

ЯК ЛЮДИ СТВОРЮЮТЬ ЗВ'ЯЗКИ - АНАЛІЗ СОЦІАЛЬНИХ МЕРЕЖ

Вінницький національний технічний університет

Анотація.

У статті на основі різноманітних джерел проведено аналіз соціальних мереж і їх вплив на розповсюдження інформації у суспільстві.

Ключові слова: соціальна мережа; суспільство; користувач; управління інформаційною безпекою; витік інформації.

Abstract.

In the article on the basis of various sources an analysis of social networks and their impact on the dissemination of information in society.

Keywords: social network; society; user; information security management; leakage of information.

Комп'ютерні мережі міцно увійшли в людське життя. І йти не збираються. Перерахуємо мети, яким служать комп'ютерні мережі.

Люди дізнаються нове. Спілкуються з близькими. Укладають угоди. Беруть участь у спільній розробці проєктів. Управляють процесами.

В інтернеті багато сервісів. Незважаючи на удавану різноманітність, основні сервіси досить централізовані. Google має сервера в певному місці, мільйони запитів проходять через цей центр і аналізуються. Facebook, Vkontakte також мають свої центри, мільйони повідомлень людей проходять через центральний сервер соціальної мережі і відслідковуються. Skype на сьогодні також має виділені сервери, через які проходить весь трафік. Мережі електронної комерції, розробки проєктів, форуми - все знаходиться на чийось центральних серверах. Вся інформація проходить через єдині центри і зберігається там. Це загальновідомо. В чому проблема?

Проблема перша. Сьогодні користувачі довіряють свою інформацію знеособленим корпораціям. Залишається щиро сподіватися на те, що в багатьох корпораціях працюють порядні люди, які поважають довіру користувачів. Але в корпораціях можуть бути люди, які розпоряджаються довіреною їм інформацією переслідуючи свої наміри.

Проблема друга. Сьогодні у всіх країнах урядові служби на законодавчому рівні контролюють всі інформаційні потоки. У кожного інтернет-провайдера встановлено обладнання спецслужб, яке повністю відстежує весь трафік. Вважається що уряди завжди діють чесно. Але серед державних службовців можуть бути люди, які переслідують інтереси певних груп.

Проблема третя. Центральні сервера в мережі схильні до ризику вийти з ладу або бути знищеними під час зовнішнього вторгнення. Випадки, коли сервера виходять з ладу (або їх виводять з ладу) і системи перестають працювати траплялися неодноразово. Крім того, деякі власники просто зупинили свої сервера, іноді навіть не повідомивши користувачів. У цих випадках користувачі більше не можуть користуватися зупиненими сервісами і часто втрачають свою інформацію. Централізовані сервіси фізично уразливі і підпорядковані волі певних осіб. Це три основні проблеми, але далеко не єдині.

На рисунку 1 зображений приклад централізованої соціальної мережі, який показує структуру мереж.

