МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ ЗАХИСТУ ІНФОРМАЦІЇ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри
_____Михайло СТЕПАНОВ

«_____»_____ 2023 р.

На правах рукопису
УДК 004.422.8

КВАЛІФІКАЦІЙНА РОБОТА

ЗДОБУВАЧА ВИЩОЇ ОСВІТИ ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»

Тема: Програмний модуль виявлення повідомлень пропагандистського змісту в соціальних мережах

Виконавець: Поліна КОТИРЛО
Керівник: к.т.н. Олена ВИСОЦЬКА
Консультант розділу «Охорона навколишнього середовища»: к.т.н., доцент Тетяна ДМИТРУХА
Нормоконтролер: к.т.н. Олена ВИСОЦЬКА

Київ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет: Кібербезпеки та програмної інженерії

Кафедра: Комп'ютеризованих систем захисту інформації

Освітній ступінь: Магістр

Спеціальність: 125 «Кібербезпека»

Освітньо-професійна програма: «Безпека інформаційних і комунікаційних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Михайло СТЕПАНОВ

«__»_____ 2023 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи

здобувача вищої освіти Котирло Поліни Ігорівни

1. Тема: *Програмний модуль виявлення повідомлень пропагандистського змісту в соціальних мережах* затверджена наказом ректора від «15» вересня 2023 р. № 1814/ст.
2. Термін виконання з 16.10.2023 р. по 31.12.2023 р.
3. Вихідні дані: проаналізувати існуючі види та форми спаму для оцінки ризиків, пов'язаних зі спамом пропагандистського характеру; проаналізувати існуючі методи боротьби зі спамом, на основі аналізу виділити технології для боротьби зі спамом; розробити алгоритм та програмний модуль для виявлення повідомлень пропагандистського змісту в соціальних мережах.
4. Зміст пояснювальної записки: аналіз існуючих систем та методик боротьби зі спамом пропагандистського характеру; розробка програмного модуля виявлення повідомлень пропагандистського змісту в соціальних мережах на основі методів машинного навчання; тестування програмного модуля, проведено аналіз теми охорони навколишнього середовища.

КАЛЕНДАРНИЙ ПЛАН

виконання кваліфікаційної роботи

№ п/п	Етапи виконання кваліфікаційної роботи	Термін виконання етапів	Примітка
1.	Уточнення постановки задачі	16.10.2023	*Виконано*
2.	Аналіз літературних джерел	16.10.2023 - 23.10.2023	*Виконано*
3.	Обґрунтування вибору рішення	24.10.2023 - 02.11.2023	*Виконано*
4.	Збір інформації	03.11.2023 - 11.11.2023	*Виконано*
5.	Дослідження існуючих форм та видів спаму, оцінка загроз від спаму-пропаганди	12.11.2023 - 19.11.2023	*Виконано*
6.	Дослідження методів боротьби зі спамом, вибір технічних засобів та методів для реалізації програмного модуля	20.11.2023 - 25.11.2023	*Виконано*
7.	Розробка програмного модуля для виявлення повідомлень пропагандистського змісту в соціальних мережах	25.11.2023 - 05.12.2023	*Виконано*
8.	Апробація роботи на міжнародній науково-практичній конференція «живучість та резильєнтність – 2023»	19.10.2023	*Виконано*
9.	Оформлення і друк пояснювальної записки	07.12.2023 - 10.12.2023	*Виконано*
10.	Оформлення презентації	11.12.2023	*Виконано*
11.	Перевірка на антиплагіат	12.12.2023	*Виконано*
12.	Отримання рецензій від рецензента	22.12.2023	*Виконано*

6. Консультанти з окремих розділів

Розділ	Консультант (посада, П.І.Б.)	Дата, підпис	
		Завдання видав	Завдання прийняв
Охорона навколишнього середовища	Дмитруха Т.І.		

7. Дата видачі завдання: «16» жовтня 2023 р.

Здобувач вищої освіти Поліна КОТИРЛО

(підпис, дата)

Керівник Олена ВИСОЦЬКА

РЕФЕРАТ

Кваліфікаційна робота складається зі вступу, чотирьох розділів, загальних висновків, списку використаних джерел, додатків і має 84 сторінки основного тексту, 37 рисунків, 7 таблиць, 8 сторінок додатків. Список використаних джерел містить 26 найменування і займає 4 сторінок. Загальний обсяг роботи 97 сторінок.

Метою роботи є розробка програмного модулю виявлення повідомлень пропагандистського змісту в соціальних мережах, в якому за рахунок використання методів машинного навчання та методів обробки природної мови досягається класифікація коментарів на такі, які містять пропаганду та не містять.

В роботі вирішено задачу розробки програмного модулю який зчитує коментарі з соціальної мережі Facebook та, основуючись на методах машинного навчання приймає рішення чи містять ці коментарі пропагандистські наративи.

