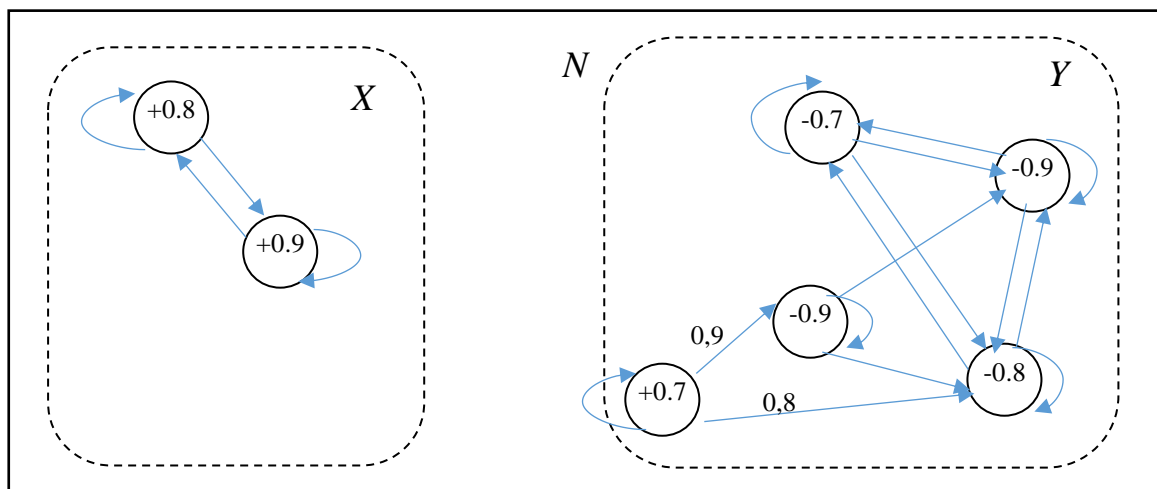


Рисунок 3.5 – Остаточний перехід агента до групи  $Y$

І, нарешті, в рамках запропонованої моделі можна ставити і вирішувати завдання визначення оптимальних варіантів захисту від інформаційних впливів на агентів, які утворюють соціальну мережу.

### 3.2 Програмна реалізація

Для перевірки та застосування у реальному світі будь-якої з описаних моделей необхідно мати в своєму розпорядженні дані про реальних агентів в існуючих соціальних системах (соціальних мережах типу Facebook, Instagram, V Kontakte та ін.). Для досягнення цих цілей необхідно виконати комплекс заходів результатом яких буде система/програмне забезпечення для збору і аналізу даних в обраній соціальній мережі. До цих заходів можна віднести: планування архітектури програмного забезпечення; написання коду для сканування даних; вибір підходящої бази даних для збереження добутих даних; інструменти для аналізу та візуалізації отриманих даних. Метою і результатом даного підрозділу є саме реалізація цих заходів та отримання програмного забезпечення для аналізу поведінки агентів та виявлення впливів між агентами у обраній соціальній системі.

Для розробки програмного забезпечення була обрана інтерпретована об'єктно-орієнтована мова програмування високого рівня Python. Python характеризується високою швидкістю написання коду, великою кількістю готових бібліотек в сфері http-трафіку та sentiment-аналізу тексту, що робить її максимально оптимальною для дослідження поведінки та впливів агентів в соціальних системах.

### 3.3 Збір даних та схема бази

Були зібрані дані за останній місяць з товариства «*Remain in the European Union - Exit from Brexit!*» соціальної мережі Facebook (<https://www.facebook.com/groups/RemainInTheEuropeanUnion/>). Усіх цього учасників товариства об'єднує спільна думка щодо конкретного питання: виходу Великобританії з Європейського союзу.

В результаті вийшла база даних на:

- 7141 постів;
- 222483 «людей, що відреагували»;
- 62766 коментарів;
- 31638 поширень.

Для зберігання даних була обрана документо-орієнтована система керування базами даних (СУБД) – MongoDB.

### 3.3.1 Колекція «posts»

▼ (2) ObjectId("5db5a1ffc59e2f307e4d8dfc")	{ 14 fields }	Object
_id	ObjectId("5db5a1ffc59e2f307e4d8dfc")	ObjectId
id	UzpfSTewMDAwNzgyOTUyODI0NjpWSzoyNDg4MDg2Mzk4MTMyNzA3	String
post_id	2488086398132707	String
url	https://www.facebook.com/groups/RemainInTheEuropeanUnion/permalink/248...	String
text	Why does Jo Swinson want a GE on the 9th December?	String
author	Carole Coote	String
author_id	100007829528246	String
source	Remain in the European Union - Exit from Brexit!	String
source_id	1693042670970421	String
likes	5	Int32
comments_count	96	Int32
reshares_count	0	Int32
created_at	2019-10-26 22:44:17.000Z	Date
scanned_at	2019-10-27 13:32:35.000Z	Date

Рисунок 3.6 - Колекція «posts»

**id** – внутрішній мережевий ідентифікатор, використовується в API мережі;

**post\_id** – «людський» ідентифікатор, котрий можна спостерігати в посиланнях:

<https://www.facebook.com/2488020931472587>;

**author\_id** – ідентифікатор автора поста;

**source\_id** – ідентифікатор джерела поста, в даному випадку товариства «*Remain in the European Union - Exit from Brexit!*».

Кількість коментарів і поширень зазначені тут можуть не відповідати реальній кількості в базі. У випадку поширення зрозуміло - користувач може приховати свій профіль і його поширення будуть приховані відповідно до політики конфіденційності мережі. З коментарями цікавіше – система фільтрує непридатні спам-коментарі і часто буває, що заявлена кількість коментарів не відповідає кількості коментарів, які можна переглянути.

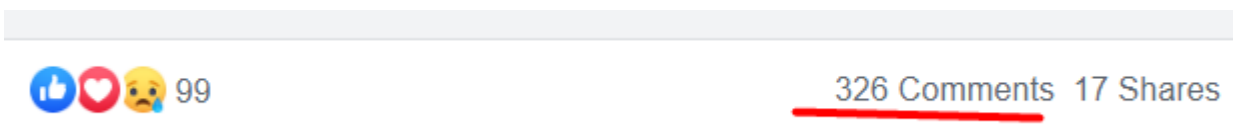


Рисунок 3.7 - Статистика в інтерфейсі

### 3.3.2 Колекція «reactors»

(1) ObjectId("5db5a247c59e2f307e4db354") <ul style="list-style-type: none"> <li>_id</li> <li>post_id</li> <li>user_id</li> <li>user_name</li> <li>feedback_reaction_info             <ul style="list-style-type: none"> <li>id</li> <li>key</li> </ul> </li> </ul>	{ 5 fields } ObjectId("5db5a247c59e2f307e4db354") 2488487781425902 1638113085 Cate Hamilton { 2 fields } 444813342392137 8	Object ObjectId String String String Object String Int32
--	---	---

Рисунок 3.8 - Колекція «reactors»

Зв'язок з постом по *post\_id*.

Тут з цікавого - поле *feedback\_reaction\_info*. Тут зберігається «тип лайку»/реакції, яку поставив користувач під постом.

Наприклад: 😄 👍 😞 ❤️

За цими реакціями в перспективі можна визначати тональність (позитивна, негативна, нейтральна). Поки в підрахунках це ніяк не використовується.

### 3.3.2 Колекція «comments»

(3) ObjectId("5db5a25fc59e2f307e4db3e2") <ul style="list-style-type: none"> <li>_id</li> <li>id</li> <li>post_id</li> <li>comment_id</li> <li>url</li> <li>text</li> <li>author</li> <li>author_id</li> <li>source_id</li> <li>likes</li> <li>comments_count</li> <li>created_at</li> <li>scanned_at</li> <li>parent</li> </ul>	{ 14 fields } ObjectId("5db5a25fc59e2f307e4db3e2") Y29tbWVudDoyNDg4NDg1NTIxNDI2MTIxXzl0ODg2ODYzNDE0MDYwNDY= 2488485591426121 2488686341406046 https://www.facebook.com/2488686341406046/ Farhad Nadjm Brexit Party members have an IQ of 100 - a total, not an average. Robert Powell 100001472721288 1693042670970421 0 0 2019-10-27T15:30:05+00:00 2019-10-27T13:57:50+00:00 Y29tbWVudDoyNDg4NDg1NTIxNDI2MTIxXzl0ODg2NTQ1MDE0MDkyMzA=	Object ObjectId String String String String String String String String Int32 Int32 String String String
---	---	--

Рисунок 3.9 - Колекція «comments»

Зв'язок з постом по *post\_id*.

Поле *parent* – опціональне, присутнє якщо даний коментар є відповіддю на інший коментар. Максимальна глибина вкладеності коментарів в facebook - 2.

### 3.3.3 Колекція «reshares»

▼ (2) ObjectId("5db5a243c59e2f307e4db33d")	{ 9 fields }	Object
_id	ObjectId("5db5a243c59e2f307e4db33d")	ObjectId
top_level_post_id	10217950008976295	String
content_owner_id_new	1666906359	Int32
original_content_id	2459459087654018	String
original_content_owner_id	100007698518650	String
url	https://www.facebook.com/johndecs/posts/10217950008976295	String
text	"They said I couldn't negotiate a deal with Brussels, but I have"! He did. He skilfu...	String
name	John Wynne	String
post_id	2488501531424527	String

Рисунок 3.10 - Колекція «reshares»

Зв'язок з постом по *post\_id*.

*content\_owner\_id\_new* – ідентифікатор користувача, який зробив поширення;

*top\_level\_post\_id* – ідентифікатор поширення, на сторінці у користувача, котрий зробив поширення;

*original\_content\_owner\_id* – ідентифікатор автора оригінального посту;

*text* – підпис користувача, котрий здійснив поширення.

## 3.4 Аналіз і візуалізація даних

За весь час було обрано топ-100 користувачів по активності (лайки, коментарі, поширення). За кожен день була підрахована активність кожного користувача в цій товаристві. За формулою: *daily\_activity* += 0.2 за реакцію, += 0.3 за коментар, += 0.5 за поширення. Такі коефіцієнти були вибрані емпіричним методом, опираючись на досвід отриманий с декількох спроб моделювання цієї характеристики. Використання саме таких коефіцієнтів дає найбільш наглядну картину для аналізу.

Тональність повідомлення ніяк не враховувалася (тобто негативних значення немає). У перспективі можна визначати тональність повідомлення за допомогою

вже існуючих sentiment-аналайзерів. Наприклад бібліотека TextBlob [5]. І враховувати тип реакції по правилу (👎 :  $\cdot (-1)$ , 👍 :  $\cdot (+1)$ ).