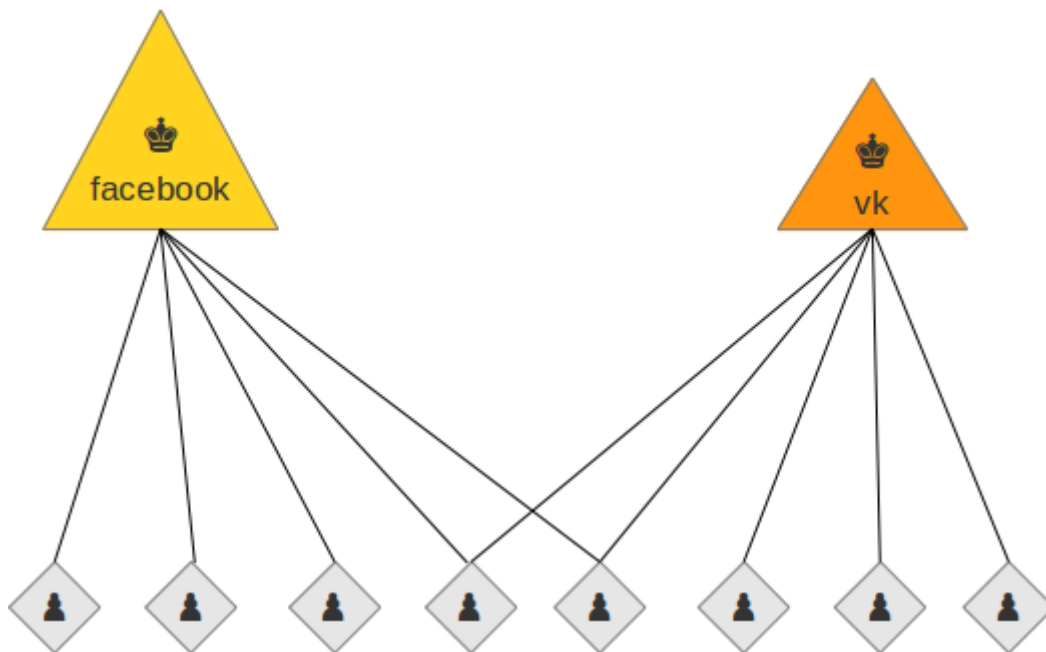


Рисунок 1 - Централізована соціальна мережа

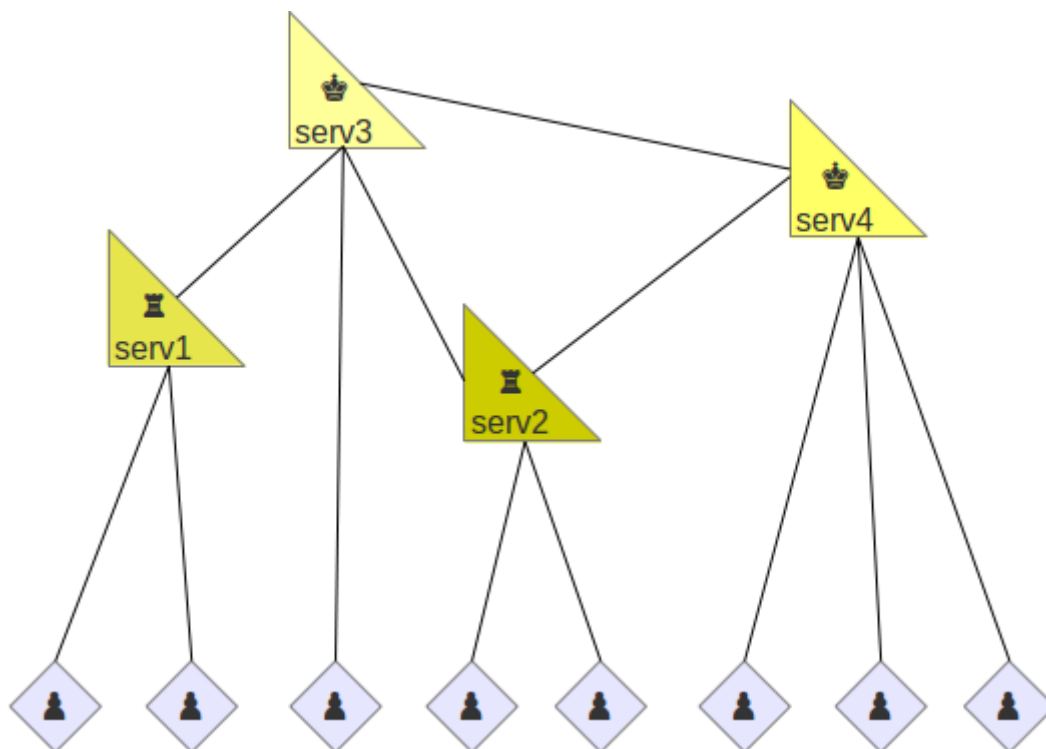


Рисунок 2 - Частково децентралізована мережа Diaspora.

У мережі частково децентралізованого типу (рис. 2) є декілька центрів, де користувачі мають змогу вибрати сервер, якому вони довіряють. Дані все ще зберігаються віддалено на сервері. Користувачі повністю довіряють інформацію власнику. При цьому деякі сервери розміщені у хостерів, з усіма наслідками, що випливають [1]. Але навіть якщо сервер належить приватній особі користувач знову повинен тренувати силу своєї довіри.

Такі мережі менш наражаються на відмову в обслуговуванні, масового витоку даних при зломі, актів вандалізму і анонімної безвідповідальності.