В роботі розроблено програмний модуль виявлення пропагандистських коментарів на основі теоретичного аналізу загроз пов’язаних зі спамом, методів боротьби з цими загрозами, аналізу методів машинного навчання як методу боротьби зі спамом.

Розроблений метод та програмне забезпечення відносяться до галузі інформаційної безпеки і можуть бути використані для підвищення рівня захищеності.

Можливі напрямки розвитку цієї роботи пов’язані з розширенням навчальної бази методів машинного навчання для більш точної оцінки коментарів з використанням розробленого програмного модуля.

Ключові слова: СПАМ, ІНФОРМАЦІЙНА БЕЗПЕКА, МЕТОДИ МАШИННОГО НАВЧАННЯ, МЕТОД НАЙБЛИЖЧОГО СУСІДА, МЕТОД ДЕРЕВО РІШЕНЬ, ОБРОБКА ПРИРОДНОЇ МОВИ, DATA SCIENCE.

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ММН − метод машинного навчання;

СМ − соціальна мережа

МНС − метод найближчого сусіда

МДР − метод дерево рішень

ОПМ − обробка природної мови

ЗМІСТ

ВСТУП

Актуальність. Популярність соціальних мереж зростає кожен день та по мірі зростання цієї популярності також розвиваються нові загрози. Людям, що є користувачами соціальних мереж важливо вчитись фільтрувати отриману звідти інформацію.

Разом з розвитком мереж комунікацій в інтернеті з’явились нові шляхи для поширення спаму. Спам - надзвичайно поширене явище з яким стикався кожен користувач інтернету хоча б раз. Через велику кількість його різновидів часто складно відразу розпізнати, що отримане повідомлення чи коментар є спамом.

Особливо небезпечним є такий спам, який містить в собі пропагандистські риторики. Це не просто надокучлива практика, але й спроба маніпуляції думками користувача, який бачить певне пропагандистське спам-повідомлення.

Для ефективного вирішення задач, що пов’язані зі спамом просто людським ресурсів часто недостатньо, оскільки для його поширення часто використовуються автоматичні розсилки чи боти. Тож методи боротьби з цим явищем теж потрібно і важливо автоматизувати.

Наразі для соціальної мережі Facebook немає такого рішення, яке б допомагало користувачу виявляти коментарі з пропагандистським вмістом. Тому в даній роботі проводиться аналіз видів спаму, методів боротьби з ними, проаналізовано алгоритми та програмні методи, які можуть допомогти в пошуці спаму та розроблено програмний модуль виявлення повідомлень пропагандистського змісту в соціальних мережах.

Відомі підходи до вирішення поставленої задачі. На сьогоднішній день існує багато різних застосунків та програмного забезпечення для пошуку спаму. Проте більшість з них шукає лише найпоширеніші його види і, спираючись на те, з яким саме видом спаму має боротись програмне

рішення мають ряд специфічних налаштувань. Тож часто постає питання як розпізнати саме якийсь певний вид спаму.

Метою роботи є розробка програмного модулю виявлення повідомлень пропагандистського змісту в соціальних мережах, в якому за рахунок використання методів машинного навчання та методів обробки природної мови досягається класифікація коментарів на такі, які містять пропаганду та не містять.

Для досягнення поставленої мети необхідно розв'язати наступні **задачі:**

- проаналізувати види та форми спаму, методи боротьби зі спамом у соціальних мережах та на основі результату проведеного аналізу обрати оптимальні методи для вирішення задачі виявлення повідомлень пропагандистського змісту в соціальних мережах;
- розробити програмний модуль виявлення повідомлень пропагандистського змісту в соціальних мережах, в якому за рахунок поєднання двох різних методів машинного навчання, досягається підвищення ймовірності правильного розпізнавання повідомлення пропагандистського змісту в соціальних мережах.
- провести тестування розробленого програмного модуля виявлення повідомлень пропагандистського змісту в соціальних мережах, що дасть змогу дослідити доцільність використання розробленого модулю для вирішення поставленої мети.

Галузь застосування. Розроблений програмний модуль відноситься до галузі інформаційної безпеки та може бути використаний користувачем соціальної мережі Facebook для розпізнавання повідомлень пропагандистського змісту.

Об’єктом дослідження є процес виявлення повідомлень пропагандистського змісту в соціальних мережах.

Предметом дослідження є методи розпізнавання спаму в соціальних мережах та програмні рішення виявлення спаму-пропаганди.

Методи дослідження базуються на основі структурного аналізу видів та форм спаму, і методів боротьби зі спамом для вибору оптимальних технологій реалізації розробленого програмного модуля та об’єктно-орієнтованому програмуванні для програмної реалізації розробленого програмного модуля. Під час розробки програмного застосунку було використано експериментальний та порівняльний методи дослідження.