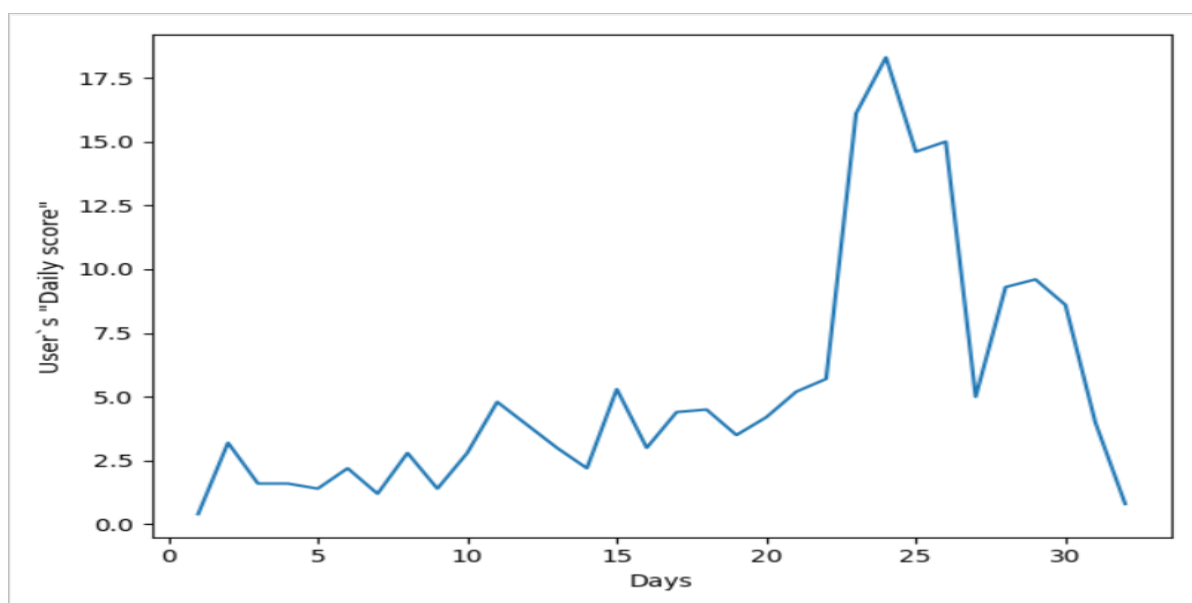


Рисунок 3.11 - Графік активності випадкового користувача №1

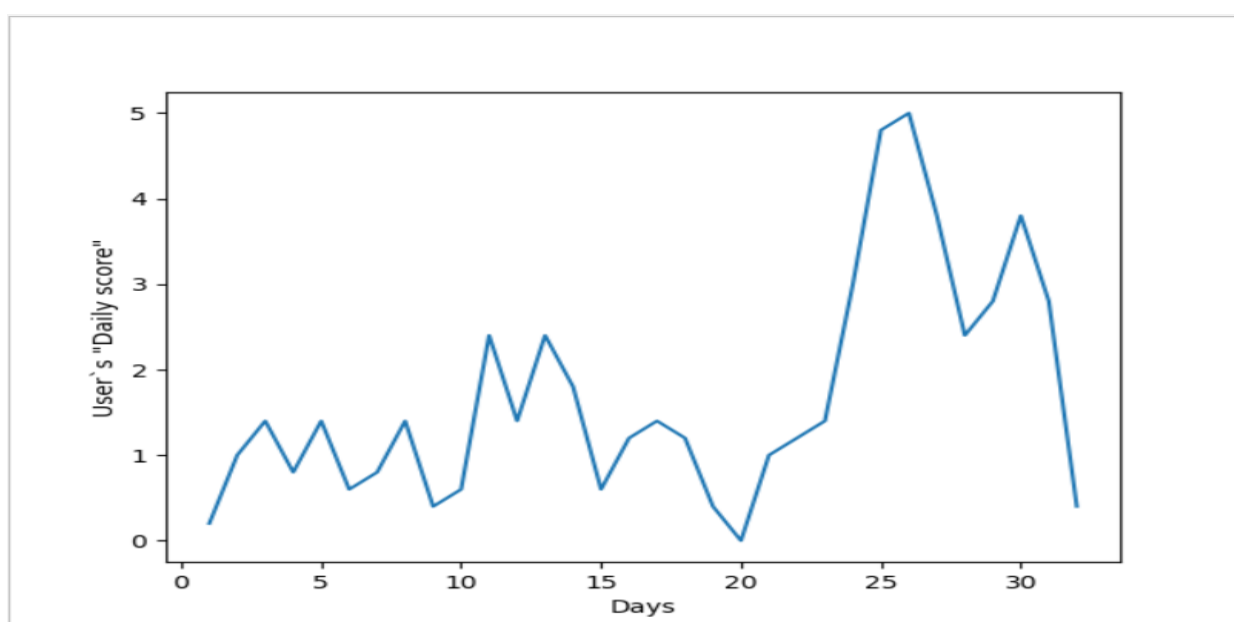


Рисунок 3.12 - Графік активності випадкового користувача №2

Якщо накласти один на одного графіки активності випадкової вибірки учасників, можна спостерігати загальну картину активності аудиторії товариства.

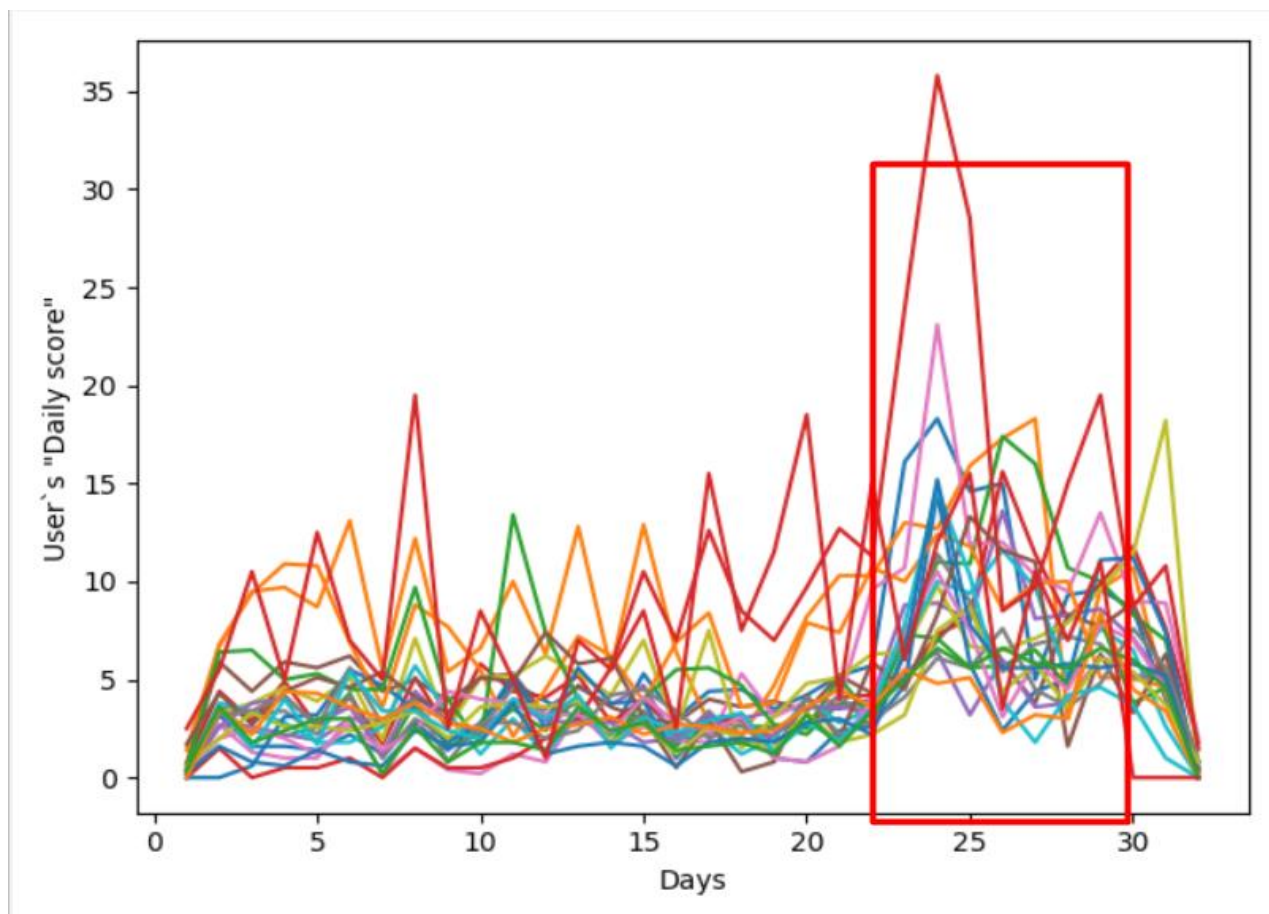


Рисунок 3.13 - Графік активності 50 випадкових користувачів

На рисунку 3.13 можна спостерігати сплеск активності в групі в період впродовж з 22 по 30 день (виділено прямокутником). Також по щільності ліній на графіку можна виділити 2 групи користувачів.

Графік активності першої групи агентів характеризується плавними переходами і великою щільністю серед ліній сусідів. Такі агенти проявляють свою активність досить рівномірно серед більшості публікацій товариства, регулярно відвідують його і рідко вступають у дискусії в коментарях.

Графіки активності другої групи людей навпаки характеризуються «піками» і розривами від сусідніх графіків. Такі агенти вибірково реагують на окремі публікації, підтримують дискусії у коментарях, часто роблять поширення публікацій і не рідко досить експресивно доводять свою точку зору в коментарях. Така поведінка як не можна краще підходить під опис спрямованого впливу на інших учасників товариства.



### 3.5 Експеримент по відслідковуванню змін думок агентів

Експеримент полягає у перевірці гіпотези щодо можливості відслідковувати зміну думки агента по певному питанню з плином часу. Пропонується взяти два ідеологічно протилежні співтовариства, які налічують велику кількість учасників, в яких учасники мають можливість дискутувати на різні гострі питання та в яких щодня публікується багато інформації. Таким чином відпадає необхідність у смислового аналізу тексту повідомлення для числового визначення думки агента щодо розглядуваного питання. Передбачається, що в одному співтоваристві вся інформація, що публікується сприймається як позиція «за», а в іншому співтоваристві – як позиція «проти». Аналогічно і з числовим виміром думки окремого агента: чим більша його активність в одному із співтовариств, тим більша його прихильність до загальної думки, яка присутня в співтоваристві.

Для проведення експерименту була зібрана інформація з публічних співтовариств «*Remain in the European Union - Exit from Brexit!*» (противники виходу Великобританії з Європейського союзу) та «*BREXIT*» (прихильники виходу Великобританії з Європейського союзу). Для дослідження були обрані учасники, які проявляли активність в обох співтовариствах, тобто перетин множин агентів:

$$X = A \cap B$$

де  $A$  – множина агентів співтовариства «за»;

$B$  – множина агентів співтовариства «проти»;

Були зібрані публікації, коментарі, реакції, поширення в кожному з співтовариств в період з серпня по жовтень поточного року. Важливо зазначити, що інформація збиралася не в «режимі реального часу» а в ретроспективі. Цей факт може нести негативний ефект на хід експерименту, адже ми втрачаємо статистику про користувачів, які в цей період покинули співтовариство.

Для вимірювання думки агента були застосовані наступні правила:

$$X \subseteq N, |x_i \in X, x_i \in [0.4, 1] \forall i = 1, \dots, n|$$

$$Y \subseteq N, |y_j \in Y, y_j \in [-1, -0.4] \forall j = 1, \dots, m|$$

$$Z \subseteq N, |z_k \in Z, z_k \in (-0.4, 0.4) \forall k = 1, \dots, z|$$

де  $X$  – множина учасників співтовариства «*BREXIT*», тобто агенти, які дотримуються думки «за»;

$Y$  – множина учасників співтовариства «*Remain in the European Union - Exit from Brexit!*», тобто агенти, які дотримуються думки «проти»;

На рисунку 3.14 показаний початковий розподіл думок агентів.

