НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Факультет прикладної математики

Кафедра прикладної математики

«До захисту допущено»

Завідувач кафедри

_____ О. Р. Чертов

«___» _____ 2016 р.

Дипломна робота

на здобуття ступеня бакалавра

з напряму підготовки 6.040301 «Прикладна математика»

на тему: «Математичне та програмне забезпечення системи збору та ранжування даних з мережі Інтернет»

Виконав: студент IV курсу, групи КМ-23

Рудник Тарас Петрович		_____
Керівник	доцент Чертов О. Р.	_____
Консультант із	старший викладач	
нормоконтролю	Мальчиков В. В.	_____
Рецензент	доцент, Зорін Ю. М.	_____

Засвідчую, що в цій дипломній роботі немає запозичень із праць інших авторів без відповідних посилань.
Студент _____

Київ — 2016

Національний технічний університет України

«Київський політехнічний інститут»

Факультет прикладної математики

Кафедра прикладної математики

Рівень вищої освіти — перший (бакалаврський)

Напрям підготовки 6.040301 «Прикладна математика»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ О. Р. Чертов

«___» _____ 2016 р.

ЗАВДАННЯ

на дипломну роботу студенту

Руднику Тарасу Петровичу

1. Тема роботи: «Математичне та програмне забезпечення системи збору та ранжування даних з мережі Інтернет»,

керівник роботи Чертов Олег Романович, доцент

затверджені наказом по університету від «06» травня 2016 р. № 1499-С.

2. Термін подання студентом роботи: «15» червня 2016 р.

3. Зміст роботи:

– провести порівняльний аналіз існуючих методів збору та ранжування інформації з мережі Інтернет;

– вибрати технології розробки та математичний метод для розв'язання задачі;

– розробити програмне забезпечення на базі обраних технологій та методу;

– провести випробування розробленої системи.

4. Перелік ілюстративного матеріалу: приклади алгоритмів кластеризації, знімки екранних форм.

5. Дата видачі завдання: «22» лютого 2016 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Терміни виконання етапів роботи	Примітка
1.	Вивчення літератури за тематикою роботи	25.02.2016	
2.	Проведення порівняльного аналізу екстракції значущої інформації з WEB-сайтів	29.02.2016	
3.	Проведення порівняльного аналізу математичних методів аналізу даних	7.03.2016	
4.	Проектування архітектури розроблюваних програмних засобів	18.03.2016	
5.	Визначення складу та форматів вихідних даних та результатів для програми	02.04.2016	
6.	Розробка алгоритмів системи	16.04.2016	
7.	Програмна реалізація системи	22.04.2016	
8.	Збір та ранжування даних з WEB-сайтів	29.04.2016	

№ з/п	Назва етапів виконання дипломної роботи	Терміни виконання етапів роботи	Примітка
9.	Відлагодження і тестування програмного забезпечення	06.05.2016	
10.	Оформлення пояснювальної записки до дипломної роботи	22.05.2016	

Студент _____ Рудник Т. П.

Керівник роботи _____ Чертов О. Р.

АНОТАЦІЯ

Дипломну роботу виконано на 50 аркушах, вона містить 2 додатки та перелік посилань на використані джерела з 20 найменувань. У роботі наведено 8 рисунків та 1 таблицю.

Роботу присвячено розробці математичного та програмного забезпечення системи збору і ранжування даних із мережі Інтернет.

У роботі проведено порівняльний аналіз існуючих рішень для збору даних з мережі Інтернет та методи кластеризації даних. Було виділено їх переваги та недоліки.

На базі фреймворку Scrapy створено систему пошуку та збереження інформації з Web-сайтів новин. Зібрану інформацію проаналізовано за допомогою реалізації на мові програмування Python методу нечіткої кластеризації, що дозволило відфільтрувати та згрупувати новини відповідно до спільних характеристик.

Ключові слова: система збору даних, нечітка кластеризація, алгоритм *c*-середніх, Scrapy, Python.

ABSTRACT

The thesis is presented in 50 pages. It contains 2 appendices and bibliography of 20 references. Eight images and 1 table are given in the thesis.

Diploma work is dedicated to mathematical and programming software development for system of data from Internet network collection and ranging.

In the thesis existing solutions for data collection from Internet network and methods of their clustering were analyzed. Their advantages and disadvantages were distinguished.

Based on Scrapy framework system for information from news web-sites search and storage was created. Collected information was analyzed with the help of Python programming language realization of Fuzzy clustering method which allowed to filter and group news according to their general characteristics.

Key words: data collection system, fuzzy clustering, C-means clustering, Scrapy, Python.

Зміст

ПЕРЕЛІК ОСНОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

API (Application programming interface) — інтерфейс прикладного програмування;

CSV (Comma-Separated Values) — текстовий формат, призначений для представлення табличних даних;

I\O (input\output) — ввід\вивід;

JSON (JavaScript Object Notation) — текстовий формат обміну даними;

NASA (National Aeronautics and Space Administration) — національне управління з аеронавтики і дослідження космічного простору;

PDF (Portable Document Format) — міжплатформний формат електронних документів;

UNIX — сімейство багатозадачних операційних систем;

WEB — інтернет простір;

АТО — Антитерористична операція;

ПЗ — програмне забезпечення;

СКБД — система керування базами даних.

Вступ

Останніми роками спостерігається збільшення «вкидань» неправдивої інформації з метою масової зміни думки населення. Подібні інформаційні атаки є невід'ємною складовою ведення сучасної гібридної війни. У результаті, дезорієнтоване населення підтримує не ту державу чи політичну силу, яка діє в інтересах людей та намагається допомогти, а ту, яка докладає більше зусиль при веденні інформаційної війни.

Інформаційні атаки застосовуються не тільки при веденні війни, але й у бізнесі. Наприклад, у вересні 2008 року масштабна інформаційна кампанія проти «Промінвестбанку» майже довела банк до банкрутства [1]. Не можна стверджувати, що лише інформаційна атака через Інтернет привела банк до складного стану, однак саме ці повідомлення підірвали довіру багатьох вкладників, що змусило їх масово забирати свої заощадження з банку. Лише 5 млрд. грн рефінансування, виділених Національним Банком України, допомогли покращити ситуацію, а 5 грудня 2008 р. у «Промінвестбанку» з'явився новий власник.

Дипломну роботу присвячено розробці системи збору даних з мережі Інтернет та використанню математичних методів аналізу зібраної інформації. Для ранжування даних використовується метод нечіткої кластеризації *c*-середніх, що дозволяє оцінювати імовірність належності статті кожному з кластерів.

1 ПОСТАНОВКА ЗАДАЧІ

Метою даної роботи є створення математичного та програмного забезпечення для збору і ранжування даних з мережі Інтернет, щоб вберегти населення від поширення неправдивої інформації та інших наслідків гібридної війни.

Система повинна автоматизувати процес класифікації зібраної інформації без залучення експерта. Вибір математичного апарату повинен найточніше описувати предметну область застосування.

Основні задачі, що підлягають розв'язанню:

а) проведення порівняльного аналізу існуючих методів збору та ранжування інформації з мережі Інтернет;

б) вибір технологій розробки та математичного методу для розв'язання задачі;

в) розробка програмного забезпечення на базі обраних технологій та методу;

г) проведення випробувань розробленої системи.

Реалізована система повинна:

а) зберігати інформацію з веб-сайту новин tsn.ua в базу даних;

б) самостійно переходити за посиланнями на сайті;

в) робити запити до веб-сайту асинхронно;

г) бути спроможною класифікувати отриману інформацію у відносно однорідні групи.