Новизна одержаних результатів полягає в наступному: розроблено програмний модуль для пошуку пропагандиських коментарів в соціальній мережі Facebook з використанням двох методів машинного навчання для підвищення ймовірності знаходження відповідних риторик, що дало змогу знаходити коментарі з пропагандистським вмістом з точністю 86%.

Практична цінність отриманих результатів:

- розроблено програмний модуль, що може використовуватись користувачем соціальної мережі Facebook для пошуку коментарів з пропагандистським вмістом, з використанням мов програмування Python, JavaScript та методів машинного навчання.

Апробація. Котирло П.І., Висоцька О.О., Використання спаму як способу поширення пропаганди в соціальних мережах // Живучість та резильєнтність – 2023: міжнародна науково-практична конференція 19 жовтня 2023 р.: тези доповіді. – К., 2023. – С.131-133.

РОЗДІЛ 1.

ПОНЯТТЯ СПАМУ. СПАМ ЯК СПОСІБ ПОШИРЕННЯ ПРОПАГАНДИ

1.1. Поняття спаму, види спаму

Збільшення користувачів Інтернету - це одна з ключових тенденцій в сучасному світі. Тож з кожним роком практика спаму може охоплювати більшу аудиторію користувачів.

Спам - це надмірна відправка небажаної або невідомої інформації, такої як комерційні пропозиції, рекламні повідомлення або навіть шкідливий вміст, на адреси електронної пошти чи інші способи зв'язку, без згоди чи інтересу одержувачів. Це надокучлива практика, яка часто спрямована на використання чужих ресурсів без дозволу та може завдати шкоди як особистій приватності, так і безпеці інтернет-користувачів.

Тенденція зростання кількості спаму підкреслюється та описується науковцями, які досліджують цю тему, особливо в контексті інформаційної війни в умовах якої знаходиться наша країна.

В роботі "Аналіз методів виявлення деструктивних впливів в соціальних інтернет сервісах" розглядаються загрози пов’язані з поширенням спаму в інтернеті, зокрема також в соціальних мережах.

В роботі "Огляд основних підходів до розпізнавання спаму методами машинного навчання"[2] проаналізовано різні методи машинного навчання для їх використання для пошуку спаму в соціальних мережах.

В роботі "Дослідження методів класифікації листів українською мовою з метою виявлення спаму" [3] розглядається класифікація листів методами машинного навчання саме українською мовою, оскільки є багато рішень

створених для англомовних ресурсів, проте важливо також створювати україномовні аналоги.

Тож питання пошуку спаму в соціальних мережах - важливе та є актуальною темою для дослідження.

Спам це глобальна проблема, актуальність боротьби з якою щороку зростає. Згідно з дослідженнями аналітичної компанії Statista (рис. 1.1) близько 84% всіх email повідомлень є спамом [4]:

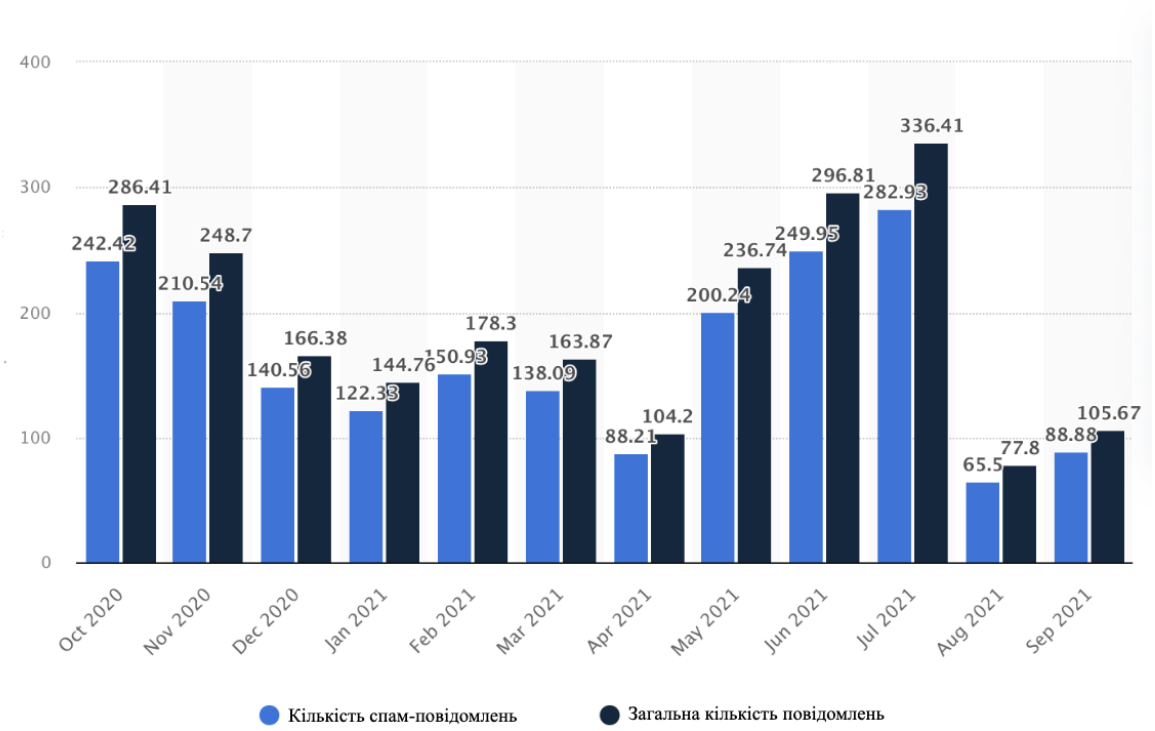


Рис. 1.1. Співвідношення кількості спам-листів до загальної кількості листів

Спам існує в різних *формах і видах*. Види спаму за шляхами його поширення (рис. 1.2):

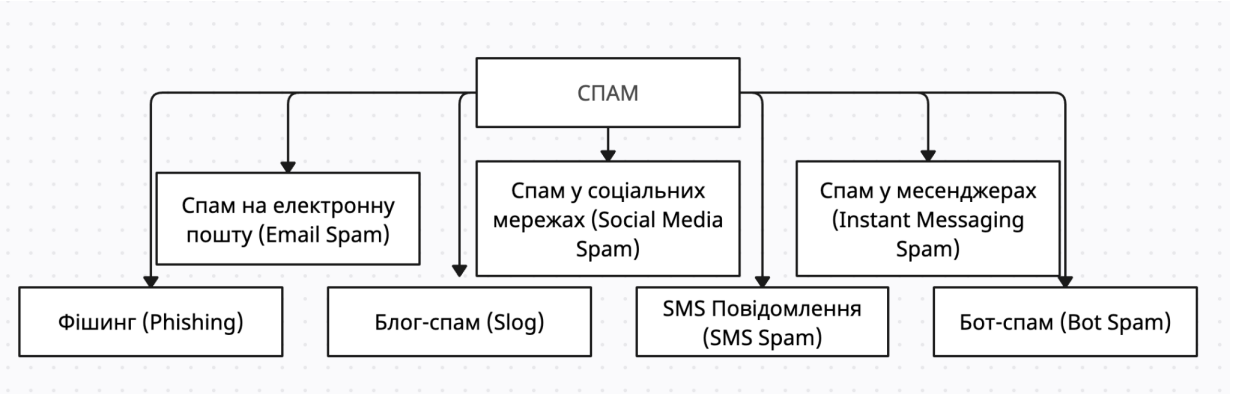


Рис. 1.2. Класифікація спаму за шляхами поширення

1. Спам на електронну пошту (Email Spam). Електронний спам — це, безумовно, найпоширеніший вид спаму. Він включає небажані електронні листи, які містять рекламу, шахрайство, фішингові атаки та інші небезпечні повідомлення. Здебільшого, такий спам має комерційний характер і містить посилання на підозрілі веб-сайти.

Загалом термін "спам" набув загальної популярності у 1994 році, коли два американських юристи розіслали велику кількість небажаних повідомлень на електронні адреси, де вони намагалися просунути свої послуги [5].

Приклад:

Привіт, Ми з радістю повідомляємо вас, що ви виграли у нашому щомісячному лотерейному розіграші! Для отримання вашого призу, будь ласка, введіть свої особисті дані та натисніть на посилання нижче. Це шанс змінити ваше життя!

2. Спам у СМ (Social Media Spam): Ця форма спаму включає небажані повідомлення, коментарі, запити на дружбу тощо в соціальних мережах, таких як Facebook, Twitter, Instagram. Зловмисники можуть використовувати це для поширення фішингу, рекламних схем або навіть поширення шкідливих посилань. Популярність соціальних мереж робить їх ідеальною платформою для поширення спаму.

Приклад:

Ми організовуємо ексклюзивний конкурс для наших підписників! Зроби репост цього повідомлення у своєму профілі, зареєструйся за посиланням нижче та познач нас, і ти можеш виграти весняний набір косметики від нашого партнера.

[Посилання на реєстрацію]

3. Фішинг (Phishing): Фішинговий спам — це вид спаму, в якому зловмисники намагаються обманом викликати у користувача передачу особистої інформації, такої як паролі або номери кредитних карток.

Приклад:

Ми виявили певні аномалії в вашому обліковому записі та вам необхідно перевірити ваші особисті дані. Будь ласка, натисніть на посилання нижче та введіть свої дані для вирішення цього питання. [Шахрайське посилання на фішинговий сайт]

Фішинговий спам може бути комбінований також з іншими видами спаму. У Міністерстві внутрішніх справ України у 2023 році повідомляли про фішинговий спам, який поширювався СМ Facebook [6].