Дані соціальних мереж - невичерпне джерело для досліджень. Використовуючи Facebook API і мову Python розібрано декілька варіантів, які допоможуть дізнатися:

- як звертатися до Facebook API з мови Python за допомогою стандартних бібліотек, зокрема, отримувати список друзів і членів груп;
- деякі можливості програми Gephi.

Побудова егоцентричного графу друзів, видаливши себе самого.

```
import requests
import networkx
import time
import collections
def get_friends_ids(user_id):
    friends_url = 'https://api.fb.com/method/friends.get?user_id={}'
    json_response = requests.get(friends_url.format(user_id)).json()
    if json_response.get('error'):
        print json_response.get('error')
    return list()
    return json_response[u'response']
graph = {}
friend_ids = get_friends_ids(1405906)
for friend_id in friend_ids:
    print 'Processing id: ', friend_id
    graph[friend_id] = get_friends_ids(friend_id)
g = networkx.Graph(directed=False)
for i in graph:
    g.add_node(i)
for j in graph[i]:
    if i != j and i in friend_ids and j in friend_ids:
        g.add_edge(i, j)
pos=networkx.graphviz_layout(g,prog="neato")
networkx.draw(g, pos, node_size=30, with_labels=False, width=0.2)
```

Результат роботи коду можна подивитись на рисунку 3.

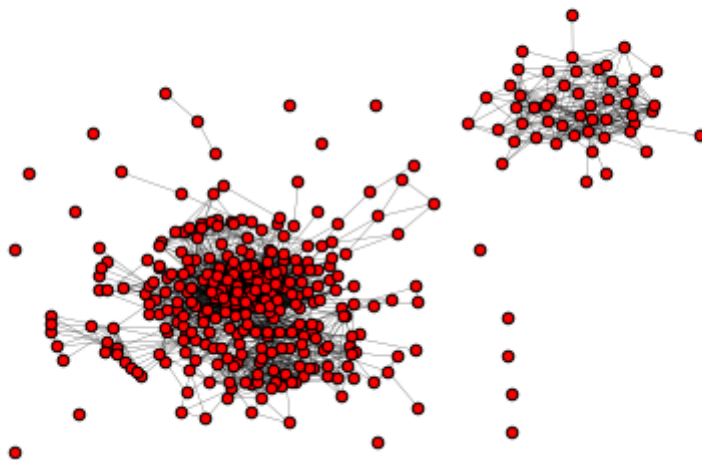


Рисунок 3 - Граф друзів

Можна виділити 2 великі компоненти пов'язаності: друзів з двох різних міст проживання.

Деякі користувачі можуть відкрити ту чи іншу інформацію тільки для зареєстрованих користувачів або для друзів, тому частина методів іноді можуть вимагати авторизації (передачі access токена). У таких випадках є обмеження у вигляді лімітів на API. У документації FB зазначено, що для клієнтського додатка ліміт - 3grps, а для серверного додатка прогресивна шкала в залежності від числа установок додатка (grps / число установок): 5 / <10000, 8 / <100000, 20 / <1000000. 35 / > 1000000 [2].

Також в документації є наступний абзац:

Крім обмежень на частоту звернень, існують і кількісні обмеження на виклик однотипних методів. Зі зрозумілих причин, ми не надаємо інформацію про точні ліміти [2].

Так, наприклад, виклик методу пошуку профілю users.search або метод перегляду стіни користувача wall.get при перевищенні деякого ліміту (але при не перевищенні документованих лімітів) починає видавати порожні результати. Ця ситуація може породити помилки: так, наприклад, при пошуку користувачів ви можете порахувати, що з даного пошуковому запиту немає результатів, а насправді вони відсутні [2].

Нижче наведено фрагмент коду, який допоможе враховувати документовані ліміти, наприклад 3 запити в секунду.

```
deq = collections.deque(maxlen=4)
def throttling_request(url):
deq.appendleft(time.time())
if len(deq) == 4:
time.sleep(max(1+deq[3]-deq[0], 0))
```

На цьому ж графі використано програму Gephi [3]. Gephi - це програма з відкритим вихідним кодом для аналізу і візуалізації графів, написана на Java, спочатку розроблена студентами Технологічного університету Компьєня у Франції [3]. Gephi була обрана для участі в Google Summer Code в 2009, 2010, 2011, 2012 і 2013 роках [3].