Програмне забезпечення має виконуватись в операційній системі Ubuntu 14.04 на IBM-сумісному комп'ютері, до складу якого входять:

– процесор із тактовою частотою, не менше 1,3 ГГц;
– оперативна пам'ять обсягом, не менше за 512 МБ.

2 ОГЛЯД ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ

2.1 Огляд існуючих рішень для збору інформації з мережі Інтернет

На сьогоднішній день є багато готових рішень для збору інформації з мережі Інтернет: Import.IO, Kimono, Apache Nutch тощо. Розглянемо детальніше вказані застосунки.

2.1.1 Import.IO

Import.IO — веб-орієнтована платформа для отримання даних з веб-сайтів без написання коду. Дозволяє розробляти власні додатки та інтерфейси до них. [2]

Користувачі заходять на сайти та виділяють приклади даних, які їх цікавлять. З часом алгоритм узагальнює отриману інформацію з вказаних прикладів та вчиться отримувати потрібні дані з веб-сторінок. Зібрана інформація зберігається на серверах Import.IO та може бути завантажена в форматах CSV чи JSON. Платформа дозволяє отримувати інформацію зі ста джерел.

Засоби збору інформації платформи доступні для всіх популярних операційних систем (Mac OS, Linux, Windows тощо). Окрім цього, розробники створили власний блог — місце де користувачі обмінюються досвідом. Також існує багато навчальних матеріалів для початківців.

Переваги застосування для поставленої задачі:

– простота використання;
– збір інформації з великої кількості джерел;
– можливість використання на багатьох операційних системах.

Недоліки застосування для поставленої задачі:

– відсутність можливості збереження інформації в базу даних;
– залежність від зовнішнього ресурсу, що може призвести до втрати дієздатності системи;
– значні затрати часу на виділення інформації.

2.1.2 Kimono

Kimono — платформа, яка дозволяє користувачам «перетворити» будь-який веб-сайт в активний застосунок без написання коду [3]. Для того, щоб використовувати платформу, необхідно встановити розширення в браузері. Перейшовши на веб-сайт, з якого потрібно отримати інформацію, достатньо активувати розширення і виділити необхідні дані. Платформа самостійно збереже дані на власному сервері у зручному для перегляду форматі.

Kimono не дуже зручна у використанні і вимагає деякого проміжку часу, щоб розібратись з її інтерфейсом, але у мережі Інтернет є багато відеоматеріалів з детальними поясненнями аспектів роботи з платформою.

Переваги застосування для поставленої задачі:

– простота використання;

– можливість збереження даних у зручних форматах.

Недоліки застосування для поставленої задачі:

– залежність від зовнішнього ресурсу;
– значні затрати часу на розбір аспектів роботи з системою;
– необхідність розбирати кожен веб-сайт окремо.

2.1.3 Apache Nutch

Apache Nutch — програмне забезпечення з відкритим кодом для збору даних з мережі Інтернет [4], яке швидко розширюється та масштабується. Модульна архітектура дозволяє користувачам створювати додатки для синтаксичного аналізу, пошуку та кластеризації даних. Apache Nutch розроблений на мові програмування Java і дозволяє користувачам змінювати функціональну частину відповідно до власних потреб.

Переваги застосування для поставленої задачі:

– можливість збереження інформації в базу даних;
– простота у використанні та модифікації (для програмістів);
– незалежність від зовнішніх ресурсів.

Недоліки застосування для поставленої задачі:

– тривалий час збору інформації;
– переривання під час роботи.

2.2 Огляд технологій розробки

У даному розділі описано мови програмування і фреймворки для створення веб-павуків та бази даних, які можуть бути використані для розв'язання поставленої задачі, виділено їх переваги та недоліки.

2.2.1 Огляд мов програмування

Perl — високорівнева інтерпретована динамічна мова програмування загального призначення [5]. Вона розроблена у 1987 році Ларі Воллом, лінгвістом і програмістом за освітою, який на той час працював системним адміністратором у NASA, як скриптова мова для UNIX, метою якої було полегшити процес обробки текстів журналів. З того часу до мови було внесено багато змін і здійснено перегляд її концепції та архітектури, в результаті чого вона стала дуже популярною серед програмістів. Ларі Волл продовжує працювати над ядром мови і наразі очікується вихід нової версії.

Perl надає потужні можливості для обробки тексту без довільних обмежень на довжину даних багатьох сучасних інструментів Unix, полегшує маніпуляції з текстовими файлами. Застосовується для програмування графіки, системного адміністрування, у мережному програмуванні, для програмного забезпечення, яке взаємодіє з базами даних та збору інформації з мережі інтернет.

Переваги використання для поставленої задачі:

– швидка розробка програми за рахунок інтерпретовності мови програмування;
– добре працює під різними операційними системами;
– наявність великої кількості бібліотек з готовими рішеннями.

Недоліки використання для поставленої задачі:

– написані програми працюють повільно;
– невелика спільнота користувачів.

PHP — скриптова мова програмування, була створена для генерації HTML-сторінок на стороні веб-сервера. Вона є однією з найпоширеніших мов, що використовуються у сфері веб-розробок (разом із Java, .NET, Perl, Python, Ruby). PHP підтримується переважною більшістю хостинг-провайдерів та є проектом відкритого програмного забезпечення.

PHP інтерпретується веб-сервером у HTML-код, який передається на сторону клієнта. На відміну від скриптової мови JavaScript, користувач не бачить PHP-коду, бо браузер отримує готовий html-код. Це є перевага з точки зору безпеки, але погіршує інтерактивність сторінок. Але ніщо не забороняє використовувати PHP для генерування і JavaScript-кодів які виконуються вже на стороні клієнта.

Переваги використання для поставленої задачі:

– наявність інтерфейсів для багатьох баз даних;
– основні функції вбудовано в інтерпретатор, нема необхідності підключати багато модулів;
– велика спільнота користувачів.

Недоліки використання для поставленої задачі:

– незручний дизайн мови;
– не підтримується юнікод в версіях до 6.0;
– непередбачуваність нових версій PHP;
– написані програми працюють повільно.

Java — об’єктно орієнтована мова програмування [7], випущена компанією Sun Microsystems у 1995 році як основний компонент платформи Java. Передусім Java розроблялась як платформо-незалежна мова, тому вона має менше низькорівневих можливостей для роботи з апаратним забезпеченням. За необхідності таких дій java дозволяє викликати підпрограми, написані іншими мовами програмування.

Переваги використання для поставленої задачі:

– повторне використання коду;
– безпечність написаних програм;
– швидка передача інформації через Java I\O API;
– велика спільнота користувачів.

Недоліки використання для поставленої задачі:

– велике навантаження на оперативну пам’ять обладнання;
– існуючі рішення для написання веб-павуків не готові до використання.

Javascript — динамічна [8], об'єктно орієнтована мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується як частина

браузера, що надає можливість коду на стороні клієнта (такому, що виконується на пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки. Мова JavaScript також використовується для програмування на стороні сервера (подібно до таких мов програмування, як Java і C#), розробки ігор, стаціонарних та мобільних додатків, сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite), всередині PDF-документів тощо.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією.

Окрім прототипної, JavaScript також підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

Переваги використання для поставленої задачі:

– можливість писати асинхронні програми;
– велика спільнота користувачів.