4. Спам у месенджерах (Instant Messaging Spam): Зловмисники можуть надсилати небажані повідомлення через месенджери, такі як WhatsApp, Telegram або Viber. Це може бути реклама, ланцюжки листів, або шахрайські пропозиції.

5. Блог-спам (Blog Comment Spam): Спам у коментарях блогів, відеоблогів і форумів. Здебільшого це автоматизовані повідомлення, які містять посилання на сайти зі східним вмістом.

Такий вид спаму вирізняється тим, що один спам може потенційно досягти стільки ж глядачів, скільки є читачів допису в блозі, до якого прикріплено спам, спам може бути замаскованим, шляхом включення в нього тексту, пов'язаного з дописом у блозі або наївним копіюванням і вставкою, або за допомогою більш просунутої ОПМ [8].

Приклад:

Привіт, друзі! Сьогодні в моєму новому блозі я розповім вам про неймовірний спосіб заробітку в Інтернеті, який змінить ваше життя! Просто перейдіть за посиланням нижче та дізнайтеся, як отримати тисячі доларів на своєму рахунку всього за кілька днів:

6. Текстові повідомлення (SMS Spam): Зловмисники можуть надсилати небажані текстові повідомлення на мобільні телефони. Це може бути спам від невідомих номерів або рекламні повідомлення від фірм і послуг.

7. Бот-спам (Bot Spam): Боти можуть надсилати автоматизовані повідомлення на форумах, СМ та інших платформах. Це може бути спам в коментарях, відгуках або публікаціях.

Спам-боти виконують свої дії автоматично. Основні шкідливі дії ботів цієї категорії: спам-боти, що збирають адреси E-mail з контактних форм і гостьових книг; програми, які завантажили Internet-канал потоком непотрібної інформації (як правило, рекламного характеру); сайти, які збирають інформацію у нешкідливих сайтах, для використання її в автоматично створюваних до- рвеях (спеціальні HTML-сторінки, складені для високого позиціонування у пошукових системах за певним ключовим словом) [8].

Класифікація спаму за *вмістом* (рис. 1.3):

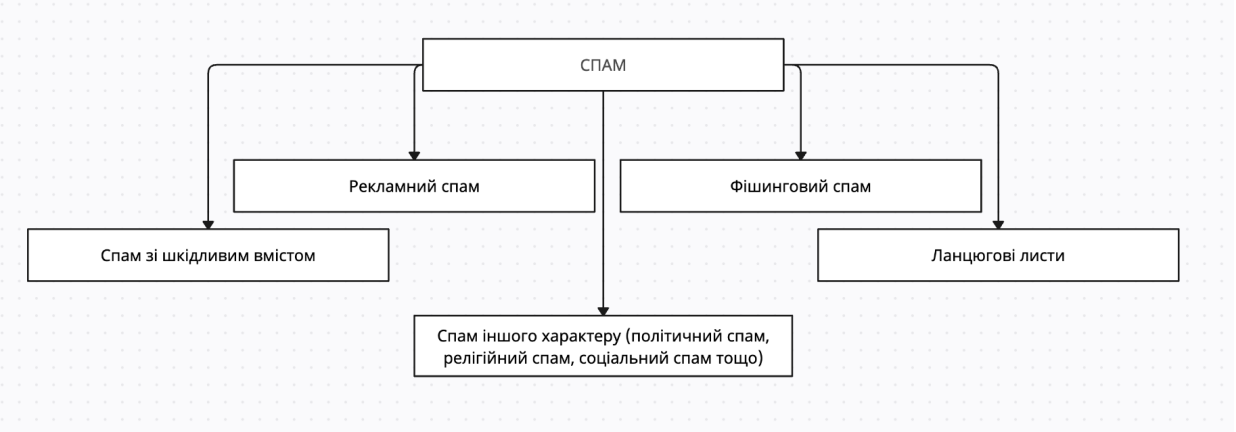


Рис. 1.3. Класифікація спаму за вмістом

1. Рекламний спам: Включає в себе комерційну рекламу продуктів, послуг чи веб-сайтів. Це може бути різноманітні рекламні пропозиції та рекламні кампанії.

2. Фішинговий спам: Містить спроби використати фальшиві чи обманливі повідомлення для отримання конфіденційної інформації, такої як паролі, номери кредитних карток чи інші особисті дані.

3. Спам зі шкідливим вмістом (Malware): Мета - зараження комп'ютера або пристрою одержувача.

Зазвичай використовують шкідливе програмне забезпечення для запуску кібератак на комп'ютери жертв. Будь-яке програмне забезпечення, яке навмисно виконує шкідливе корисне навантаження на комп'ютерах (комп'ютерах, смартфонах, комп'ютерних мережах тощо), вважається шкідливим програмним забезпеченням. Існують різні типи шкідливих програм, зокрема віруси, хробаки, троянські програми, руткіти та програми-вимагачі.