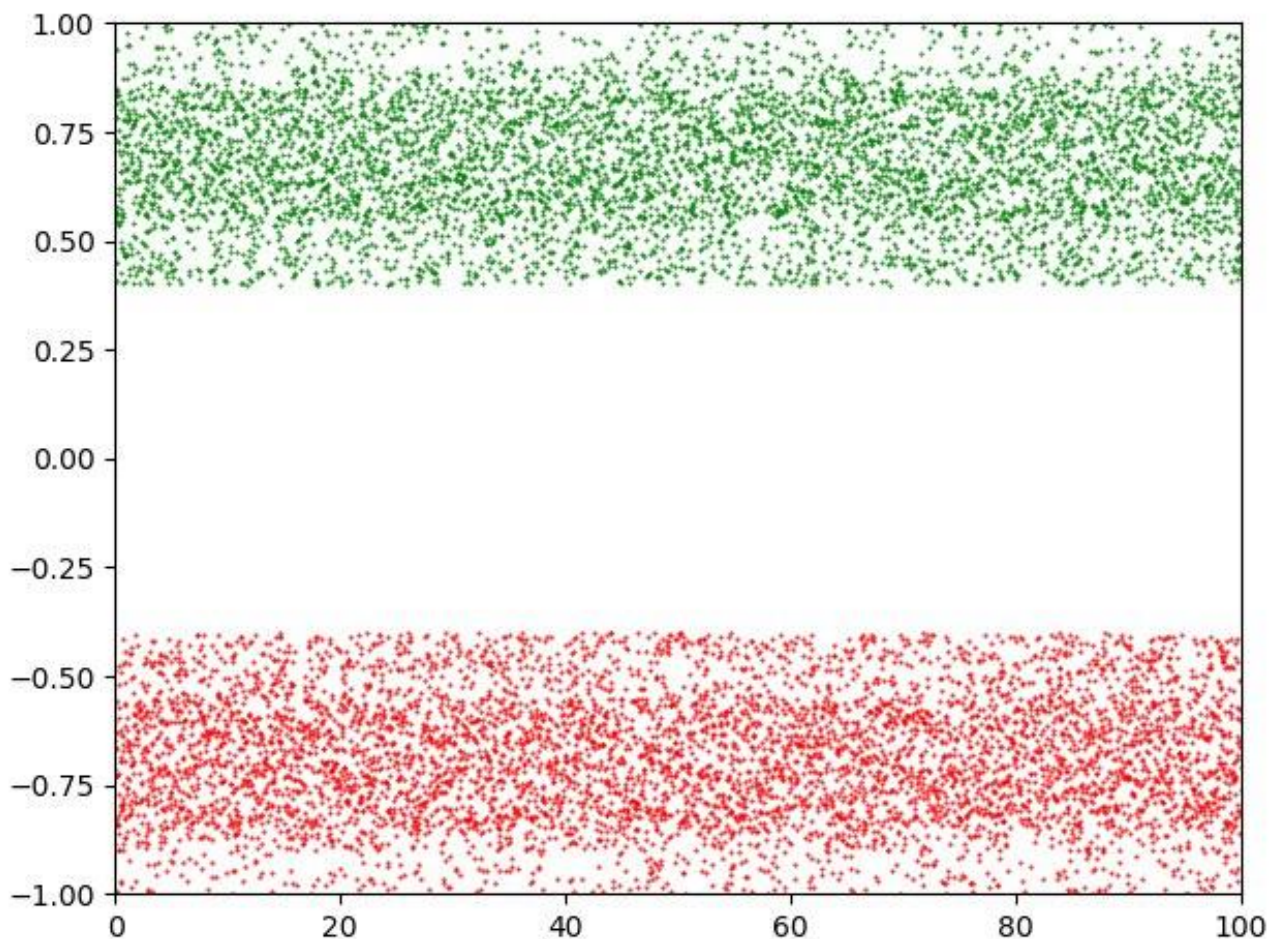


Рисунок 3.14 - Розподіл думок користувачів.

Тут зеленим виділені користувачі, які дотримуються думки «за»  $d_i \in [+0.4, +1]$ , червоним – користувачі, які дотримуються думки «проти»  $d_i \in [-1, -0.4]$ . Для більшої зручності і інформативності були заздалегідь відфільтровані «нейтральні» користувачі, тобто такі, у яких числова міра думки знаходиться в межах  $d_i \in (-0.4, 0.4)$ . Вісь абсцис – часовий проміжок тривалістю в один місяць для зручності і наочності розбитий на 100 однакових частин (в реальності кожній точці на осі абсцис відповідає *timestamp*, тобто момент часу за який ведеться спостереження). Як видно на графіку основна маса користувачів кожного співтовариства зосереджена в проміжку  $[0.55, 0.85]$  для користувачів групи «За» та  $[-0.85, -0.55]$  групи «Проти». Це основна маса та найбільш насичена маса користувачів, їх можна охарактеризувати як «агенти що визначилися» або основного носія загальноприйнятої думки. Проміжок  $[0.85, 1]$  є менш тісно заповненим, його населяють «лідери думок» - агенти, що задають тенденцію співтовариства, створюють контент, який супроводжується реакціями і дискусіями інших користувачів. Таких агентів можна охарактеризувати як ідеологічно сильних, тобто мало вразливих до зміни своєї думки під стороннім впливом.

На рисунку 3.15 зображений підсумок наступного місяця. Множина користувачів, зображених на графіку зафіксована з попереднього етапу і не включає в себе нових користувачів задля збереження інформативності. Тут видно як в основному агенти з нижчих слоїв (проміжки  $[0.4, 0.6]$  для групи «за» та  $[-0.6, -0.4]$  групи «проти») починають поступово змінювати свою думку в підтримку протилежної групи. Примітно, що процес стосується користувачів обох співтовариств, та майже не зачіпає складів груп «ідейних лідерів». Разом перебіжчики утворюють тимчасову «нейтральну зону». Очевидно, вплив завданий на агентів в нейтральній зоні поки що не є достатнім для остаточної зміни думки на протилежно, але достатній, щоб більше не асоціювати розглядуваних агентів з одним із співтовариств.

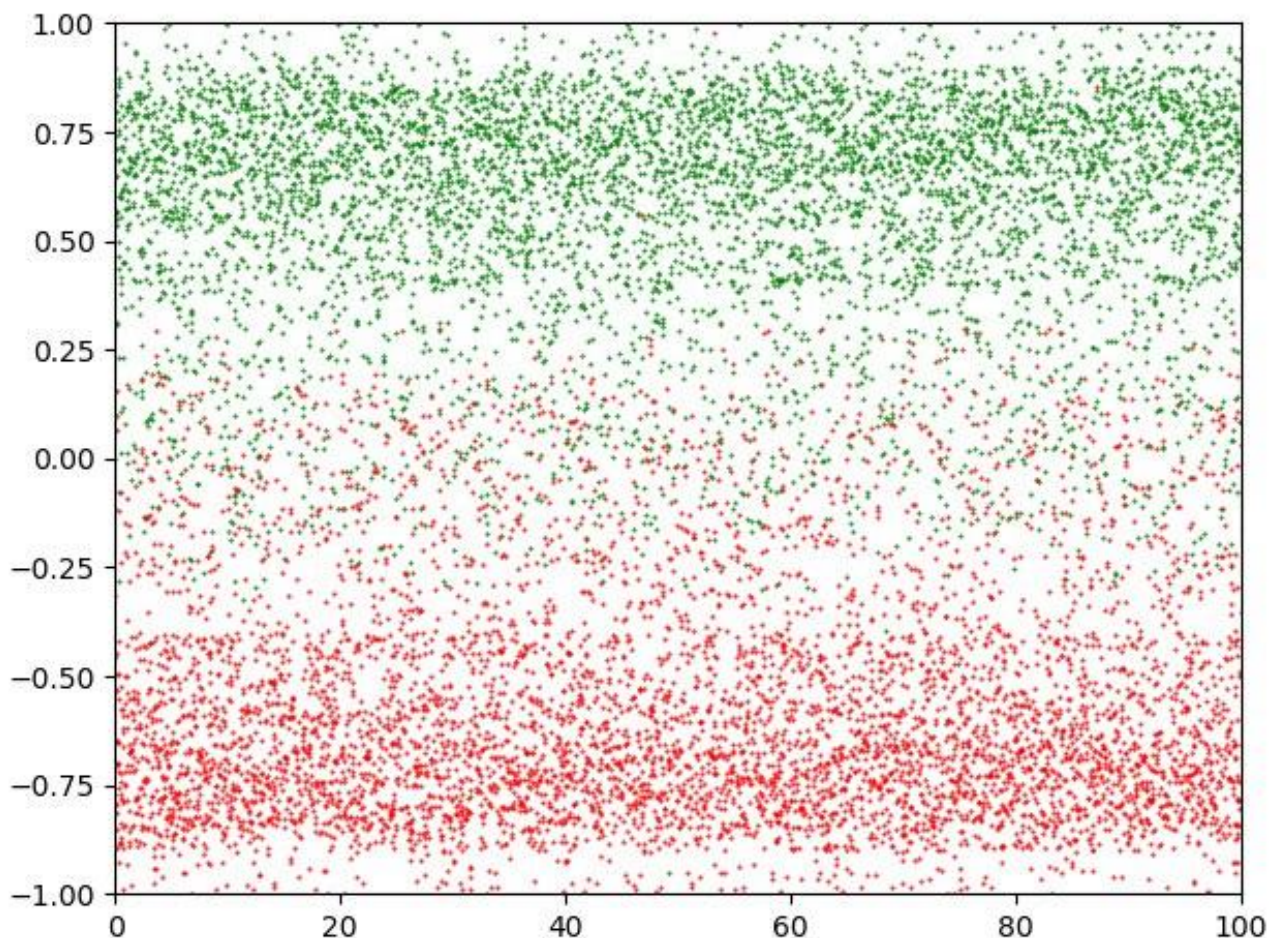


Рисунок 3.15 - Процес «переходу думок».

На кінцевому етапі експерименту зображеному на рисунку 3.16 спостерігаємо ситуацію остаточної зміни агентами переконань щодо розглядуваного питання. Видно, що прогалини в рядах кожного співтовариства утворені на попередньому етапі заповнилися новими агентами. «Нейтральна зона» тепер заповнена рівномірно по всій площині. Проміжки  $[0.55, 0.85]$  та  $[-0.85, -0.55]$  так і залишилися найбільш густо заселеними, а прогалини які наявні при порівнянні першого та останнього графіків насправді заповнені новими користувачами, або користувачами, які перебували в «нейтральній зоні» на початку експерименту. Переважна більшість «лідерів думок» залишилися при своїй думці в зонах  $[0.85, 1]$  і  $[-1; -0.85]$ .



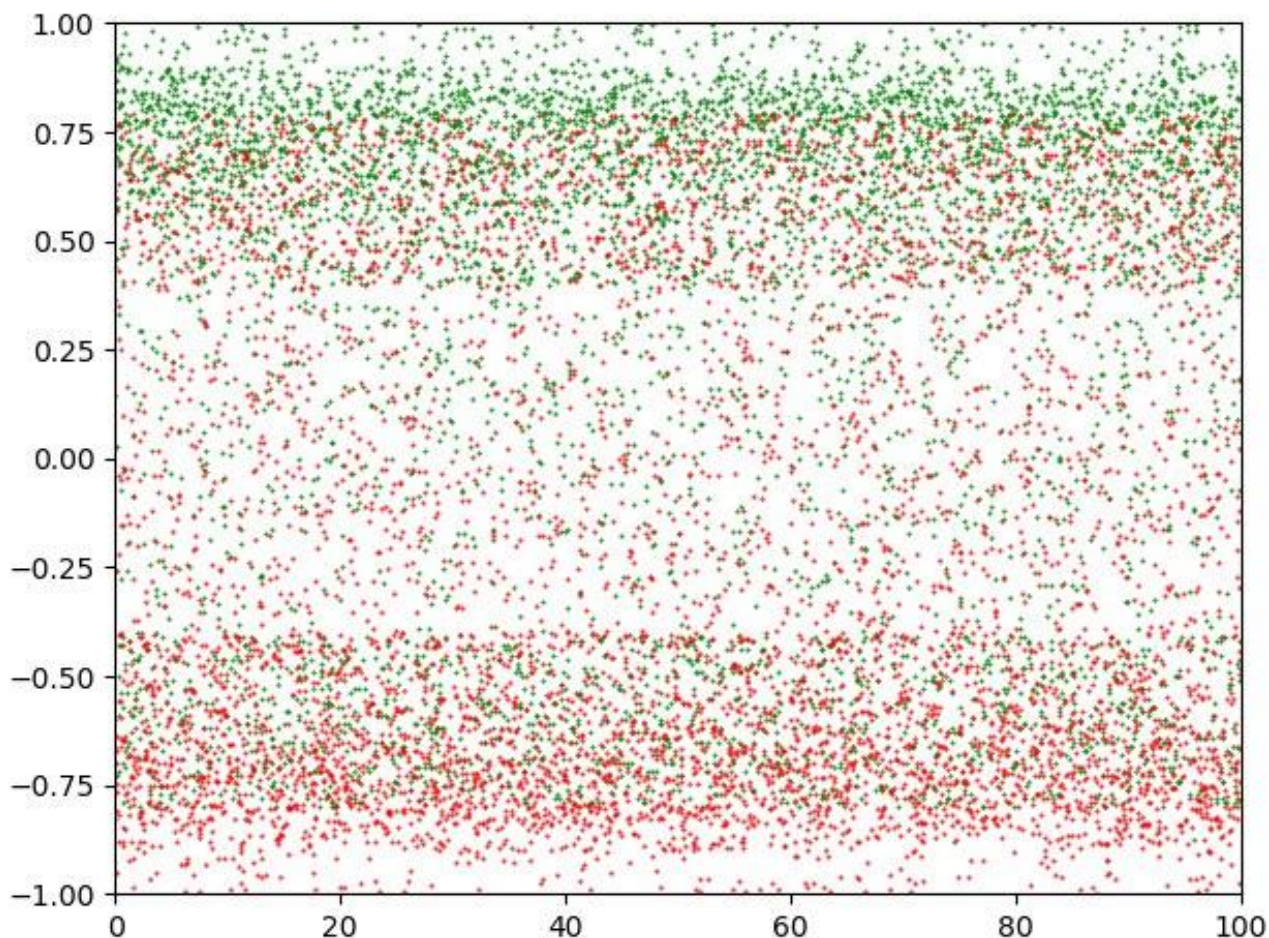


Рисунок 3.16 - Фінальний етап «переходу думок».