Для початку збережено граф в формат .graphml - формат опису графів на основі XML.

```
networkx.write_graphml(g, 'graph.graphml')
```

Експортувавши, завантажено в Gephi і отримано приблизно такий результат:

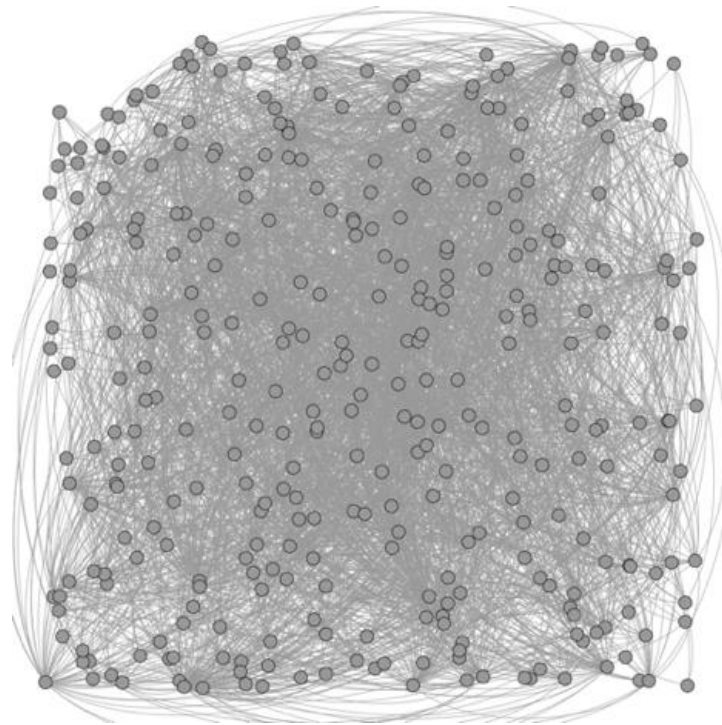


Рисунок 4 – Результат роботи програми Gephi (за [3])

Gerpi має більшу функціональність, яка розширюється за допомогою безлічі плагінів [4]. На рисунках 5 і 6 наведені приклади візуалізації.

Різним кольором відзначені різні значення центральності.

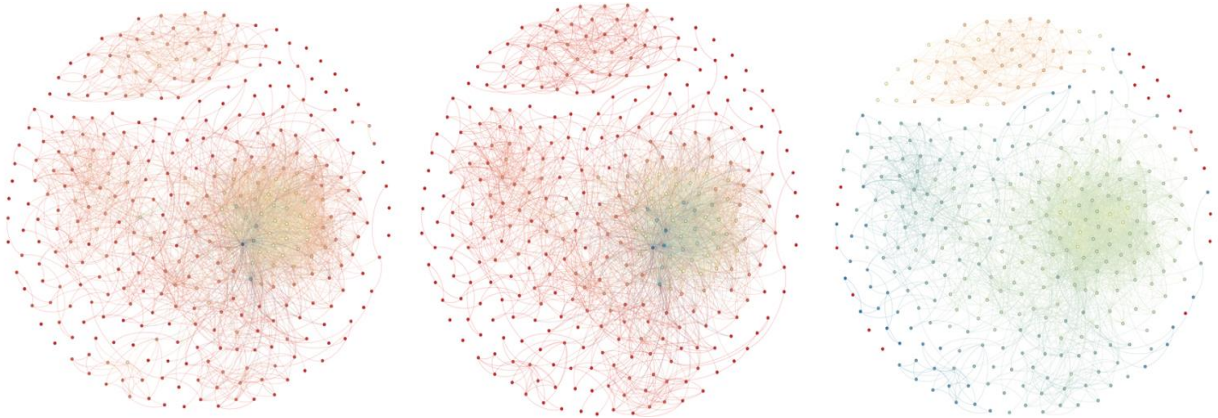


Рисунок 5 - Центральність (PageRank centrality, Degree centrality, Eccentricity centrality) (за [4])

На рисунку 6 наведений візуальний приклад кластеризації, різним кольором відзначені різні класи вибрані алгоритмом.