Недоліки використання для поставленої задачі:

– важка система наслідування;
– погана «читабельність» коду;
– існуючі рішення для написання веб-павуків не готові до використання.

Python — інтерпретована [9] об'єктно-орієнтована мова програмування високого рівня з динамічною семантикою. Розроблена в 1990 році Гвідо ван Россумом. Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднання існуючих компонентів. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій так і у вихідній формі на всіх основних платформах. В мові програмування Python підтримується декілька парадигм програмування, зокрема: об'єктно-орієнтована, процедурна, функціональна та аспектно-орієнтована.

Переваги використання для поставленої задачі:

– дозволяє швидше розробляти програми за рахунок інтерпретовності мови програмування та структур даних високого рівня;
– динамічна семантика;
– наявність великої кількості бібліотек з готовими рішеннями;
– якісна документація та велика спільнота користувачів.

Недоліки використання для поставленої задачі:

– низька швидкодія розроблених програм;
– глобальне блокування інтерпретатора.

2.3 Огляд реалізацій веб-павуків

Python має декілька бібліотек для синтаксичного аналізу сайтів та збору інформації з них: lxml, BoutiulSoup, pyquery, re. Вони виконують задачу парсингу сайту, проте не дають можливості написання Web-павука для збору інформації з декількох сайтів. Також є фреймворки для збору інформації з мережі Інтернет: Grab і Scrapy. Grab написаний однією людиною і нею підтримується, в той час як Scrapy – це великий проект з відкритим кодом та хорошою документацією [10], який підходить для вирішення поставленої задачі.

Переваги застосування Scrapy для поставленої задачі:

– асинхронне виконання запитів;
– вбудовані інструменти для синтаксичного розбору сторінки;
– веб-павук переходить по посиланнях;
– підтримка багатьох баз даних;
– якісна документація.

Недоліки застосування для поставленої задачі:

– нема стабільної реалізації на Python 3;
– погана робота з не-ascii символами.

2.4 Огляд СКБД

Oracle — об'єктно реляційна система керування базами даних [11], яка має великі можливості для зберігання даних, їх захисту, аналітики та реорганізації.

Переваги застосування для поставленої задачі:

– надійність та захищеність системи;
– можливість використання на більшості популярних платформ.

Недоліки використання для поставленої задачі:

– платна ліцензія та підтримка.

MySQL — вільна система керування реляційними базами даних [12].

Вона була розроблений компанією «TcX» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL — одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

Переваги застосування для поставленої задачі:

– простота адміністрування;
– безкоштовність.

Недоліки застосування для поставленої задачі:

– проблеми з реплікацією та транзакціями;
– не підтримуються стандарти SQL.

PostgreSQL — об'єктно-реляційна [13] система керування базами даних (СКБД). Є альтернативою як комерційним СКБД (Oracle Database, Microsoft SQL Server, IBM DB2 та інші), так і СКБД з відкритим кодом (MySQL, Firebird, SQLite).

Порівняно з іншими проектами з відкритим кодом, такими як Apache, FreeBSD або MySQL, PostgreSQL не контролюється якоюсь однією компанією, її розробка можлива завдяки співпраці багатьох людей та компаній, які хочуть використовувати цю СКБД та впроваджувати у неї найновіші досягнення.

Переваги застосування для поставленої задачі:

– зручний механізм транзакцій та реплікації;
– висока швидкість виконання запитів;
– якісна документація;
– можливості повнотекстового пошуку та аналізу даних;
– відповідність стандартам SQL-92, SQL-98, SQL-2003, SQL-2011;
– безкоштовність.

SQLite — полегшена реляційна [14] система керування базами даних, втілена у вигляді бібліотеки, де реалізовано багато зі стандарту SQL-92. Сирцевий код SQLite поширюється як суспільне надбання (англ. public domain), тобто може використовуватися без обмежень та безоплатно з будь-якою метою.

Переваги застосування для поставленої задачі:

– надається інтерфейс для більшості сучасних мов програмування високого рівня;
– швидко працює;

– проста у використанні;
– безкоштовна.

Недоліки застосування для поставленої задачі:

– недостатньо надійна та захищена від зовнішніх атак;
– погано працює з великими транзакціями;
– не пристосована для зберігання великих об’ємів даних.

MongoDB — документо-орієнтована [15] система керування базами даних (СКБД) з відкритим сирцевим кодом, яка не потребує опису схеми таблиць. MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД, функціональними і зручними у формуванні запитів.

Переваги використання для поставленої задачі:

– динамічні запити;
– документо-орієнтоване сховище;
– хороша масштабованість.

Недоліки використання для поставленої задачі:

– необхідно багато пам’яті;
– однопоточність;
– збереження надлишкових даних.

2.5 Вибір технології розробки

Було проаналізовано існуючі рішення для реалізації веб-павука, мови програмування для розробки та бази даних для збереження інформації.

У якості мови програмування було обрано Python через те, що вона надає можливості швидко розробляти веб павуків з використанням фреймворку Scrapy та надає можливості аналізу даних.

Так як для системи важлива швидкодія та цілісність інформації, в якості СКБД було обрано PostgreSQL, додатковою перевагою якої є можливості повнотекстового пошуку.

2.6 Висновки до розділу

Було проаналізовано існуючі рішення для збору інформації з мережі Інтернет та виявлено їх основні недоліки: повільна робота та неможливість збереження в базу даних отриману інформацію, а також залежність від зовнішніх ресурсів, що може призвести до втрати дієздатності системи.

В якості основної мови програмування було обрано — Python. Для написання веб-павуків використовувався фреймворк Scrapy. У якості СКБД було обрано PostgreSQL.

3 ОГЛЯД МАТЕМАТИЧНИХ МЕТОДІВ

В даному розділі описуються методи та алгоритми кластерного аналізу.

Для класифікації зібраної інформації було обрано кластерний аналіз, оскільки він дозволяє розглядати достатньо великі об’єми інформації та робити її компактною і наглядною. З його допомогою можна провести класифікацію будь-яких об’єктів, які характеризуються рядом ознак. Кластерний аналіз надає ряд переваг для розв’язання поставленої задачі:

– отримані кластери можна інтерпретувати, тобто описувати, які групи існують;

– можна відкидати окремі кластери, в які потрапили не цікаві для користувачів дані.

3.1 Кластеризація даних

Кластеризація даних [17] є процесом розподілу елементів на класи або групи так, що елементи в одному класі є якомога ближчими, а елементи різних класів є настільки різнорідними, наскільки це можливо. Залежно від характеру даних та мети кластеризації можуть використовуватися різні міри подібності для розміщення елементів в класах, причому вони визначають самі кластери. Приклади мір, які можуть бути використані для кластеризації, включають відстань, зв'язок та інтенсивність.

У жорсткій кластеризації дані розділені на окремі кластери, де кожен елемент належить одному кластеру. В нечіткій кластеризації елементи даних можуть належати до більш ніж одного тематичного напряму і з кожним елементом множини пов'язана функція належності до кожного кластеру. Вона вказує на силу зв'язку між елементом даних і конкретною групою. Нечітка кластеризація є процесом присвоєння мір належності та їх використання для визначення складу кожного з кластерів.

3.2 Огляд алгоритмів кластеризації

3.2.1 Алгоритми ієрархічної кластеризації

Серед алгоритмів ієрархічної кластеризації [18] виділяються два основних типи: висхідні і низхідні. Низхідні алгоритми працюють за принципом «зверху до низу»: спочатку всі об’єкти розміщуються в один кластер, який потім розбивається на менші кластери. Висхідні алгоритми спочатку розміщують кожен об’єкт в окремий кластер, а потім об’єднують кластери в більш крупні, поки всі об’єкти вибірки не будуть знаходитись в одному кластері. Таким чином будується система вкладених розбиттів.