Зважаючи на зовнішню і змістовну подібність першого та останнього графіків можна зробити висновок, що процес «переходу думок» є процесом циклічним і невід’ємним у життєдіяльності соціальної мережі.

У відсотковому відношенні результати спостереження можна описати так: приблизно 29,8% прихильників співтовариства «За» в кінці спостереження перейшли в склад співтовариства «Проти»; лише 22,3% учасників товариства «Проти» перейшли до складу співтовариства «За». Очевидно, що існує тенденція відтоку користувачів з групи «За» у групу «Проти», а зважаючи на те, що цей процес є циклічним — ця тенденція може в подальшому посилитися.

### 3.6 Виявлення агентів з найбільшим впливом

Користуючись результатами, які ми отримали в попередньому експерименті спробуємо виявити агентів, які за допомогою свого впливу на агентів з протилежного співтовариства спровокували перехід цих самих агентів до свого співтовариства.

Для початку необхідно виділити вибірку агентів, які в результаті впливу від інших агентів змінили свою думку, назовемо цю вибірку *групою перебіжчиків*  $P$ . Це будуть ті самі агенти які в початку експерименту перебували в «зоні зелених», а в кінці експерименту опинилися в «зоні червоних»:

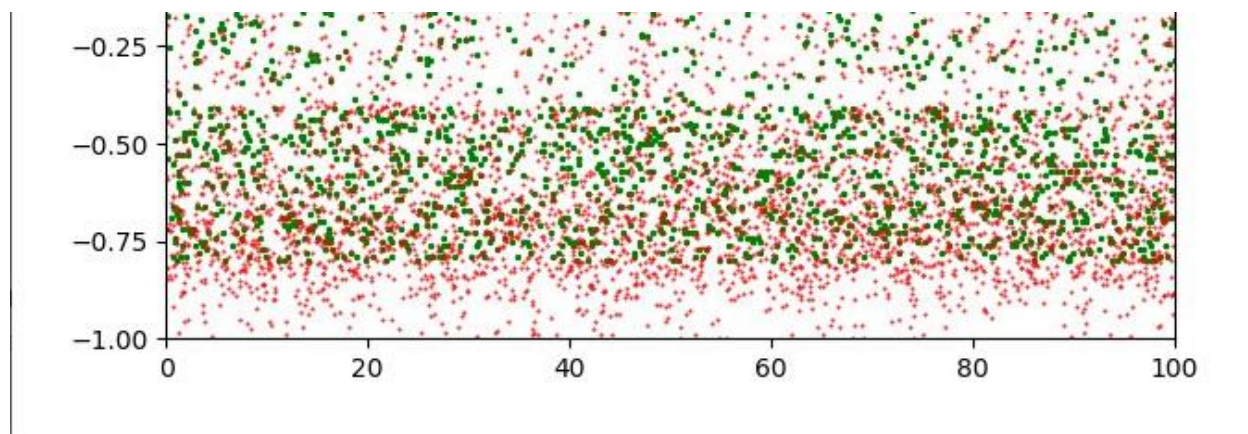


Рисунок 3.16 – Виділення агентів, що перейшли у групу опонентів.

Наступним кроком буде пошук зв'язків між учасниками *групи перебіжчиків* та «лідерами думок» співтовариства «Проти» на кожному з часових етапів розглянутих у експерименті. Під зв'язками мається на увазі пошук реакцій, коментарів та поширень та визначення індексу впливовості  $t_{ij}$  за формулою (3.1). В результаті отримаємо матрицю впливу  $T$ , проаналізувавши яку можна буде виділити агентів з найбільшими індексами впливу.

На рисунку 3.17 зображені «вектори впливу»  $T_j$ ,  $u_j \in X$ ,  $|T_j| = |P|$  найбільш впливовіших агентів множини  $X$ . Тут вектор впливом – це набір індексів впливу  $j$ -

го агента на агентів с множини  $P$ . Для більшої наочності вміст вектора, тобто індекси впливовості, відсортовані по спаданню.

$X_1$	$X_2$	$X_3$	$X_4$
[0.9250831952074099,	[0.9452447336047394,	[0.9397721282057596,	[0.8692568036904638,
0.9056676004771075,	0.9317924830740745,	0.9287836359786189,	0.8575754803587928,
0.904268927182066,	0.9276365819318395,	0.9267355461700046,	0.853905504970608,
0.8740679187983762,	0.9088073968301905,	0.9092162415049553,	0.8502223458862375,
0.8135014026728045,	0.8468852324093784,	0.8977219427605283,	0.8113026561295146,
0.7586364004631108,	0.7979319709083006,	0.8975409028919524,	0.8102926770615684,
0.7406326530325632,	0.7380096408234147,	0.8962263759135407,	0.8028194219965203,
0.7405493355454132,	0.7335106570402617,	0.8904449478251654,	0.7501814973379618,
0.7084532309563494,	0.7218112433195203,	0.8878089105974565,	0.6939487789810823,
0.7007147349758218	0.7086079406499239	0.8757027138902534	0.6504802380575067
...	...	...	...
]	]	]	]
...			
$X_{110}$	$X_{111}$	$X_{112}$	$X_{113}$
[0.24974700354027835,	[0.197397344231558,	[0.1830142310381989,	[0.1657867983815960,
0.23014849311038987,	0.1962013968022707,	0.18267821237892318,	0.16303500673032612,
0.22983318261871843,	0.1939236748278433,	0.18151592919911053,	0.16024922805765993,
0.20365267034027978,	0.1911989160896352,	0.18090525227993848,	0.1578801189051335,
0.19802903652783554,	0.1732092075080525,	0.17979272927172554,	0.15517096499584335,
0.19412861091939482,	0.1719207544476279,	0.17241857687976242,	0.1499693712221539,
0.18371528432422563,	0.1678042538101275,	0.16407949091073437,	0.14919701284547104,
0.17040474768714928,	0.1655013239994944,	0.16383079735540643,	0.14169147190352882,
0.168398602046893,	0.1519868290163349,	0.1637700434219647,	0.13878689891588594,
0.16123521537826044	0.1508281831965307	0.15346911087812423	0.13433052249529384
...	...	...	...
]	]	]	]

Рисунок 3.17 – Вектори впливу агентів з  $X$

Всього було знайдено 113 агентів впливу. Проаналізувавши вміст векторів впливу для кожного агента можна виділити найбільш активних агентів. Так, на рисунку 3.17 бачимо, що початок списку очолюють агенти з досить високими показниками індексів впливу – це і є наші потенційні маніпулятори.

### **Висновки до розділу 3**

В цьому розділі була розроблена і реалізована модель, яка дозволяє відстежувати впливи між користувачами в соціальній мережі. Для цього була розроблена структура моделі, сформульовані правила, за якими користувачі соціальної мережі взаємодіють між собою. Також були проведені експерименти по визначенню найвпливовіших агентів конкретних співтовариств та моніторингу динаміки зміни загальної думки агентів в цілому.

До списку задач, які можна вирішити за допомоги запропонованої моделі можна віднести:

- 1) моніторинг активності звичайних користувачів;
- 2) виділення найбільш активних користувачів;
- 3) відслідковування змін думок агентів;
- 4) виявлення впливів між агентами;
- 5) виявлення найбільш впливових агентів.



## 4 РОЗРОБКА СТАРТАП-ПРОЕКТУ

### 4.3 Опис ідеї стартап-проекту

Ідеї та напрацьована інформація, що викладені в даній магістерській дисертації мають практичну та комерційну цінність для областей пов'язаних з моделюванням соціальних мереж. Цільове призначення розробленої моделі — розробка спеціалізованого програмного комплексу для моніторингу, виявлення та керування впливами в існуючих соціальних мережах.

Опис ідеї проекту наведено в таблиці 4.1.

Таблиця 4.1 - Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Стартап проект передбачає створення зручного застосунку для моніторингу, виявлення та керування впливами в існуючих соціальних мережах	1. Маркетинг	Можливість ефективно спостерігати, супроводжувати або керувати маркетинговою кампанією пов'язаною з поширенням необхідної інформації в соціальній мережі

Кінець таблиці 4.1

	2. Адміністрування соц-мережі	Можливість відслідковувати потенційно небажану або шкідливу поведінку окремих агентів та, при необхідності, завчасно їх блокувати
	3. Аналітика	Зручний графічний інтерфейс дасть можливість генерувати звіти, візуалізувати дані для представлення у доступному для сприйняття людиною вигляді

Оскільки на ринку моніторингу соціальних мереж вже існують продукти, які надають послугу підрахунку і моніторингу впливу, то нам доведеться конкурувати з ними. Із найкрупніших конкурентів можна виокремити Khoros, та Brandwatch. Порівняння даного проекту із ними можна побачити в таблиці 4.2.

Таблиця 4.2 - Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№	Технікоекономічні характеристики ідеї	(потенційні) товари/концепції конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Маг. проект	Khoros	Brandwatch			
	Вартість експлуатації	6000 грн/місяць	14000 грн/місяць	19000 грн/місяць			+
	Вартість обслуговування	6500 грн/місяць	-	-			+
	Розмір клієнтської бази	Жодного	2,000+ брендів	-	+		
	Аналітичні інструменти	Присутні	Присутні	Присутні		+	
	Підтримка декількох соціальних мереж	Присутня	Присутня	Присутня		+	
	Підтримка російськомовних соціальних мереж	Присутня	Відсутня	Відсутня			+

#### 4.4 Технологічний аудит ідеї проекту

Оскільки стартап-проект є більше програмним продуктом, то більшість технологій для його створення є доступними для використання. Хоча головну ідею проекту можна реалізувати і без додатків, фінальний продукт має мати мобільний додаток, сайт, десктопний додаток або все одразу, тому ці пункти і вказані в таблиці. Технології, що доцільно використовувати можна побачити на таблиці 4.3.