4. Ланцюгові листи (Chain Letters): Масові повідомлення, які закликають одержувачів надсилати їх далі. Зазвичай містять погрози негараздів у випадку, якщо лист не буде розісланий далі.

Приклад:

Привіт, це особливий лист!

Якщо ви перешлете його ще 5 друзям, то ваш тиждень буде наповнений позитивом та везінням! Не порушуйте ланцюг і дивіться, як ваші дні наповнюються щасливими подіями! [Посилання на неперевірений веб-сайт] Поділіться цим листом для щасливого тижня!

5. Спам іншого характеру: політичний спам, релігійний спам, соціальний спам тощо.

Приклад:

Важливо для вас! Розкрийте таємниці, які приховує влада! Натискайте на посилання та дізнайтеся правду, яку намагаються сховати від громадськості.

Класифікація спаму за вмістом допомагає ідентифікувати та фільтрувати небажані повідомлення, сприяючи поліпшенню якості комунікацій та захисту від шкідливого вмісту.

Спам може бути в текстовій або мультимедійній формах. Розглянемо характеристику кожної з *форм* спаму.

Текстовий спам

Текстовий спам представляє собою масове розсилання небажаних текстових повідомлень з рекламою, фішинговими спробами або шахрайством, пропагандою. Ця форма спаму часто використовується через електронну пошту, але також може з'являтися у коментарях на веб-сайтах, СМ чи інших комунікаційних платформах. Спамові повідомлення можуть містити недоречний контент, порнографію, неперевірені пропозиції чи просити особисту інформацію.

Мультимедійний спам

Мультимедійний спам включає в себе розсилку небажаних повідомлень, які містять мультимедійний вміст, такий як відео, аудіо або зображення. Ця форма спаму може виявитися в електронній пошті, СМ та месенджерах. Спамові повідомлення можуть містити рекламу, фішингові спроби або шкідливий вміст. Користувачі часто отримують небажані файли, відправлені як вкладення, які можуть містити віруси або інші шкідливі програми.

Для того щоб протидіяти спаму, якнайперше потрібно ефективно його виявляти. Спам може мати різноманітні ознаки, які допомагають визначити його. Ось деякі загальні ознаки спаму (рис. 1.4):