Для обчислення відстані між об’єктами користуються: одиночним або повним зв’язком.

Результатом роботи ієрархічних алгоритмів є дендрограма. Найпопулярнішими алгоритмами, які будують розбиття «знизу догори» є:

– single-link — даний алгоритм на кожному кроці об'єднує кластери з найменшою відстанню між двома представниками. На рисунку 3.1 зображено приклад роботи даного алгоритму;

– complete-link — на кожному кроці об'єднує два кластери з найменшою відстанню між двома найбільш віддаленими представниками. На рисунку 3.2 зображено приклад роботи даного алгоритму.

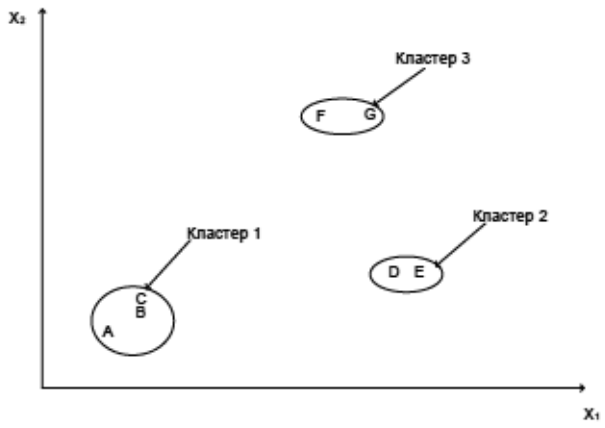


Рисунок 3.1 — Приклад роботи single-link алгоритму

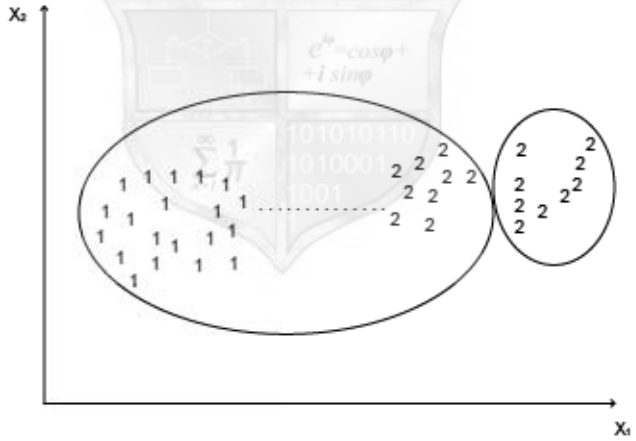


Рисунок 3.2 — Приклад роботи complete-link алгоритму

Під час ієрархічної кластеризації відбувається послідовний поділ великих кластерів на менші або об’єднання менших кластерів у великі.

Алгомеративні методи послідовно об’єднують вихідні елементи, в результаті чого зменшується кількість кластерів. На початку роботи алгоритму всі об’єкти є окремими кластерами. Спочатку найбільш схожі об’єкти об’єднуються в один кластер. Об'єднання продовжується до тих пір,

доки всі об’єкти не будуть складати один кластер.

Алгоритм CURE (Clustering Using Representatives) здійснює ієрархічну кластеризацію і для визначення об’єкта у кластер використовує набір визначаючих точок.

Дивизимні методи послідовно розділяють вихідний кластер, який складається з усіх об’єктів, збільшуючи кількість кластерів. Спочатку всі об’єкти належать до одного кластеру, який на подальших кроках ділиться на менші кластери, в результаті чого утворюється послідовність груп розщеплення.

Алгоритм BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) здійснює двоетапний процес кластеризації. При цьому будується кластерне дерево з заданими коефіцієнтами розгалуження та пороговою величиною. Кожен листовий вузол має посилання на два сусідні вузли. Кластер, який складається з двох елементів листового вузла, повинен задовольняти наступній умові: діаметр або радіус отриманого кластера повинен бути не більше порогової величини T.

Дерево мінімального покриття проводить ієрархічну кластеризацію «зверху донизу». Спочатку всі об'єкти розміщені в одному кластері, на кожному кроці один з кластерів розділяється на два, так щоб відстань між ними була максимальною. На рисунку 3.3 наведено приклад алгоритму дерева мінімального покриття.

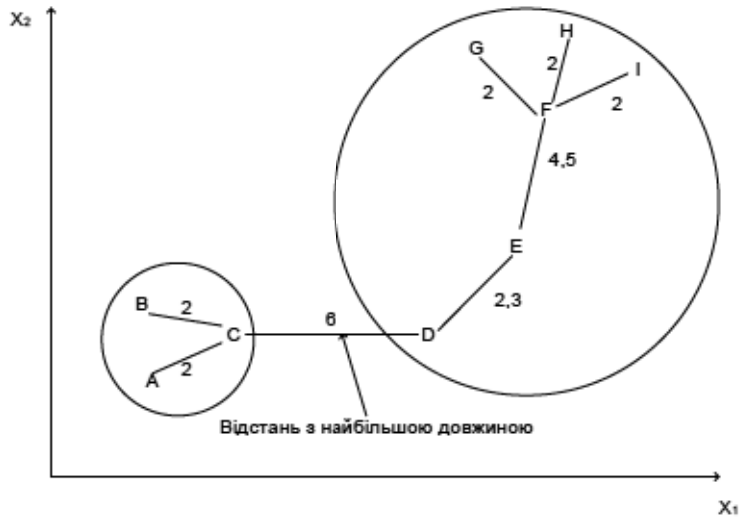


Рисунок 3.3 – Приклад алгоритму дерева мінімального покриття

3.2.2 Метод найближчого сусіда

Даний метод кластеризації вважається одним з найстаріших [18]. Він був придуманий в 1978 році, є простим та найменш оптимальним з усіх представлених.

Для всіх об’єктів, які не належать жодному кластеру роблять наступні кроки:

а) Знаходять найближчого сусіда, кластер якого визначений.

б) Якщо відстань до цього сусіда менша за вказану межу, то відносять його в той самий кластер. Інакше з об'єкта, який розглядається, створюється ще один кластер. Після цього розглядається результат і при необхідності збільшується межа.

3.2.3 Алгоритми, які базуються на теорії графів

В даних алгоритмах [19] вибірка об’єктів представляється в вигляді графа, вершинам якого відповідають об’єкти, а ребра мають вагу, яка дорівнює відстані між об’єктами. Перевагою алгоритмів кластеризації, які базуються на теорії графів, є наочність, відносна простота реалізації та можливість внесення різноманітних вдосконалень, які засновані на геометричних міркуваннях.

3.2.4 Алгоритм *k*-середніх

Даний алгоритм [20] складається з наступних кроків:

а) випадковим чином вибрати *k* точок, які є початковими координатами «центрами мас» кластерів (будь-які *k* з *n* об'єктів, або взагалі *k* випадкових точок);

б) віднести кожен об'єкт до кластеру з найближчим «центром мас»;

в) перерахувати «центри мас» кластерів згідно поточного членства;

г) якщо критерій зупинки не задоволений, повернутися до кроку *б*.

Критерієм зупинки вибирається один з двох варіантів: відсутність переходу об'єктів з кластера в кластер на кроці 2 або мінімальна зміна средньоквадратичної похибки. Алгоритм чутливий до початкового вибору центру мас. На рисунку 3.4 наведено приклад алгоритму *k*-середніх.