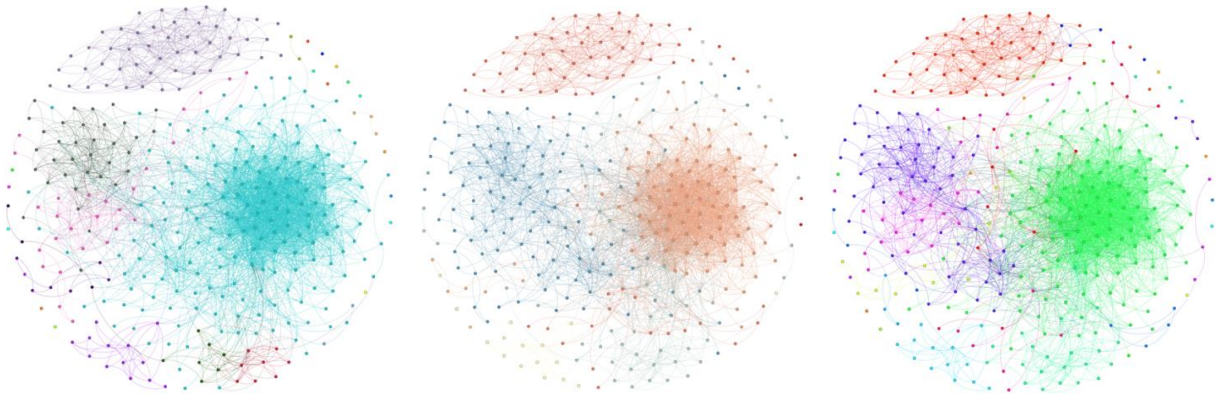


Рисунок 6 - Кластеризація (Modularity clustering, Markov clustering, Chinese Whispers clustering). (за [4])

На основі заздалегідь відомого списку груп соціальної мережі Facebook побудуємо граф:

- вершини - групи соціальної мережі;
- ребра - наявність загальних підписників;
- чим більше у цієї групи підписників, тим більше розмір вершини;
- чим більше у груп загальних підписників, тим ближче один до одного розташовуються вершини.

Як приклад груп розглянуто групи новинних видань.

```
%matplotlib inline
import networkx
import requests
import json
def getFBMembers(group_id, count=1000, offset=0):
    host = 'http://api.fb.com/method'
    if count > 1000:
        set_members_id = set()
        count_members = -1
        offset = 0
    if count_members != response['count']:
        count_members = response['count']
```



```

new_members_id = response['users']
offset += len(new_members_id)
if set_members_id | set(new_members_id) == set_members_id != set(): # without new members
    print 'WARNING: break loop', count_members, len(set_members_id)
    break
set_members_id = set_members_id.union(new_members_id)
return set_members_id
groups = ['http://facebook.com/meduzaproject',
'http://facebook.com/oldlentach',
'http://facebook.com/sport.tvrain',
'http://facebook.com/afishagorod',
'http://facebook.com/1tv',
'http://facebook.com/russiatv']
members = {}
for g in groups:
    name = g.split('http://facebook.com/')[1]
    print name
    members[name] = allCountOffset(getFBMembers, name)
matrix = {}
size = []
max_size = 900
min_size = 100
for node in g.nodes():
    pos=networkx.spring_layout(g)
plt.figure(figsize=(20,20))
networkx.draw_networkx(g, pos, node_size=size, width=0.5, font_size=8)
plt.axis('off')
plt.show()

```

Результатом роботи даного коду буде граф (рис. 7), в якому будуть зображені зв'язки між спільнотами соціальної мережі, які досліджувалися.