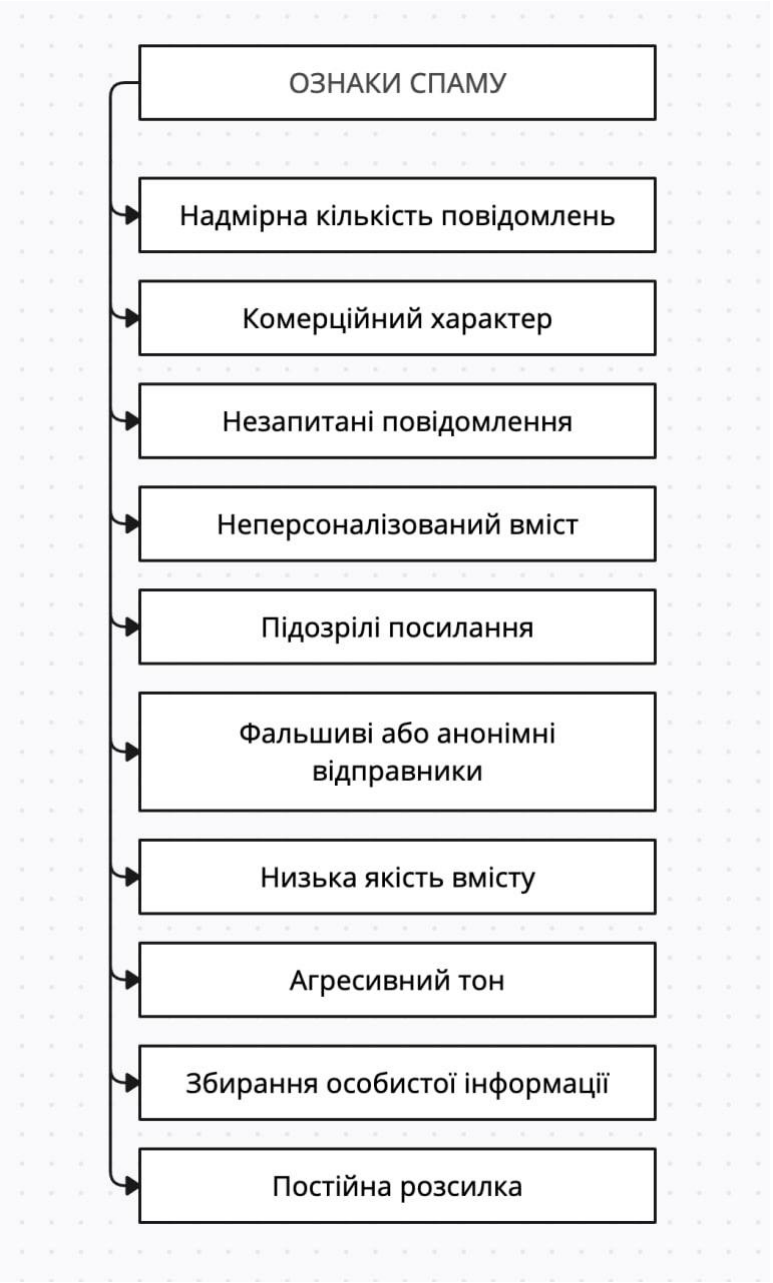


Рис. 1.4. Ознаки спаму.

- Надмірна кількість повідомлень: Спамери зазвичай відправляють велику кількість повідомлень, коментарів або листів в короткий час.
- Комерційний характер: Більшість спаму має комерційний характер і містить рекламу продуктів, послуг чи веб-сайтів.

Проте реклама, яка надсилається за допомогою спаму, має суттєві відмінності від звичайної, по-перше – звичайна реклама поширюється традиційними способами: через телебачення, радіо або ж друковані видання. До того ж, користувач у будь який момент може відмовитися від нецікавої для нього реклами, на відміну від спаму [12].

- Незапитані повідомлення: Спам зазвичай відправляється людям, які не виражали бажання отримувати ці повідомлення або не підписувалися на розсилку.

Тобто, якщо одержувач погоджується на розсилку, то це вже не спам; спам розсилається у великих кількостях (відправник розсилає багато однакових повідомлень, а одержувачі вибираються випадково) [11].

- Неперсоналізований вміст: Спам може містити загальні повідомлення, які не враховують індивідуальних потреб чи інтересів одержувача.

- Підозрілі посилання: Спамові повідомлення можуть містити посилання на підозрілі веб-сайти, які можуть бути шкідливими.

- Фальшиві або анонімні відправники: Спамери можуть приховувати свою справжню ідентифікацію або використовувати фейкові акаунти.

- Низька якість вмісту: Багато спамових повідомлень мають низьку якість вмісту, помилковий текст, граматичні помилки чи низькорівневу інформацію.

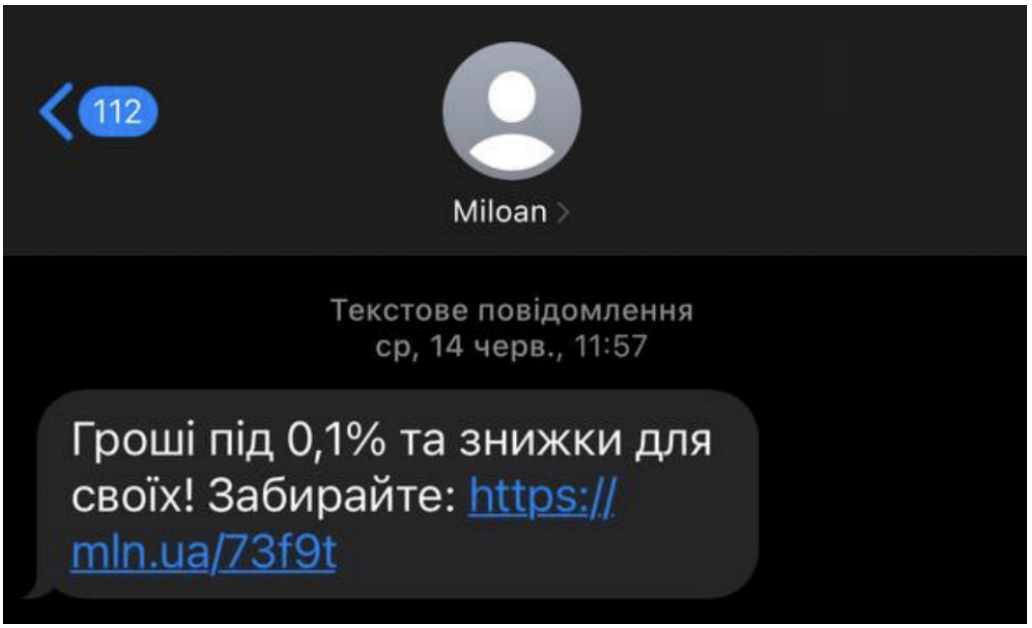
- Агресивний тон: Деякі спамові повідомлення можуть бути агресивними, намагаючись викликати відгук або дію від одержувача.

- Збирання особистої інформації: Деякий спам може намагатися обманом отримати особисту інформацію, таку як паролі, номери кредитних карток чи інші конфіденційні дані.