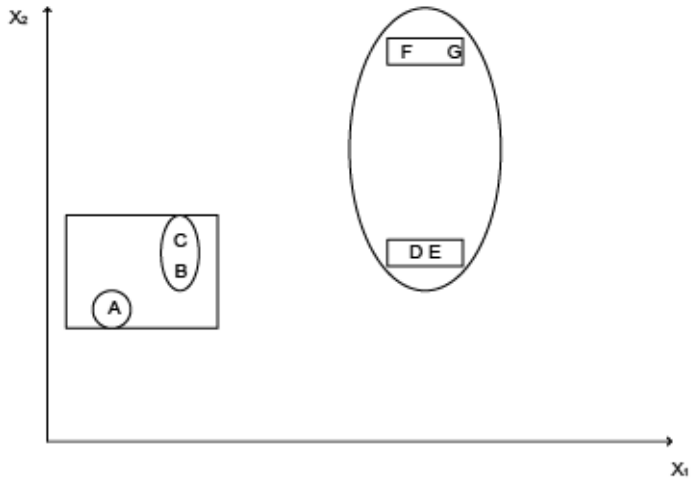


Рисунок 3.4 – Приклад алгоритму *k*-середніх

Існують алгоритми, похідні від *k*-середніх, робота яких суттєво не відрізняється. Прикладом може слугувати алгоритм PAM (partitioning around medoids), що відрізняється від алгоритму *k*-середніх тим, що при його роботі перерозподіляються об'єкти відносно медіани кластера, а не його центра.

3.2.5 Алгоритм нечітких *c*-середніх

Нечітка кластеризація — це [17] клас алгоритмів кластерного аналізу, в яких розподіл точок даних для кластеризації є не «чітким» («0 або 1», «так або ні»), а «нечітким» (в тому ж значенні, що й у нечіткій логіці).

При необхідності нечіткої кластеризації об'єктів [17] найчастіше використовують алгоритм *c*-середніх.

Вхідною інформацією для даного алгоритму є матриця спостережень X розмірності $l \times n$:

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \dots & \vdots \\ x_{l1} & x_{l2} & \dots & x_{ln} \end{bmatrix}$$

де l — число об'єктів, n — число ознак для кожного об'єкта. [5]

Задача кластеризації полягає в розбитті множини об'єктів на групи (кластери) «схожих» між собою об'єктів. В n-вимірному метричному просторі ознак мірою «схожості» двох об'єктів будемо вважати відстань між ними.

Метод нечіткої кластеризації дозволяє кожному об'єкту належати з різним степенем до декількох кластерів одночасно. Число кластерів c вважається заздалегідь відомим.

Кластерна структура задається матрицею належності:

$$M = \begin{bmatrix} m_{11} & m_{12} & \dots & m_{1l} \\ m_{21} & m_{22} & \dots & m_{2l} \\ \vdots & \vdots & \dots & \vdots \\ m_{c1} & m_{c2} & \dots & m_{cl} \end{bmatrix}$$

де m_{ij} — степінь належності j-го елемента i-му кластеру.

Відмітимо, що матриця належності повинна задовольняти наступним умовам:

– $m_{ij} \in [0,1], i = \overline{1,c},\ \ j = \overline{1,l}$;

– кожен об'єкт повинен бути розподілений між всіма кластерами;

– $0 < \sum_{j=1}^{l} m_{ij} < l,\ i = \overline{1, c}$, тобто жоден кластер не повинен бути порожнім чи містити всі елементи.

Для оцінки якості розбиття використовується критерій відхилення, який показує суму відстаней від об'єктів до центрів кластерів з відповідними степенями належності:

$$J = \sum_{i=1}^{c} \sum_{j=1}^{l} (m_{ij})^{w} d(v_i, x_j)$$

де $d(v_i, x_j)$ — Евклідова відстань між j-м об'єктом $x_j = (x_{j1}, x_{j2}, \dots, x_{jn})$ та *i-м* центром кластера $v_i = (v_{i1}, v_{i2}, \dots, v_{in})$, $w \in (1, \infty)$ — експоненційна вага, яка визначає нечіткість кластерів,

$$V = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ v_{c1} & v_{c2} & \cdots & v_{cn} \end{bmatrix}$$ — матриця координат центрів

кластерів, елементи якої обчислюються за формулою:

$$v_{ik} = \frac{\sum_{j=1}^{l} (m_{ij})^{w} x_{jk}}{\sum_{j=1}^{l} (m_{ij})^{w}}, k = \overline{1, n(v)}.$$

Задачею є знаходження матриці *M,* яка мінімізує критерій *J.* Для цього використовується алгоритм нечітких *c*-середніх, в основі якого лежить метод множників Лагранжа. Він дозволяє знайти локальний оптимум, тому при різних запусках може вийти різний результат.

На першому кроці матриця належності M генерується випадковим чином. Далі запускається ітерацій процес обчислення центрів кластерів і перерахунку елементів матриці степенів належності:

$$m_{ij} = \frac{1}{(dij)^{\frac{2}{w-1}}} \sum_{k=1}^{c} \frac{1}{(d_{kj})^{\frac{2}{w-1}}} \text{ при } d_{ij} > 0 \text{ та}$$

$$m_{kj} = \begin{cases} 1, k = i \\ 0, \ \ k \neq i \end{cases}, \text{ при } d_{ij} = 0$$

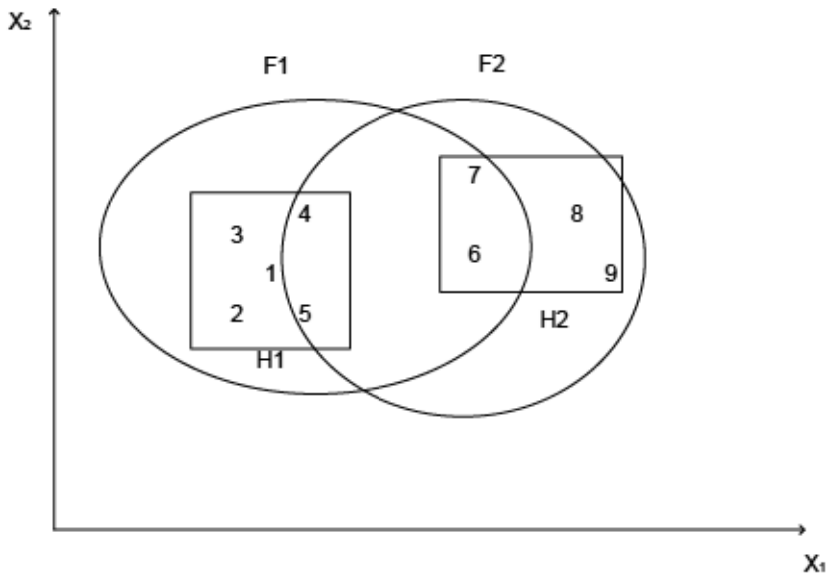
де $d_{ij} = d\left(v_i, x_j\right),\ i = \overline{1,c}, j = \overline{1,l}.$

Умова завершення обчислень:

$$\|M - M^*\|^2 < \ \varepsilon$$

де M^* — матриця на попередній ітерації, а ε заздалегідь заданий параметр завершення.

На рисунку 3.5 наведено приклад алгоритму нечітких c-середніх.