Таблиця 4.3 – Технологічна здійсненність ідеї проекту

№	Ідея проекту	Технології її реалізації	Наявність технології	Доступність технологій
1	Реалізація програмного забезпечення з використанням мови програмування Python, C/C++.	Мова програмування Python та бібліотеки numpy, scipy, pyqtgraph, requests, scrapy, celery, flask, flask-admin	Технології активно розвивається в області open source	Доступна для розробників
2	Створення кросплатформного графічного інтерфейсу для користувачів системи	Фреймворк Angular.js для реалізації кросплатформного графічного web-інтерфейсу на ОС Unix, Windows, Mac OS тощо.	Технології активно розвивається в області open source	Доступна для розробників

Кінець таблиці 4.3

3	Реалізація багатопотокового збору інформації	Використання модулів типу threading, multiprocessing, Scrapy мови Python	Технологія знаходиться у відкритому доступі	Доступна для розробників
4	Визначення сентименту зібраної інформації	Google cloud Natural Language service	Технологія знаходиться у закритому доступі та передбачає помісячну плату за користування	Доступна для розробників

Вибір перерахованих технологій зумовлюється відразу декількома вимогами: по-перше необхідно забезпечити достатньо високу швидкість збору і обробки інформації, так як для досягнення представницьких результатів необхідно буде моніторити досить велику групу множини користувачів; по-друге необхідно забезпечити підтримку багатоплатформності користувацького інтерфейсу.

#### 4.5 Аналіз ринкових можливостей запуску стартап-проекту

Попередня характеристика потенційного ринку стартап-проекту та характеристика потенційних клієнтів стартап-проекту представлені в таблиці 4.4 та таблиці 4.5 відповідно. Проводиться дефініція ринкових можливостей, які можна застосувати під час ринкового впровадження стартап-проекту, та загроз, які можуть

завадити реалізації проекту. Все це дозволить спроектувати напрями розвитку стартап-проекту із урахуванням обставин ринкового середовища, потреб потенційних клієнтів та пропозицій конкурентів.

Таблиця 4.4 - Попередня характеристика потенційного ринку проекту

№	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	2
2	Загальний обсяг продаж, грн/ум.од	20 000 000 ум.од
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Недискримінаційні якісні
5	Специфічні вимоги до стандартизації та сертифікації	Відсутні
6	Середня норма рентабельності в галузі (або по ринку), %	69,5%

Таблиця 4.5 – Характеристика потенційних клієнтів стартап-проекту

Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
--------------------------	--	---	-----------------------------

Кінець таблиці 4.5

Потреба моніторингу керованих впливів між користувачами соціальних мереж	1. Маркетологи 2. Аналітики 3. Держ-структури 4. Адміністрація соц-мереж	Потребують різну якість обслуговування, зацікавленість у різних групах користувачів	Зручність доступу до інформації, повнота та вичерпність інформації, можливість побудови графіків і звітів
--	---	---	---

Фактори загроз та фактори можливостей наведено в таблиці 4.6 та в таблиці 4.7 відповідно.

Таблиця 4.6 – Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Крадіжка інтелектуальної власності	Крадіжка ідеї або ключової інтелектуальної інновації	Відсудження прав інтелектуальної власності Забезпечення якіснішого захисту інформації Зміна методики шифрування приватного ключа Попередження користувачів із подальшою співпрацею для мінімізації фактору загрози

Кінець таблиці 4.6

2	Вихід на ринок нових конкурентів	З урахування помилок даного проекту можуть з'явитися рішення, які реалізують більшу кількість функціоналу та мають більше можливостей або працюють краще	Ретельний розгляд проблем потенційних клієнтів Спроба переорієнтувати рішення Консультації із спеціалістами
3	Недостатнє фінансування	Недостача початкових інвестицій для реалізації мінімально життєздатного продукту	Пошук нових джерел інвестицій
4	Відсутність ринку	Відсутність шляху збуту товару внаслідок помилкового орієнтування	Ретельний розгляд проблем потенційних клієнтів Консультації із спеціалістами Продовження роботи над шаблоном з метою вдосконалення
5	Програмні помилки або неефективні результати у тестовому проекті	У процесі розробки виникли програмні помилки внаслідок специфічної природи продукту	Затримка релізу проекту, продовження роботи над продуктом з метою вдосконалення



Таблиця 4.7 - Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Отримання інвестицій	Отримання капіталу що необхідний для реалізації продукту	Розробка продукту
2	Висока зацікавленість користувачів	Обіг використання продукту через веб-інтерфейс становить більше 1000 запитів в день	Підтримка стабільної роботи системи та проведення масштабування системи Збільшення цін на використання сервісу
3	Успішна маркетингова політика	В результаті проведеної маркетингової політики отримана висока зацікавленість користувачів	Продовження роботи над вдосконаленням продукту, написання статей, інструкцій, створення промо-акцій, безкоштовних версій продукту для залучення більшої аудиторії
4	Поглинання конкурентами	Пропозиція купівлі продукту або розроблених технологій одним із конкурентів	Розвиток розроблених технологій Оцінка вартості розроблених технологій

Кінець таблиці 4.7

5	Ліквідація конкурента	Конкурент ліквідував свою компанію у результаті власного бажання або зовнішніх чинників	Проведення маркетингової кампанії для монополізації ринку
---	-----------------------	---	---

Ступеневий аналіз конкуренції на ринку та аналіз конкуренції в галузі за М. Портером надані в таблицях 4.8 та таблиці 4.9 відповідно. Конкурентний аналіз спрямований на визначення можливостей, загроз і відшукування стратегічних невизначеностей, що можуть створюватися конкурентами, оскільки результати роботи не знаходяться в відкритому доступі і являються найбільш цінним активом проекту, відповідно постає питання конкуренції.

Таблиця 4.8 - Ступеневий аналіз конкуренції на ринку

№	Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1	Олігополія	Незначна кількість конкурентів Велика ринкова сила Схожість використовуваних технологій	Інформування ринку щодо появи нової вебслужби Співпраця із провідними сервісами

Кінець таблиці 4.8

2	Галузевий	Загроза появи нових конкурентів Ринкова влада споживачів Висока потреба у послугах	Інформування ринку щодо якості використовуваної новаторської технології Пропозиція гнучких цін
3	Внутрішньогалузева	Діяльність в одній галузі економіки Надання сервісів одного типу	Зменшення вартості сервісу Примноження каналів розподілу
4	Товарно-видова	Надання різних сервісів одного виду	Маркетингова політика
5	Цінова	Використання цін для покращення економічних умов збуту	Зменшення вартості сервісу Використання нових каналів розподілу
6	Марочна	Пропозиція схожого сервісу Спільна цільова аудиторія	Інформування ринку щодо якості використовуваної новаторської технології Примноження каналів розподілу

Аналіз конкуренції за М. Портером дало більш детально інформацію про стан ринку і можливості конкурентів.

Таблиця 4.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Khoros Brandwatch Kred	Цінність ідеї та її якісна реалізація	Немає	Споживачі прямо впливають на успішність продукту, оскільки сплачують кошти за ліцензію користування	Конкуренти можуть стати більш привабливими для клієнта або стануть надавати якісніші послуги
Висновки	Конкурентів небагато, відсутність монополії, проте конкурентна боротьба значна за рахунок хорошої якості проектів конкурентів	Є можливість виходу на ринок; ідентичних продуктів немає і не передбачається	Немає постачальників	Так, реалізація потреб клієнтів вирішує успішність проекту	Достатньо важко презентувати на клієнтів, які вже використовують інший готовий продукт

Навіть при врахуванні конкурентів розроблена система має низку переваг, які наведені в таблиці 4.10.

Таблиця 4.10 — Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
	Унікальність сервісу	Розроблений продукт представляє унікальну технологію імплементації моніторингу впливів між користувачами соц-мереж
	Цінова політика	Продукт пропонує найбільш вигідні тарифи порівняно з конкурентами
	Потреба в інвестиціях	Потреба в інвестиціях є, оскільки такого роду системи можуть вдосконалюватись та покращуватись довгий час за рахунок додавання нових функцій на вимоги користувачів
	Команда	Інтелектуальний рівень команди розробників дає можливість якісно та швидко розроблювати необхідний функціонал. При цьому, важливим фактором є наявність технічної освіти.

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін шаблону проектування

№	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні із розробленою системою						
			-3	-2	-1	0	+1	+2	+3
1.	Унікальність сервісу	15						+	
2.	Цінова політика	20					+		
3.	Потреба в інвестиціях	7				+			

Кінець таблиці 4.11

4.	Команда	15						+	
----	---------	----	--	--	--	--	--	---	--

Складений SWOT-аналіз на основі вище отриманих даних відображено в таблиці 4.12

Таблиця 4.12 — SWOT-аналіз стартап-проекту

<p>Сильні сторони:</p> <p>Якість та довготривалість послуги</p> <p>Привабливі тарифи</p> <p>Додаткові функції та сервіси</p> <p>Супровід клієнта</p> <p>Зручний графічний інтерфейс користувача</p> <p>Невелика кількість сильних конкурентів;</p>	<p>Слабкі сторони: Недостача стартових капіталовкладень</p> <p>Бізнес модель залежить від політики окремих бізнесів</p> <p>Необхідність входу на ринок в умовах конкуренції</p>
<p>Можливості:</p> <p>Інвестиції</p> <p>Реалізація бізнес-моделі</p> <p>Висока зацікавленість цільової аудиторії</p>	<p>Загрози:</p> <p>Крадіжка інтелектуальної власності</p> <p>Відмова фінансуванні</p> <p>Вихід на ринок нових конкурентів;</p> <p>Недовіра користувачів</p>

Визначення альтернатив було зроблено на основі часових термінів та ймовірності отримання ресурсів і відображено в таблиці 4.13

Таблиця 4.13 - Альтернативи ринкового впровадження стартап-проекту

№	Альтернатива (орієнтований комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Створення власної компанії	Ймовірно	6 місяців
2	Маркетингова кампанія для приваблювання користувачів	Ймовірно	2 місяці
3	Безкоштовний пробний період	Дуже ймовірно	1 місяць
4	Пошук бізнесів іншої галузі для співпраці	Мало ймовірне	2 місяці

Найкраща альтернатива — безкоштовний пробний період для кожного користувача, який вирішить спробувати дану систему.

#### 4.6 Розробка ринкової стратегії проекту

Для розробки ринкової стратегії проекту необхідно спершу визначити стратегії охоплення ринку, а саме описати характеристики потенційних клієнтів стартап-проекту. Ця інформація наведена в таблиці 4.14.

Таблиця 4.14 Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Малі компанії та приватні особи	Низька	35%		Високі бар'єри входу
2	Великі компанії	Висока	85%		Низькі бар'єри входу
3	Середні компанії	Середня	65%		Низькі бар'єри входу
Які цільові групи обрано: Великі та середні компанії					

За результатами аналізу потенційних клієнтів, було обрано дві цільові групи: великі та середні компанії і на їх основі в таблиці 4.15 було обрано стратегію ринкового впливу.

Таблиця 4.15 - Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції	Базова стратегія розвитку
Пропонування програмного комплексу для великих і середніх компаній	Диференційований маркетинг	Здатність пропонувати кращий та широкий спектр функціоналу	Стратегія диференціації



Визначення базової стратегії конкурентної поведінки та визначення стратегії позиціонування представлені в таблиці 4.16 та в таблиці 4.17 відповідно.

Таблиця 4.16 - Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
Ні	Забирати та залучати нових	Ні	Стратегія лідера. Розширення первинного попиту

Таблиця 4.17 - Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспромо жні позиції власного стартап- проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)

Кінець таблиці 4.17

Якість та правдивість запропонованої інформації Зручний інтерфейс користувача Підтримка клієнта Можливість налаштування під екзотичні потреби клієнта Гнучкі ціни Оперативний зворотній зв'язок	Стратегія диференціації	Розширення базового функціоналу	Вплив в соціальних мережах Моніторинг впливів в соціальних мережах Система управління впливами в соціальних мережах
--	-------------------------	---------------------------------	---

#### 4.7 Розробка маркетингової програми стартап-проекту

Першим кроком до формування маркетингової концепції товару, який отримає споживач. Для цього у таблиці 4.18 підсумовано результати попереднього аналізу конкурентоспроможності послуги.