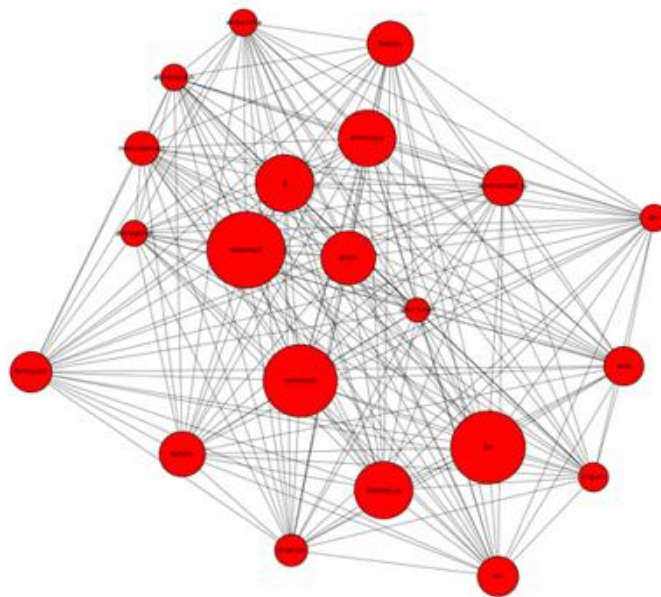


Рисунок 7 – Результат роботи коду

Висновки: Представлені графи демонструють простоту і доступність роботи з соціальними мережами. Дані соціального профілю можуть збагатити дані DMP систем (вік, інтереси, соціальна група): головним завданням є знайти і поставити у відповідність користувачеві DMP-системи його соціальний профіль. З'явилося багато стартапів, які використовують соціальні мережі як джерело для створення резюме: amazinghiring, entelo, profiscope, gild і ін. Головними завданнями є: знайти одного і того ж користувача в різних соціальних мережах і на основі даних, отриманих з соціальних мереж, створити резюме користувача, так як більшість соціальних мереж, крім, хіба що, linkedin, не мають достатньої кількості придатної для резюме інформації.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Шаньгин В.Ф. Защита информации в компьютерных системах и сетях. М.: ДМК Пресс, 2016. 592 с.
2. Документація Facebook [Электронный ресурс] : <https://developers.facebook.com/docs/>
3. Питиляк Д. А., Рожкова А. О. Средства визуализации данных Gephi и Google в экономических исследованиях // Молодой ученый. — 2016. — №12. — С. 1408-1412. — URL <https://moluch.ru/archive/116/31497/> (дата обращения: 28.12.2018)
4. BASTIAN M., HEYMANN S., JACOMY M. Gephi: An Open Source Software for Exploring and Manipulating Networks // Proc. of the 3rd International ICWSM conference, in American Journal of Sociology. - 2009. - P. 361-362
5. HAGBERG A., SWART P., CHULT D.S. Exploring network structure, dynamics, and function using NetworkX // Proceedings of the 7th Python in Science conference, 2008. - P. 11-15.
6. Яремчук Ю.С. Модель процесу передавання управлінської інформації в соціально-економічних мережах / Яремчук Ю.С., Шиян А.А., Заячківський В.М. – Вінниця: ВНТУ, 2014. – 497-504 с.
7. MISLOVE A., MARCON M., GUMMADI K.P., DRU-SCHEL P., BHATTACHARJEE B. Measurement and Analysis of Online Social Networks // Proc. of 7th ACM SIGCOMM Internet Measurement Conference. - 2007. - P. 29-42.

Лемешко Максим Володимирович — студент групи УБ-17мі, факультет менеджменту інформаційної безпеки, Вінницький національний технічний університет, Вінниця, e-mail: MaxKondor227@gmail.com.

Данілов Андрій Олегович — студент групи УБ-17мі, факультет менеджменту інформаційної безпеки, Вінницький національний технічний університет, Вінниця, e-mail: adanilov.955@gmail.com.

Мусій Владислав Сергійович — студент групи УБз-17м, факультет менеджменту інформаційної безпеки, Вінницький національний технічний університет, Вінниця, e-mail: vladmusii24@gmail.com.

Науковий керівник: **Шиян Анатолій Антонович** – к.ф.-м.н, професор кафедри менеджменту та безпеки інформаційних систем, Вінницький національний технічний університет, Вінниця.

Lemeshko Maxim - student group UB17mi, Faculty of Management Information Security, Vinnytsia National Technical University, Vinnytsia, e-mail: MaxKondor227@gmail.com.

Danilov Andrii - student group UB-17mi, Faculty of Management Information Security, Vinnytsia National Technical University, Vinnytsia, e-mail: adanilov.955@gmail.com.

Musii Vladislav - student group UBz17m, Faculty of Management Information Security, Vinnytsia National Technical University, Vinnytsia, e-mail: vladmusii24@gmail.com.

Scientific supervisor: **Shiyan Anatoliy** – Candidate of Physics and Mathematics Sciences, Associate Professor of the Department of Management and Security of Information Systems, Vinnytsia National Technical University, Vinnytsia.