F1={(1;0,9), (2;0,8), (3;0,7), (4;0,6), (5;0,55); (6;0,2), (7;0,2), (8;0,0), (9;0,0)}
F2={(1;0,0), (2;0,0), (3;0,0), (4;0,1), (5;0,15); (6;0,4), (7;0,35), (8;1,0), (9;0,9)}

Рисунок 3.5 – Приклад алгоритму нечітких *c*-середніх

3.3 Вибір та обґрунтування вибору рішення

Для реалізації системи було обрано метод нечіткої кластеризації *c*-середніх, який надає можливість робити висновки про імовірність відношення кожної статті до певного кластеру та дозволяє кожному об’єкту належати до декількох кластерів одночасно. Ієрархічні алгоритми не задовольняють вимогам, висунутим в розділі 1, оскільки вони спираються на «гіпотезу компактності»: всі близькі об’єкти ставляться до одного кластеру, а далекі об’єкти знаходяться в різних кластерах, що може призвести до великої похибки при кластеризації даних. Метод *k*-середніх вимагає задання користувачем кількості кластерів та еталонів, що не завжди можна зробити раціонально. Окрім цього алгоритм потребує велику кількість розрахунків, пов'язаних з обчисленням та стабілізацією центрів

кластерів та перерахунків, якщо заданої кількості ітерацій недостатньо для точного визначення результатів. Алгоритм найближчого сусіда швидко виконується, але є «жадібним», тому може видавати неоптимальні результати.

3.4 Висновки до розділу

У даному розділі обрано та розглянуто математичне забезпечення для кластеризації об'єктів. Під час вибору математичного забезпечення було враховано вимоги до системи та нечітку природу зібраних даних. Обраний метод надає можливості для класифікації об'єктів і наглядного представлення отриманих результатів. При розгляді статей, зібраних з сайтів новин, потрібно враховувати, на скільки відсотків кожна стаття належить певному кластеру, щоб зрозуміти важливість інформації та кількість статей, пов'язаних з певною тематикою.

4 СТРУКТУРА І ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Структура автоматизованої системи

Розроблена автоматизована система складається з таких підсистем:

– підсистема збору даних з мережі Інтернет;
– підсистема розподілу зібраної інформації по датах появи;
– підсистема нечіткої кластеризації об’єктів.

Вхідними даними для системи є ключові слова для кластеризації статей відповідно до тематики, яка цікавить користувача системи, а також періоди появи статей, для того, щоб досліджувати, як змінювалась інформація протягом певного проміжку часу. Дані вносяться у відповідні текстові файли.

На виході система видає графіки нечіткої кластеризації статей за вказані проміжки часу. Відповідно до вимог, висунутих в розділі 1, система забезпечує:

– збереження інформації з веб-сайтів новин в базу даних;
– самостійно переходить за посиланнями на сайті;
– робить запити до веб-сайтів асинхронно;
– класифікує отриману інформацію у відносно однорідні групи.

4.1.1 Підсистема збору даних з мережі Інтернет

Підсистема збору даних з мережі Інтернет забезпечує пошук та збереження інформації з сайтів новин в базу даних системи для подальшої класифікації та аналізу. Відбувається збір таких атрибутів кожної статті:

– заголовок;
– посилання;
– дата появи;
– повний текст статті.

Вхідними даними для даної підсистеми є лише посилання на початкову сторінку пошуку, яке задається в файлі налаштувань системи. Також в даному файлі користувач може заборонити системі переходити на певні сайти, обмежити кількість переходів та об'єм збору інформації, налаштувати затримку між запитами, щоб не відбулось блокування IP адреси електронно-обчислюваної машини, на якій проводиться збір даних, за велику кількість безперервних запитів до веб-сайту. При потребі користувач може змінити налаштування використовуваної бази даних чи задати збереження інформації в файл формату JSON, CSV чи XML без використання бази даних.

Даний етап є одним з основних етапів в роботі системи, оскільки він забезпечує збір інформації, необхідної для подальшої класифікації та аналізу. Виділяють наступні кроки в роботі даної системи:

– зчитування налаштувань бази даних системи;
– зчитування початкового посилання для збору інформації та списку заборонених посилань;
– збір та збереження інформації з веб-сайту;

– перехід за посиланнями, які зустрічаються на сторінці веб сайту;

– запис в лог-файл інформації про роботу системи;

– вивід користувачу інформації про кількість зібраної інформації та переходів по посиланнях, а також помилки, які виникли під час роботи системи.

4.1.2 Підсистема розподілу зібраної інформації за датами появи

Дана підсистема забезпечує розподіл інформації, яка знаходиться в базі даних, відповідно до вимог користувача, щоб забезпечити подальшу кластеризацію та аналіз отриманих даних за певні проміжки часу, зрозуміти, яким чином змінювалась кількість новин протягом вказаних проміжків часу.

Вхідними значеннями для даної підсистеми є списки з датами, які цікавлять користувача. При цьому клієнт не обмежений вибором, тобто в одному списку може вказати лише один день, а в інших тиждень, місяць, рік тощо.

Даний етап є підготовчим в роботі системи. Основна його задача — сформулювати проміжки часу для розподілу статей по них та окремої класифікації кожного набору інформації. Виділяють наступні кроки в роботі даної підсистеми:

– зчитування списків з проміжками часу з текстового файлу;

– вибір інформації з бази даних, та збереження в окремі файли відповідно до списків з проміжками часу.

4.1.3 Підсистема нечіткої кластеризації об’єктів

Дана підсистема забезпечує класифікацію зібраних статей за допомогою методу нечіткої кластеризації *c*-середніх. Відбувається підрахунок кількості ключових слів в кожній статті та знаходиться їх відносна кількість в порівнянні з статтею, з найбільшим показником ключових слів. За отриманими числовими даними відбувається нечітка кластеризація об’єктів, підрахунок центрів кластеризації та побудова графіку з результатами. Якщо в статті не було знайдено жодного з ключових слів, то така стаття не враховується при подальшому аналізі даних.

Вхідними значеннями для даної підсистеми є файли, які містять ключові слова та тексти статей. Матриця спостережень, яка є необхідною для методу нечітких *c*-середніх, обраховується програмно, без додаткового вводу користувачем.

Виділяють наступні кроки в роботі даної підсистеми:

– ініціалізація списків з ключовими словами;

– зчитування текстів статей;

– підрахунок ключових слів в кожній статті;

– переведення кількості ключових слів в кожній статті у відносну величину, в порівняння з максимумом;

– запис даних в матрицю спостережень;

– нечітка кластеризація даних за отриманою матрицею спостережень;

– побудова графіків з отриманими результатами та центрами кластеризації.

5 ВИПРОБУВАННЯ СИСТЕМИ

У рамках виконання дипломної роботи було перевірено працездатність та ефективність роботи розроблюваної автоматизованої системи. Зібрані статті було проаналізовано за кожен тиждень в період з 9 по 26 травня 2016 року, з метою відслідковування динаміки змін новин протягом обраних періодів часу. Було обрано відфільтровувати всі новини, які не пов'язані з зоною АТО та військовими діями на сході України чи з обміном Надії Савченко. Для цього сформульовано ключові слова, які наведено в таблиці 5.1. Статті, в яких не було знайдено жодного з ключових слів відкидались.