В таблиці 4.18 визначено ключові переваги концепції потенційного послуги. Таблиця 4.18 – Визначення ключових переваг концепції потенційного послуги

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
---	---------	----------------------------	--

Кінець таблиці 4.18

1	Попит аналіз та моніторинг впливів між користувачами соціальних мереж	Зручний інтерфейс Низькі ціни	Якість послуги Інноваційність підходу Цінова перевага
2	Формування звітів по діяльності окремих користувачів	Гарантоване спрощення роботи, що економить час	Зручність у користуванні Інформативність

На даному етапі необхідно розробити трирівневу маркетингову модель, де уточнюється ідея продукту, його фізичні складові, особливості процесу його надання. Ці дані наведено в таблиці 4.19.

Таблиця 4.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Система моніторингу та аналізу впливів між користувачам соціальних мереж		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	Кількість		1 шт.
	Якість: стандарти якості написання технічної документації		
	Марка: Відсутня		
III. Товар із підкріпленням	Тестовий програмний продукт		
	Програмний продукт, технічна підтримка та підписка на оновлення		
За рахунок чого потенційний товар буде захищено від копіювання: весь виконуваний код, що представляє цінність буде зберігатися на захищених серверах			

Таблиця 4.20 – Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товарианалоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
—	14000 - 25000 грн	>50000 грн	6000 - 20000

Таблиця 4.21 – Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Платна пілписка	Створення проектної документації про програмний комплекс	Канал нульового рівня	Дистрибуція через Інтернет

Останньою складовою маркетингової програми є розробка концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів. Результати наведено в таблиці 4.22.

Таблиця 4.22 — Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
---------------------------------------	--	--	----------------------------------	--------------------------------

Кінець таблиці 4.22

Ведення бізнесу в інтернеті	Прямі неофіційні	Послідовність в реалізації обраної позиції  Доступність та об'єктивність інформації про фірму і товар	Формування у цільовій аудиторії обізнаності про появу нового сервісу	Рационалістична стратегія реклами
-----------------------------------	------------------	--	---	---

#### Висновки до розділу 4

Під час проведення дослідження магістерської дисертації у якості стартап-проекту було створено відповідну аргументовану документацію. Протягом усього процесу дослідження було досягнуто повне розуміння цілей проекту, його переваги та недоліки, які в свою чергу дали можливість точніше визначити мету проекту та користувацьку аудиторію програмного застосунку, визначити можливі методи монетизації, маркетингову стратегію для комерційного ринку.

Основними споживачами системи, що розроблена в рамках магістерської дисертації, є великий та середній бізнеси.

На початковому етапі входження у ринок попит може бути не дуже високим через наявну конкуренцію та відсутність клієнтської бази.

Перевагами проекту є:

- підтримка декількох соціальних мереж;
- простота у використанні;
- кросплатформність;

- зручний інтерфейс користувача;

## ВИСНОВКИ

Результатом даної роботи є запропонована власна модель моніторингу впливів між користувачами соціальних мереж. Дана модель може бути застосована до більшості відомих популярних соціальних мереж. Перевагою даної моделі є той факт, що в ній не використовуються випадкові значення для підрахунку індексів впливовості, тобто дана модель не є стохастичною.

Отримана реалізація є мінімально необхідним програмним продуктом, який покриває всі вимоги до рішення проблеми аналізу поведінки агента та виявлення впливів між агентами у соціальній системі.

Розроблене програмне забезпечення дозволяє збирати дані у обраній соціальній мережі серед агентів пов'язаних спільними ідеями та інтересами (товариства, публічні сторінки), аналізувати їх, представляти у вигляді придатному до застосування до обраної математичної моделі та на підставі всього вище перерахованого заключення висновків щодо поведінки агента та його впливу на інших агентів мережі.

До списку задач, які можна вирішити за допомоги запропонованої моделі можна віднести:

- 1) моніторинг активності звичайних користувачів;
- 2) виділення найбільш активних користувачів;
- 3) відслідковування змін думок агентів;
- 4) виявлення впливів між агентами;
- 5) виявлення найбільш впливових агентів.

Також, в ході написання роботи був розроблений стартап-проект по впровадженню запропонованої моделі в реальну систему по моніторингу за впливами між агентами в соціальних мережах.

## СПИСОК ДЖЕРЕЛ ПОСИЛАННЯ

1. Glossary on Control Theory and its Applications. – [Електронний ресурс] / Інтернет-портал – Режим доступу: <http://glossary.ru>.
2. Oxford English Dictionary. – [Електронний ресурс] / Інтернет-портал – Режим доступу: <http://www.askoxford.com>.
3. A Study of Normative and Informational Social Influences upon Individual Judgment [Текст] / Deutch M., Gerard H. B. // Journal of Abnormal and Social Psychology. – 1955 – №2. С. 37 – 38.
4. Модели и методы организационного управления инновационным развитием фирмы [Текст] / Иващенко А. А., Новиков Д. А. – 2006.
5. Threshold Models of Collective Behavior [Текст] / Granovetter M. // American Journal of Sociology. – 1978 – С. 1 – 5.
6. Social Networks and Threshold Models of Collective Behavior. Preprint. [Текст] / Rolfe M. // Chicago: University of Chicago – 2004 – С. 15 – 18.
7. Contagion [Текст] / MORRIS S. // The Review of Economic Studies. – 2000
8. Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth [Текст] / Goldenberg J., Libai B., Muller E. // Marketing Letters. – 2001.
9. Maximizing the Spread of Influence through a Social Network [Текст] / Kempe D., Kleinberg J., Tardos E. // Proceedings of the 9-th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. – 2003 – С. 20 – 23.
10. Cellular Automata: A Discrete View of the World. [Текст] / Schiff J. L. // NY: Wiley – 2007 – С. 2 – 6.
11. TextBlob: Simplified Text Processing – [Електронний ресурс] / Інтернет-портал – Режим доступу: <https://textblob.readthedocs.io/en/dev/quickstart.html#sentiment-analysis>



12. Learning Influence among Interacting Markov Chains [Текст] / Zhang D., Gatica-Perez D., Bengio S., Roy D. // Neural Information Processing Systems (NIPS), 2005. – P. 132-141.
13. Graphical Models for Recognizing Human Interactions [Текст] / Oliver N., Rosario B., Pentland A. // Proceedings of International Conference on Neural Information and Processing Systems (NIPS). – 1998. – С. 924-930.
14. Dynamical Systems Trees [Текст] / Howard A., Jebara T. // Uncertainty in Artificial Intelligence. – 2003. – С. 260-267.
15. A formal theory of social power [Текст] / French J. R. // The Psychological Review. – 1956. – №63. – С. 181-194.
16. A Model of Influence in a Social Network – [Электронный ресурс] / Grabisch M., Rusinovska A. – URL: <http://halshs.archives-ouvertes.fr/docs/00/34/44/57/PDF/B08066.pdf>.
17. A Theory of Decisional Power [Текст] / Hoede C., Bakker R. // Journal of Mathematical Sociology. – 1982. – №8. – С. 309-322.
18. Class and Committees in a Norwegian Island Parish [Текст] / Barnes J. A. // Human Relations. – 1954. – №7. – С. 39-58.
19. Анализ социальных сетей. Автореф [Текст] / Градосельская Г. В. – Москва, 2001.
20. Моделирование социальных сетей / Давыденко В. А., Ромашкина Г. Ф. // Вестник Тюменского государственного университета. – 2005. – №1. – С. 68-79.

## Додаток А Код програмної реалізації

### reshares\_grabber.py

```

import re
import json
import requests

from bs4 import BeautifulSoup

from .utils import json_resp

from http.cookiejar import CookieJar, Cookie

class ResharesGrabber:
    def __init__(self, dyn, fb_dtsg_ag, cookies, from_file=False, proxy=None):
        self.dyn = dyn
        self.fb_dtsg_ag = fb_dtsg_ag
        self.user_id = None
        self.proxy = proxy

        self.sess = requests.session()

        cookie_jar = CookieJar()
        if from_file:
            with open(cookies) as f:
                cookies = f.read()
            if not isinstance(cookies, (list, tuple)):
                cookies = json.loads(cookies)
            for cookie in cookies:
                if cookie['name'] == 'c_user':
                    self.user_id = cookie['value']
                cookie_jar.set_cookie(
                    Cookie(version=0, name=cookie['name'], value=cookie['value'],
port=None, port_specified=False,
                        domain=cookie['domain'], domain_specified=False,
domain_initial_dot=False, path=cookie['path'],
                        path_specified=True, secure=cookie['secure'],
expires=cookie.get('expirationDate'),

```

```

        discard=True, comment=None,
        comment_url=None, rest={'HttpOnly': None}, rfc2109=False))
self.sess.cookies = cookie_jar

if self.proxy:
    if '://' not in self.proxy:
        self.sess.proxies = {
            'http': 'http://' + self.proxy,
            'https': 'http://' + self.proxy
        }
    else:
        self.sess.proxies = {
            'http': self.proxy,
            'https': self.proxy
        }
    self.sess.trust_env = False

self.sess.headers.update({'Dnt': '1',
                           'User-Agent': 'Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.70
Safari/537.36',
                           'Viewport-Width': '1536',
                           'Accept': '*/*',
                           'Sec-Fetch-Site': 'same-origin',
                           'Sec-Fetch-Mode': 'cors',
                           'Referer': 'https://www.facebook.com/',
                           'Accept-Encoding': 'gzip, deflate',
                           'Accept-Language': 'en-US,en;q=0.9,ru-
UA;q=0.8,ru;q=0.7,uk;q=0.6'})

@staticmethod
def _process_shares_resp(response):
    reshares = []
    next_cursor = None

    try:
        next_cursor = re.search(r'cursor"\":\"(.*)\""', response).group(1)
    except AttributeError:
        pass

    j_r = json_resp(response)
    if j_r.get('payload'):

```

```

        s = BeautifulSoup(j_r['payload'], 'lxml')
    else:
        s = BeautifulSoup(j_r['jsmods']['markup'][0][1]['__html'], 'lxml')

    for div in s.select('div[data-ft*="top_level_post_id"]'):
        share_data = json.loads(div['data-ft'])
        share_data.pop('story_location', None)
        share_data.pop('story_attachment_style', None)
        share_data.pop('fbfeed_location', None)

        share_data['url'] = 'https://www.facebook.com' + div.select('div[data-
testid="story-subtitle"] a')[0]['href']

        text = div.select('div[data-testid="post_message"]')
        if text:
            text = text[0].text.strip().replace('Показать перевод', '')
        else:
            text = ''
        share_data['text'] = text

        name_tags = div.select('a[data-hovercard*="/ajax/hovercard/"]')
        try:
            share_data['name'] = name_tags[1].text.strip()
        except IndexError:
            share_data['name'] = name_tags[0]['title']

        reshares.append(share_data)

    return reshares, next_cursor

def _open_shares(self, post_id):
    params = [
        ('target_fbid', post_id),
        ('av', self.user_id),
        ('fb_dtsg_ag', self.fb_dtsg_ag),
        ('__asyncDialog', 1),
        ('__user', self.user_id),
        ('__a', 1),
        ('__dyn', self.dyn),
        ('__csr', ''),
        ('__req', 23),
        ('__be', 1),

```

```

        ('__pc', 'PHASED:DEFAULT'),
        ('dpr', '1.5'),
        ('__rev', 1001347882),
        ('__s', ':lxzgdi:kemf5u'),
        ('__hsi', '6752111825873053825-0'),
        ('jazoest', 27863),
        ('__spin_r', 1001347882),
        ('__spin_b', 'trunk'),
        ('__spin_t', 1572086889)
    ]

    r = self.ssess.get('https://www.facebook.com/ajax/shares/view',
                       params=params,
                       verify=False)

    return self._process_shares_resp(r.text)

def _shares_pagination(self, post_id, cursor, on_init=False):
    data = {"target_fbid": int(post_id),
            "cursor": cursor}

    if on_init:
        data['pager_fired_on_init'] = True

    params = [
        ('fb_dtsg_ag', self.fb_dtsg_ag),
        ('data', json.dumps(data, separators=(',', ':'))),
        ('__asyncDialog', 1),
        ('__user', self.user_id),
        ('__a', 1),
        ('__dyn', self.dyn),
        ('__csr', ''),
        ('__req', 23),
        ('__be', 1),
        ('__pc', 'PHASED:DEFAULT'),
        ('dpr', '1.5'),
        ('__rev', 1001347882),
        ('__s', ':lxzgdi:kemf5u'),
        ('__hsi', '6752111825873053825-0'),
        ('jazoest', 27863),
        ('__spin_r', 1001347882),
        ('__spin_b', 'trunk'),
        ('__spin_t', 1572086889)
    ]