Таблиця 5.1 — Списки ключових слів

	Ключові слова
Події пов'язані з зоною АТО та військовими діями на сході України	«АТО», «бойовики», «спецслужби», «розстріляли», «вибух», «бійці», «зброя», «кіборги», «загинув», «поранені», «Мінські» «домовленості», «ЗСУ», «Донбас»
Події пов'язані з обміном Надії Савченко	«Надія Савченко», «обмін», «льотчиці», «Мінські», «домовленості», «Порошенко», «полон», «президент», «території», «України»

5.1 Контрольні приклади

5.1.1 Новини за період з 9 по 16 травня 2016 року

Даний період мав такі характеристики:

– кількість зібраних новин — 320;

– кількість новин, пов'язаних з Надією Савченко — 12;

– кількість новин, пов'язаних з зоною АТО та військовими діями на сході України — 74.

На рисунку 5.1 наведено результат нечіткої кластеризації зібраної за даний період інформації. Отримані результати свідчать, що на сході України велись активні військові дії, а ситуація з Надією Савченко була невизначеною.

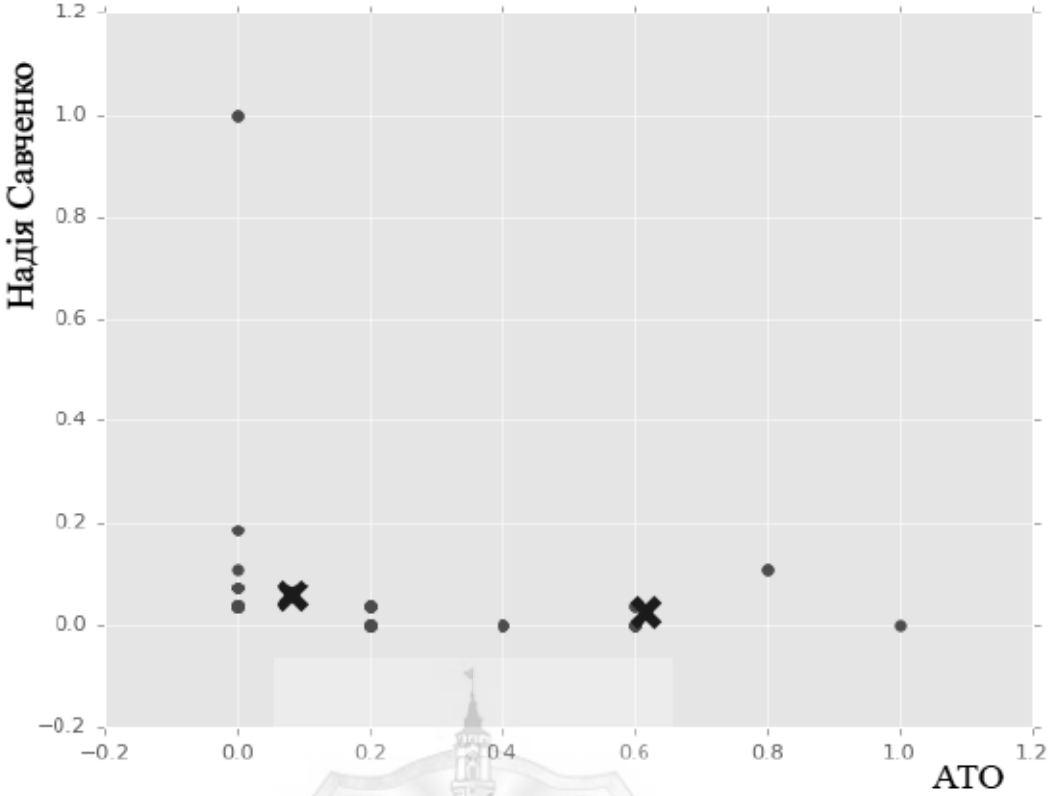


Рисунок 5.1 — Результат нечіткої кластеризації в період з 9 по 16 травня

5.1.2 Новини за період з 16 по 23 травня 2016 року

Даний період мав такі характеристики:

– кількість зібраних новин — 284;

– кількість новин, пов’язаних з Надією Савченко — 41;

– кількість новин, пов’язаних з зоною АТО та військовими діями на сході України — 37.

На рисунку 5.2 наведено результати нечіткої кластеризації зібраної інформації за даний період. Отримані результати свідчать, що наближався час повернення Надії Савченко на Батьківщину. В зоні АТО проводилось менше бойових дій в порівнянні з минулим тижнем.

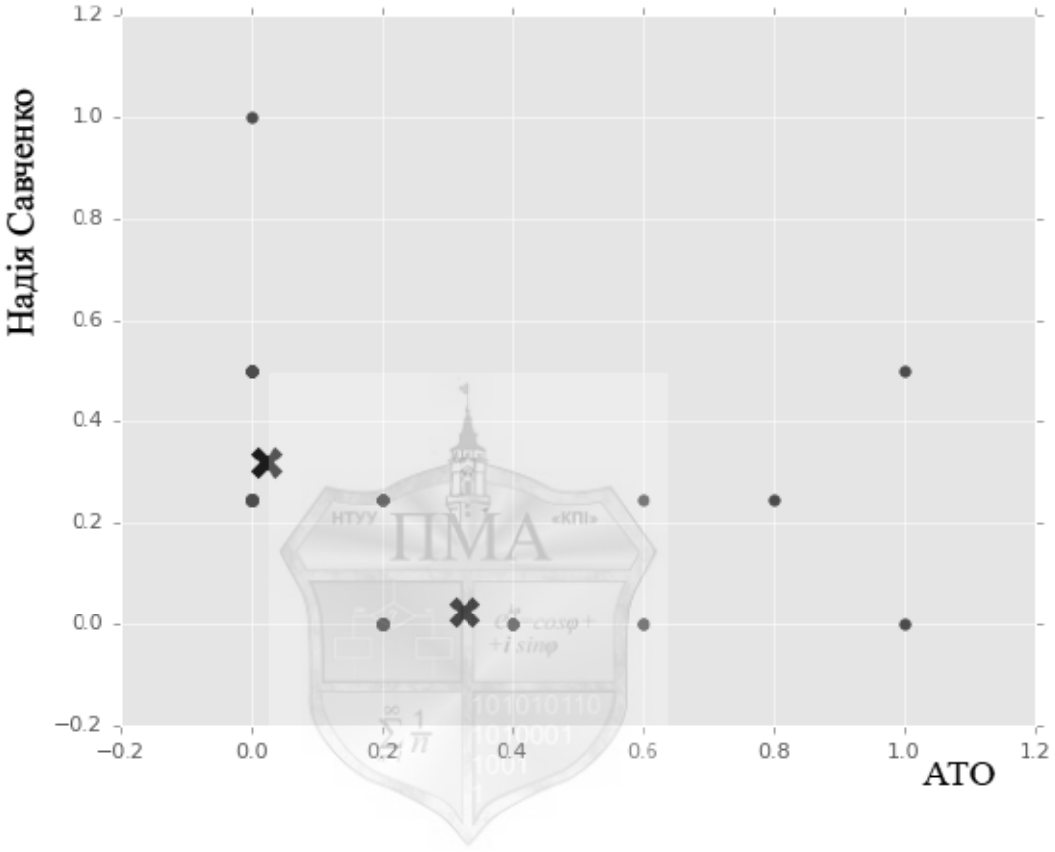


Рисунок 5.2 — Результат нечіткої кластеризації в період з 16 по 23 травня

5.1.3 Новини за період з 23 по 30 травня 2016 року

Даний період мав такі характеристики:

– кількість зібраних новин — 323;

– кількість новин, пов’язаних з Надією Савченко — 64;

– кількість новин, пов’язаних з зоною АТО та військовими діями на сході України — 55.

На рисунку 5.3 наведено результати нечіткої кластеризації зібраної інформації за даний період. Отримані результати пов’язані з поверненням Надії Савченко на Батьківщину та підтверджують цей факт. В зоні АТО проводилось більша кількість бойових дій, ситуація стала напруженішою.

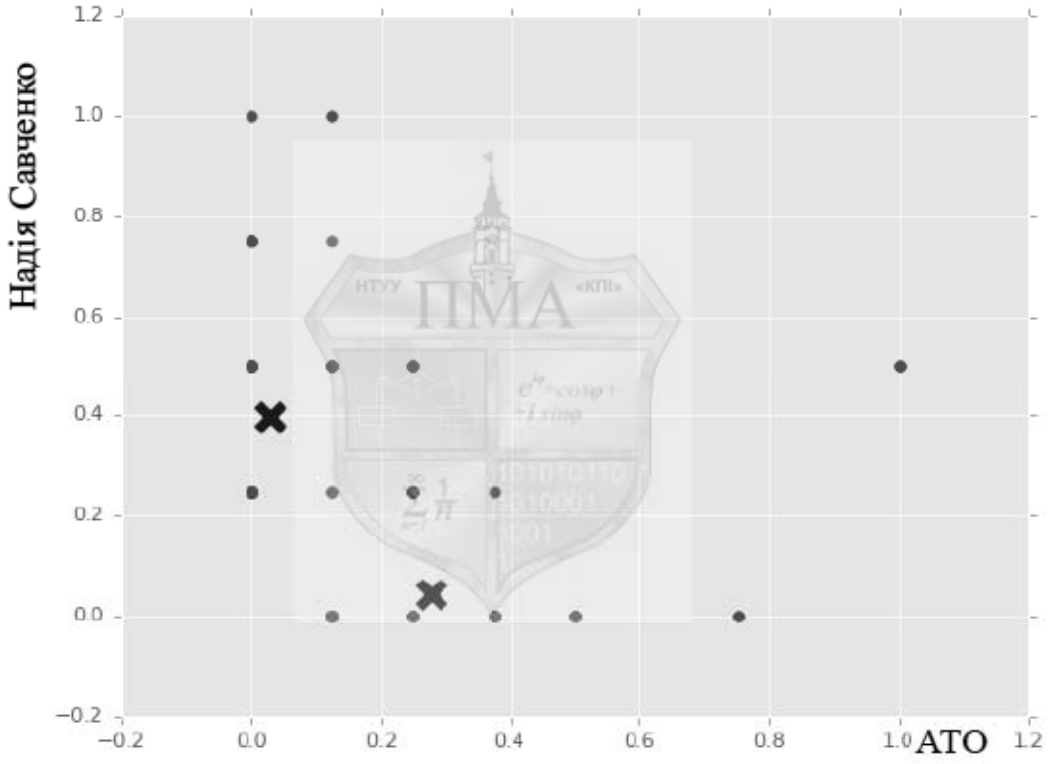


Рисунок 5.3 — Результат нечіткої кластеризації в період з 23 по 30 травня

5.2 Висновки до розділу

Розроблена система успішно шукає та зберігає інформацію з сайтів новин. Нечітка кластеризація зібраної інформації за три тижні підтверджує зміни, які відбувались на фронті, в зоні АТО, протягом вказаного періоду, а також показує, як змінювались новини напередодні повернення Надії Савченко в Україну.

ВИСНОВКИ

У дипломній роботі проведено огляд існуючих методів збору та ранжування даних з мережі Інтернет, включаючи веб додатки та фреймворки. Досліджено математичні методи класифікації та аналізу даних.

На базі фреймворку Scrapy створено систему пошуку, збереження та ранжування інформації з Web-сайтів новин tsn.ua та ukr.net. Зібрану інформацію проаналізовано за допомогою методу нечіткої кластеризації, що дозволило відфільтрувати та згрупувати новини відповідно до спільних характеристик. Отриману інформацію було класифіковано не статично, а на основі даних за кожен тиждень в період з 9 по 30 травня 2016 року, що дозволило оперативно відслідковувати ситуацію та зробити висновки про те, що в період з 16 по 23 травня було найбільше новин пов'язаних з АТО та військовими діями, а в період з 23 по 30 травня різко збільшилась кількість новин про Надію Савченко, що підтверджувало факт її повернення на Батьківщину.

Система ще буде вдосконалюватись і розширюватись для збільшення об'ємів корисної інформації та пошуку першоджерел. Також буде розроблено графічний інтерфейс, який дозволить швидше реагувати на виявленні загрози.

Основні ідеї, викладені в роботі, опубліковано в тезах [21] доповіді «Програмне забезпечення збору та ранжування даних з мережі Інтернет» на 8-й конференції молодих вчених ПМК-2016.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Додонов А. Г., Ланде Д. В., Прищепа В. В., Путятин В. Г. Конкурентная разведка в компьютерных сетях. — К.: ИПРИ НАН Украины, 2013. — 250 с.
2. Import.io [Електронний ресурс] — Режим доступу: https://www.import.io/
3. Kimono [Електронний ресурс] — Режим доступу: https://www.kimonolabs.com/
4. Apache nutch [Електронний ресурс] — Режим доступу: http://nutch.apache.org/
5. Perl [Електронний ресурс] — Режим доступу: https://www.perl.org/
6. Php [Електронний ресурс] — Режим доступу: http://php.net/
7. Java [Електронний ресурс] — Режим доступу: https://java.com/en/
8. Javascript [Електронний ресурс] — Режим доступу: https://www.javascript.com/
9. Python [Електронний ресурс] — Режим доступу: https://www.python.org/
10. Scrapy [Електронний ресурс] — Режим доступу: http://scrapy.org/
11. Oracle [Електронний ресурс] — Режим доступу: http://www.oracle.com
12. Mysql [Електронний ресурс] — Режим доступу: https://www.mysql.com/
13. Postgresql [Електронний ресурс] — Режим доступу: http://www.postgresql.org/

14. Sqlite [Електронний ресурс] — Режим доступу: https://www.sqlite.org/

15. Mongodb [Електронний ресурс] — Режим доступу: https://www.mongodb.com/

16. PostgreSQL: textsearch [Електронний ресурс] — Режим доступу: http://www.postgresql.org/docs/9.1/static/textsearch-controls.html.

17. Dembele D., Kastner P. C-means method for clustering microarray data // Bioinfor-matics. — 2003. — Vol. 19(8). — P. 973–980

18. Интеллектуальные технологии и системы: сборник учебно-методических работ и статей аспирантов и студентов. – М.: НОК «CLAIM», 2006. – С. 130-142.

19. Лялька Б. О. Оценка эффективности кластеризационных алгоритмов / Б. О. Лялька, Ю. В. Антонова-Рафи // Научные труды SWorld. — Выпуск 2. Том 2. — Иваново: Научный мир, 2015 — С. 25—29.

20. Котов А. А. Кластеризация данных / А. А. Котов, Н.О. Красильников, 2006. — 16 с.

21. Чертов О. Р. Програмне забезпечення системи збору та ранжування даних із Інтернету / О. Р. Чертов, Т. П. Рудник // Прикладна математика та комп'ютинг. ПМК, 2016 : восьма наук. конф. магістрантів та аспірантів, Київ, 20—22 квіт. 2016 р. : зб. тез доп. / [редкол.: Дичка І. А. та ін.]. — К. : Просвіта, 2016. — С. 351—356.