```

```

    ]

    r =

self.sess.get('https://www.facebook.com/ajax/pagelet/generic.php/ResharesPagelet',
              params=params,
              verify=False)

    return self._process_shares_resp(r.text)

def get_post_reshares(self, post_id):
    reshares, next_cursor = self._open_shares(post_id)

    on_init = True
    while next_cursor:
        n_reshares, next_cursor = self._shares_pagination(post_id,
next_cursor, on_init=on_init)
        reshares += n_reshares
        on_init = False

    return reshares

```

## analyze.py

```

all_activity = []
posts = set()

reactors = db.actors.find({}, {'_id':0, 'user_id': 1, 'post_id':1})
for x in reactors:
    all_activity.append(x['user_id'])
    posts.add(x['post_id'])

comments = db.comments.find({}, {'_id':0, 'author_id': 1, 'post_id':1})
for x in comments:
    all_activity.append(x['author_id'])
    posts.add(x['post_id'])

reshares = list(db.reshares.find({}, {'_id':0, 'content_owner_id_new': 1,
'post_id': 1}))
for x in reshares:

```

```

all_activity.append(x['content_owner_id_new'])
posts.add(x['post_id'])

posts = db.posts.find({'post_id': {'$in': list(posts)}})
posts = {x['post_id']: x for x in posts}
print(len(posts))
top_users = Counter(all_activity).most_common(100)
print(top_users)

dt_mapping = {dateutil.parser.parse('2019-09-26T00:00:00') + timedelta(days=x): x
for x in range(32)}
print(dt_mapping)
n = 0
for user in top_users:
    user_data = [[x, 0] for x in range(1, 33)]
    reactors = db.reactors.find({'user_id': user[0]}, {'_id': 0, 'user_id': 1,
'post_id': 1})
    for x in reactors:
        date = posts[x['post_id']]['created_at'].replace(hour=0, minute=0,
second=0, microsecond=0)
        try:
            user_data[dt_mapping[date]][1] += 0.2
        except KeyError:
            pass

    comments = db.comments.find({'author_id': user[0]}, {'_id': 0, 'author_id': 1,
'post_id': 1})
    for x in comments:
        date = posts[x['post_id']]['created_at'].replace(hour=0, minute=0,
second=0, microsecond=0)
        try:
            user_data[dt_mapping[date]][1] += 0.3
        except KeyError:
            pass

    reshares = list(db.reshares.find({'content_owner_id_new': user[0]}, {'_id': 0,
'content_owner_id_new': 1, 'post_id': 1}))
    for x in reshares:
        date = posts[x['post_id']]['created_at'].replace(hour=0, minute=0,
second=0, microsecond=0)
        try:

```

```
        user_data[dt_mapping[date]][1] += 0.5
    except KeyError:
        pass

x_vals = [x[0] for x in user_data]
y_vals = [x[1] for x in user_data]
plt.plot(x_vals, y_vals)
plt.xlabel('Days')
plt.ylabel('User`s "Daily score"')
plt.savefig(f'top_100_users/{user[0]}')
plt.clf()
```



```

import re
import time
import json
import string
import random
import requests

from ..errors import InvalidType, IncompleteNesting
from ..utils import logger, gzip_body, rand_uuid, from_json

class PagesMixin:
    sess: requests.session()
    access_token: str
    user_id: str
    device_data: dict
    locale: str
    styles_id: str

    # barcelona_id: 197394889304
    def _process_page_post_resp(self, response, page_id, created_after,
doc_process=None):
        j_r = from_json(response.text)
        if isinstance(j_r, list):
            single_j_r = j_r[0]
            for _json in j_r[1:]:
                if 'page_info' in _json.get('data', {}):
                    single_j_r['data']['page_info'] = _json['data']['page_info']
            j_r = single_j_r

        posts = []
        if j_r.get('data'):
            page_key = list(j_r['data'].keys())[0]
            if j_r['data'].get(page_key) and j_r['data'][page_key].get('id') !=
str(page_id):
                logger.error(f'[FB] Unexpected response
(page_id:{page_id}): \n{response.text}')
                return None, None

            if 'page_info' in j_r['data']:
                has_more = j_r['data']['page_info'].get('has_next_page')
                end_cursor = j_r['data']['page_info'].get('end_cursor')
            else:
                has_more =
j_r['data'][page_key]['content_list_view']['cards']['page_info'].get('has_next_pag
e')
                end_cursor =
j_r['data'][page_key]['content_list_view']['cards']['page_info'].get('end_cursor')

            for _ in
range(len(j_r['data'][page_key]['content_list_view']['cards']['edges']) * 2):
                log_extra = self._gen_log_extra(name='pages_event',
                                                event='IMPRESSION',
                                                event_location='PAGES_HOME_TAB',
                                                event_target='CONTENT_CARD',
card_type='post', context_id=page_id)
                self.add_log('pages_event', log_extra)
                self._log_page_home_tab(page_id)

            for node in
j_r['data'][page_key]['content_list_view']['cards']['edges']:
                try:
                    post = node['node'].get('components', [])[0].get('story')

```

```

except IndexError:
    logger.error(f'Unexpected post_type:\n{node}')
    continue

if not post:
    continue

log_extra = self._gen_log_extra(name='vpv_duration',
pigeon_module='pages_public_view', tracking_data=post['tracking'])
self.add_log('vpv_duration', log_extra)
self._log_video_playing(post)

# logger.debug(f"{node['node']['cache_id']}: {post.get('message',
{}})}")

if post['creation_time'] > created_after:
    if doc_process:
        posts += [x for x in doc_process(post) if x]
    else:
        posts.append(post)
else:
    logger.debug('Out of date limit.')
    has_more = False
    break

if has_more and end_cursor:
    return posts, end_cursor
else:
    return posts, None

else:
    logger.error(f'[FB] Unexpected response
(page_id:{page_id}):\n{response.text}')
    return None, None

def _get_page_posts_first(self, page_id, **kwargs):
    self.sess.headers.update({'X-FB-Friendly-Name':
'PageTabContentDataQuery'})

    variables = {"0": str(page_id), "148": 270, "19": 96, "85": 360, "138":
120, "84": 2048,
                "139": "image/x-auto", "156": "3", "137": 5, "193": 1}

    post_data = [
        ('doc_id', '2590921827627090'),
        ('method', 'post'),
        ('locale', self.locale),
        ('pretty', 'false'),
        ('format', 'json'),
        ('purpose', 'fetch'),
        ('fb_api_client_context',
'{"request_source":"graph_search_results_page_blended"}'),
        ('variables', json.dumps(variables, separators=(',', ':'))),
        ('fb_api_req_friendly_name', 'PageTabContentDataQuery'),
        ('fb_api_caller_class', 'graphservice'),
        ('fb_api_analytics_tags', '["GraphServices","At_Connection","pages
android tab surface timeline"]'),
        ('server_timestamps', 'true')
    ]

    r = self.sess.post('https://graph.facebook.com/graphql',
                        data=gzip_body(post_data))

    return self._process_page_post_resp(r, page_id, **kwargs)

```

```

    def _get_page_posts_pagination(self, page_id, cursor, posts_per_page=3,
**kwargs):
        self.ssess.headers.update(
            {'X-FB-Friendly-Name':
'PageTabContentDataQuery_At_Connection_Pagination_Page_cards_connection'})

        variables = {
            "paginationPK": str(page_id)}

        post_data = [
            ('doc_id', '2062540407165984'),
            ('method', 'post'),
            ('locale', self.locale),
            ('pretty', 'false'),
            ('format', 'json'),
            ('purpose', 'fetch'),
            ('fb_api_client_context',
'{"load_next_page_counter":1,"client_connection_size":1}'),
            ('variables', json.dumps(variables, separators=(',', ':'))),
            ('fb_api_req_friendly_name',
'PageTabContentDataQuery_At_Connection_Pagination_Page_cards_connection'),
            ('fb_api_caller_class', 'ConnectionManager'),
            ('fb_api_analytics_tags', f'["GraphServices","At_Connection"]'),
            ('server_timestamps', 'true')
        ]

        r = self.ssess.post('https://graph.facebook.com/graphql',
            data=gzip_body(post_data))

        return self._process_page_post_resp(r, page_id, **kwargs)

    def _page_base_info(self, page_id):
        self.ssess.headers.update({'X-FB-Friendly-Name': 'PageBaseInfoQuery'})

        variables = {"0": str(page_id)}

        post_data = [
            ('doc_id', '2195459427141316'),
            ('method', 'post'),
            ('locale', self.locale),
            ('pretty', 'false'),
            ('format', 'json'),
            ('fb_api_client_context', '{"request_source":"unknown"}'),
            ('variables', json.dumps(variables, separators=(',', ':'))),
            ('fb_api_req_friendly_name', 'PageBaseInfoQuery'),
            ('fb_api_caller_class', 'graphservice'),
            ('fb_api_analytics_tags', '["GraphServices"]'),
            ('server_timestamps', 'true')
        ]

        r = self.ssess.post('https://graph.facebook.com/graphql',
            data=gzip_body(post_data))

        return

    def _page_nav_bar_query(self, page_id):
        self.ssess.headers.update({'X-FB-Friendly-Name': 'PageNavBarQuery'})

        variables =
{"5":{"styles_id":self.styles_id,"pixel_ratio":3},"3":True,"149":"3","0":str(page_
id)}

        post_data = [

```

```

        ('doc_id', '2189915554464574'),
        ('method', 'post'),
        ('locale', self.locale),
        ('pretty', 'false'),
        ('format', 'json'),
        ('fb_api_client_context', '{"request_source":"unknown"}'),
        ('variables', json.dumps(variables, separators=(',', ':'))),
        ('fb_api_req_friendly_name', 'PageNavBarQuery'),
        ('fb_api_caller_class', 'graphservice'),
        ('fb_api_analytics_tags', '["GraphServices"]'),
        ('server_timestamps', 'true')
    ]

    r = self.ssess.post('https://graph.facebook.com/graphql',
                        data=gzip_body(post_data))

    return

    def _log_page_home_tab(self, page_id):
        logging_card_types = ['about', 'page_transparency', 'create_page_upsell',
                              'recommendations', 'videos',
                              'photos']
        for card_type in logging_card_types:
            log_extra = self._gen_log_extra(name='pages_event',
                                            event='IMPRESSION',
                                            event_location='PAGES_HOME_TAB',
                                            event_target='CONTENT_CARD',
                                            card_type=card_type, context_id=page_id)
            self.add_log('pages_event', log_extra)

    def _get_community_info(self, page_id):
        self.ssess.headers.update({'X-FB-Friendly-Name':
                                   'PageTabContentDataQuery'})

        variables = {"0": str(page_id)}

        post_data = [
            ('doc_id', '2590921827627090'),
            ('method', 'post'),
            ('locale', self.locale),
            ('pretty', 'false'),
            ('format', 'json'),
            ('purpose', 'fetch'),
            ('fb_api_client_context', '{"request_source":"pages_public_view"}'),
            ('variables', json.dumps(variables, separators=(',', ':'))),
            ('fb_api_req_friendly_name', 'PageTabContentDataQuery'),
            ('fb_api_caller_class', 'graphservice'),
            ('fb_api_analytics_tags', '["GraphServices","At_Connection","pages
android tab surface community_tab"]'),
            ('server_timestamps', 'true')
        ]

        r = self.ssess.post('https://graph.facebook.com/graphql',
                            data=gzip_body(post_data))

        # results = re.findall(r'\\\\"((\\d+(,|.)?\\d+([KM])?)| (\\d+([KM]))))\\\\"',
        r.text)

        total_likes =
        re.search(r'\\\\"((\\d+(,|.)?\\d+)?([KM])?)| (\\d+([KM]))\\\\"'.*Total Likes',
        r.text).groups()[0]
        total_likes = int(total_likes.replace(',', ' ').replace('.',
        '').replace('K', '000').replace('M', '000000'))

```

```

        try:
            total_followers =
re.search(r'\\\\"((\\d+[,|.]?(\\d+)?([KM])?)|(\\d+([KM]))))\\\\"}.*Total Follows',
r.text).groups()[0]
            total_followers = int(total_followers.replace(',', '').replace('.',
'').replace('K', '000').replace('M', '000000'))
        except AttributeError:
            total_followers = 0

    return total_likes, total_followers

def get_page_info(self, page_id, get_alias=False):
    self.sess.headers.update({'X-FB-Friendly-Name': 'PageSurfaceHeaderQuery'})

    variables = {"92": "3", "89": str(page_id),
                "88": {"using_white_navbar": True, "styles_id":
self.styles_id,
                    "pixel_ratio": 3}, "100": True}

    post_data = [
        ('doc_id', '2225571600875212'),
        ('method', 'post'),
        ('locale', self.locale),
        ('pretty', 'false'),
        ('format', 'json'),
        ('purpose', 'fetch'),
        ('fb_api_client_context',
'{"request_source": "graph_search_results_page_blended"}'),
        ('variables', json.dumps(variables, separators=(',', ':'))),
        ('fb_api_req_friendly_name', 'PageSurfaceHeaderQuery'),
        ('fb_api_caller_class', 'graphservice'),
        ('fb_api_analytics_tags', ['"GraphServices"]'),
        ('server_timestamps', 'true')
    ]

    r = self.sess.post('https://graph.facebook.com/graphql',
                      data=gzip_body(post_data))

    j_r = r.json()

    if not j_r.get('data', {}).get('page'):
        raise InvalidType

    # log fact of page visitation
    log_extra = self._gen_log_extra(name='visit_page', context_id=page_id)
    self.add_log('visit_page', log_extra)

    log_extra = self._gen_log_extra(name='pages_event', event='IMPRESSION',
event_target='PAGE_PROFILE',
                                context_id=page_id)
    self.add_log('pages_event', log_extra)

    log_extra = self._gen_log_extra(name='page_load_successful',
context_id=page_id)
    self.add_log('page_load_successful', log_extra)

    log_extra = self._gen_log_extra(name='pages_event', event='IMPRESSION',
event_location='PAGES_HEADER',
                                context_id=page_id)
    self.add_log('pages_event', log_extra)

    log_extra = self._gen_log_extra(name='pages_event', event='impression',

```

```

event_location='pages__header',
event_target='social_context',
context_id=page_id)
self.add_log('pages_event', log_extra)

log_extra = self._gen_log_extra(name='pages_event', event='IMPRESSION',
event_location='PAGES__HOME_TAB',
event_target='CONTENT_CARD',
card_type='about', context_id=page_id)
self.add_log('pages_event', log_extra)

log_extra = self._gen_log_extra(name='pages_event', event='impression',
event_location='pages__home_tab',
event_target='content_card_header_link',
context_id=page_id)
self.add_log('pages_event', log_extra)

log_extra = self._gen_log_extra(name='pages_event', event='impression',
event_location='pages__home_tab',
event_target='page_message',
context_id=page_id)
self.add_log('pages_event', log_extra)

# content_card_footer -----

log_extra = self._gen_log_extra(name='pages_event', event='impression',
event_location='pages__home_tab',
event_target='content_card_footer',
card_type='about', context_id=page_id)
self.add_log('pages_event', log_extra, tags=196)

log_extra = self._gen_log_extra(name='pages_event', event='impression',
event_location='pages__home_tab',
event_target='content_card_footer',
card_type='page_transparency',
context_id=page_id)
self.add_log('pages_event', log_extra, tags=196)

log_extra = self._gen_log_extra(name='pages_event', event='impression',
event_location='pages__home_tab',
event_target='content_card_footer',
card_type='recommendations',
context_id=page_id)
self.add_log('pages_event', log_extra, tags=196)

log_extra = self._gen_log_extra(name='pages_event', event='impression',
event_location='pages__home_tab',
event_target='content_card_footer',
card_type='videos', context_id=page_id)
self.add_log('pages_event', log_extra, tags=196)

log_extra = self._gen_log_extra(name='pages_event', event='impression',
event_location='pages__home_tab',
event_target='content_card_footer',
card_type='photos', context_id=page_id)
self.add_log('pages_event', log_extra, tags=196)

# -----

log_extra = self._gen_log_extra(name='pages_event', event='IMPRESSION',
event_location='pages__footer_launchpad',
event_target='PAGE_LAUNCHPAD',
context_id=page_id)

```

```

self.add_log('pages_event', log_extra)

self._log_page_home_tab(page_id)

try:
    for component in
j_r['data']['page']['page_component_based_header']['components']:
        for cover_item in component.get('cover_items', []):
            if cover_item.get('__typename') == 'PageVideoCoverItemData':
                video_sess_id =
''.join([random.choice(string.ascii_lowercase + string.digits) for _ in range(6)])
                log_extra = self._gen_log_extra(name='requested_playing',
                                                tracking_data=[],

context_id=cover_item['video']['id'],

video_sess_id=video_sess_id)
                self.add_log('requested_playing', log_extra)

except Exception as e:
    logger.error(f'[FB] Unexpected response get_page_info({page_id})
({str(e)})\n{r.text}')
    return

if not j_r.get('data'):
    logger.error(f'[FB] Unexpected response
get_page_info({page_id})\n{r.text}')
    return

page_name = None

for component in j_r['data'].get('page',
{}).get('page_component_based_header', {}).get('components', []):
    for action in component.get('primary_actions', []) +
component.get('extra_actions', []):
        if action.get('page', {}).get('id'):
            page_name = action.get('page', {}).get('name')
            break
    if page_name:
        break

if not page_name:
    raise InvalidType('Could not find page name.')

try:
    total_likes = int(re.search(r'\("((?:?!")\.)* people like this',
r.text).groups()[0].replace(',', ''))
except AttributeError:
    # logger.error(f'Failed to get subscribers_count.\nResponse:
{r.text}')
    total_likes = None

total_likes_, total_followers = self._get_community_info(page_id)

if not total_likes:
    total_likes = total_likes_

page_info = {
    'id': page_id,
    'name': page_name,
    'likes': total_likes,
    'followers': total_followers
}

```

```

    if get_alias:
        page_info['url'] = self.get_page_alias(page_id)
    else:
        page_info['url'] = f'https://www.facebook.com/{page_id}/'

    return page_info

def get_page_alias(self, page_id):
    self.sess.headers.update({'X-FB-Friendly-Name': 'TabDataQuery'})

    variables = {"3": "1", "2": 24, "136": "1",
                 "111": {"styles_id": self.styles_id, "pixel_ratio": 3},
                 "1": {"tab_action_type": "TAB_HOME",
                       "supported_actions": ["COPY_TAB_LINK", "DELETE_TAB",
                                             "CREATE_SHORTCUT", "REORDER_TABS",
                                             "SHARE_TAB", "VISIT_PAGE"],
                       "is_deeplink": False}, "0": page_id}

    post_data = [
        ('doc_id', '2078504935565788'),
        ('method', 'post'),
        ('locale', self.locale),
        ('pretty', 'false'),
        ('format', 'json'),
        ('variables', json.dumps(variables, separators=(',', ':'))),
        ('fb_api_req_friendly_name', 'TabDataQuery'),
        ('fb_api_caller_class', 'graphservice'),
        ('fb_api_analytics_tags', '["GraphServices"]'),
        ('server_timestamps', 'true')
    ]

    r = self.sess.post('https://graph.facebook.com/graphql',
                       data=gzip_body(post_data))

    try:
        j_r = r.json()
    except Exception as e:
        logger.error(f'[FB] Unexpected response ({str(e)}):\n{r.text}')
        return

    if not j_r.get('data'):
        logger.error(f'[FB] Unexpected response\n{r.text}')
        return

    for action in j_r['data'].get('page', {}).get('tabAdminSettingChannel',
{}).get('actions', []):
        if action.get('tab_type') == 'TAB_HOME':
            uri = action['uri']
            return uri

def _page_like_status(self, page_id):
    self.sess.headers.update({'X-FB-Friendly-Name': 'PageLikeStatusQuery'})

    post_data = [
        ('doc_id', '1644666315598373'),
        ('method', 'post'),
        ('locale', self.locale),
        ('pretty', 'false'),
        ('format', 'json'),
        ('fb_api_client_context', '{"request_source": "unknown"}'),
        ('variables', '{"0": %s}' % page_id),
        ('fb_api_req_friendly_name', 'PageLikeStatusQuery'),
    ]

```



```

        ('fb_api_caller_class', 'graphservice'),
        ('fb_api_analytics_tags', '["GraphServices"]'),
        ('server_timestamps', 'true')
    ]

    r = self.ssess.post('https://graph.facebook.com/graphql',
                        data=gzip_body(post_data)).json()

    return r.get('data', {}).get('node', {}).get('does_viewer_like')

def _like_page(self, page_id):
    self.ssess.headers.update({'X-FB-Friendly-Name': 'PageLike'})

    variables = {"0": {"tracking": [],
                        "source": "search",
                        "page_id": page_id,
                        "client_mutation_id": rand_uuid(),
                        "actor_id": self.user_id}
                }

    post_data = [
        ('doc_id', '1286979098028181'),
        ('method', 'post'),
        ('locale', self.locale),
        ('pretty', 'false'),
        ('format', 'json'),
        ('variables', json.dumps(variables)),
        ('fb_api_req_friendly_name', 'PageLike'),
        ('fb_api_caller_class', 'graphservice'),
        ('fb_api_analytics_tags', '["GraphServices"]'),
        ('server_timestamps', 'true')
    ]

    r = self.ssess.post('https://graph.facebook.com/graphql',
                        data=gzip_body(post_data)).json()

    return r.get('data', {}).get('node', {}).get('does_viewer_like')

def _fetch_page_tabs(self, page_id):
    self.ssess.headers.update({'X-FB-Friendly-Name': 'FetchPageTabListsQuery'})

    post_data = [
        ('doc_id', '2355897177831887'),
        ('method', 'post'),
        ('locale', self.locale),
        ('pretty', 'false'),
        ('format', 'json'),
        ('purpose', 'fetch'),
        ('fb_api_client_context',
         '{"request_source": "graph_search_results_page_blended"}'),
        ('variables', '{"0": "%s"}' % page_id),
        ('fb_api_req_friendly_name', 'FetchPageTabListsQuery'),
        ('fb_api_caller_class', 'graphservice'),
        ('fb_api_analytics_tags', '["GraphServices", "pages android tab list
query"]'),
        ('server_timestamps', 'true')
    ]

    r = self.ssess.post('https://graph.facebook.com/graphql',
                        data=gzip_body(post_data)).json()

    def get_page_posts(self, page_id, created_after=0, doc_process=None,
simulate=True):

```

```

"""

:param page_id: <str> : page id
:param created_after: <int> : timestamp
:param doc_process: <function> : callback function to process facebook
story
:param simulate: <bool> : simulate user activity
:return: <list> : list of posts
"""
if simulate:
    self.get_page_info(page_id)

    posts, next_cursor = self._get_page_posts_first(page_id,
created_after=created_after, doc_process=doc_process)

    if simulate:
        self._page_base_info(page_id)
        self._page_like_status(page_id)
        self._page_nav_bar_query(page_id)
        self._fetch_page_tabs(page_id)
        self.get_page_alias(page_id)

    while next_cursor:
        time.sleep(random.uniform(0.5, 2))
        n_posts, next_cursor = self._get_page_posts_pagination(page_id,
                                                                next_cursor,

created_after=created_after,

doc_process=doc_process)
        posts += n_posts

    return posts

```