- Постійна розсилка: Спам може надходити регулярно та систематично, іноді щоденно чи навіть кілька разів на день.

Важливо розуміти, що не кожне повідомлення яке підпадає під одну або навіть декілька вищевказаних ознак є спамом, проте ці ознаки дозволяють краще фільтрувати звичайні повідомлення.

Розглянемо приклад спам-повідомлення (рис 1.5):



1.5. Приклад повідомлення, що містить спам

За вищеописаними ознаками повідомлення має комерційний характер, є незапитаним, має неперсоналізованих вміст, містить підозріле посилання, має анонімного відправника (без номеру телефону). З упевненістю можна сказати, що це повідомлення - *спам*.

Оскільки програмний модуль, що буде розроблятися в даній роботі застосовується для виявлення спаму вСМ, розглянемо більш детально саме спам у СМ.

1.2. Спам у соціальних мережах

СМ є привабливою місцевістю для поширення спаму з кількох причин:

- Велика аудиторія: СМ мають мільйони користувачів з усього світу, і це робить їх привабливими для спамерів, які намагаються досягти широкої аудиторії.

Зараз у світі налічується 4,76 мільярда користувачів СМ, що становить трохи менше 60 відсотків населення світу. Чисте збільшення користувачів СМ

цього року на 137 мільйонів нових користувачів дорівнює річному зростанню на 3 відсотки [7]. Таким чином щорічно аудиторія для поширення спаму з кожним роком росте.

- Спільноти і групи: Багато СМ мають функціонал, що дозволяє користувачам приєднуватися до спільнот, груп і форумів. Це створює можливість для спамерів надсилати небажані повідомлення великим групам людей, які мають спільні інтереси.
- Анонімність: У деяких випадках спамери можуть створювати фейкові профілі або використовувати анонімні акаунти, що робить їх важко відстежуваними та виявляє їхні справжні мотиви.

Якщо розглядати досліджувану мережу Facebook, вона містить функціонал публікувати дописи анонімно або під вигаданим іменем, що робить проблематику поширення спаму ще більш глобальною.

- Можливість поширення вірусів і шкідливого вмісту: Спамери можуть використовувати СМ для поширення вірусів, шкідливого програмного забезпечення або фішингових атак, спрямованих на отримання особистої інформації користувачів.

Спам у СМ це глобальна проблема, яка з кожним роком стає все більш поширеною, згідно з дослідженнями аналітичної компанії. 60% користувачів СМ стверджують, що бачать спам-коментарі щодня і щотижня, 31% спам-повідомлень припадають на СМ Facebook [9] (рис 1.6):

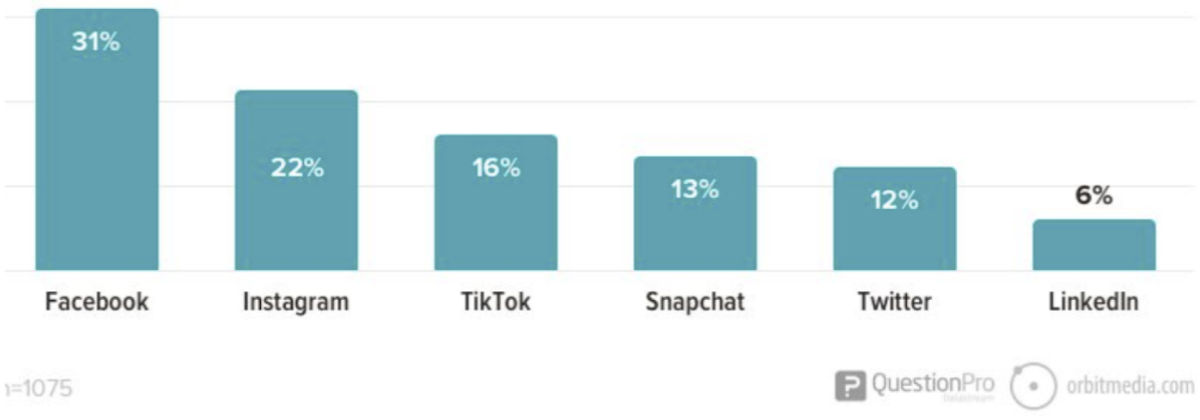


Рис. 1.6. Найпоширеніші за кількістю спаму соціальні мережі.

СМ *Facebook* найпоширеніша за кількістю спаму та найпопулярніша СМ в Україні, тому саме вона була обрана як мережа для дослідження.

За загальнонаціональним дослідженням ГО «Детектор медіа» в період із 17 до 22 грудня 2021 року щодо медіаспоживання в Україні 56% українців використовують СМ Facebook задля отримання інформації (рис. 1.6), ще 25% використовують Instagram, яка є дочірньою компанією Meta [10] (рис. 1.7):

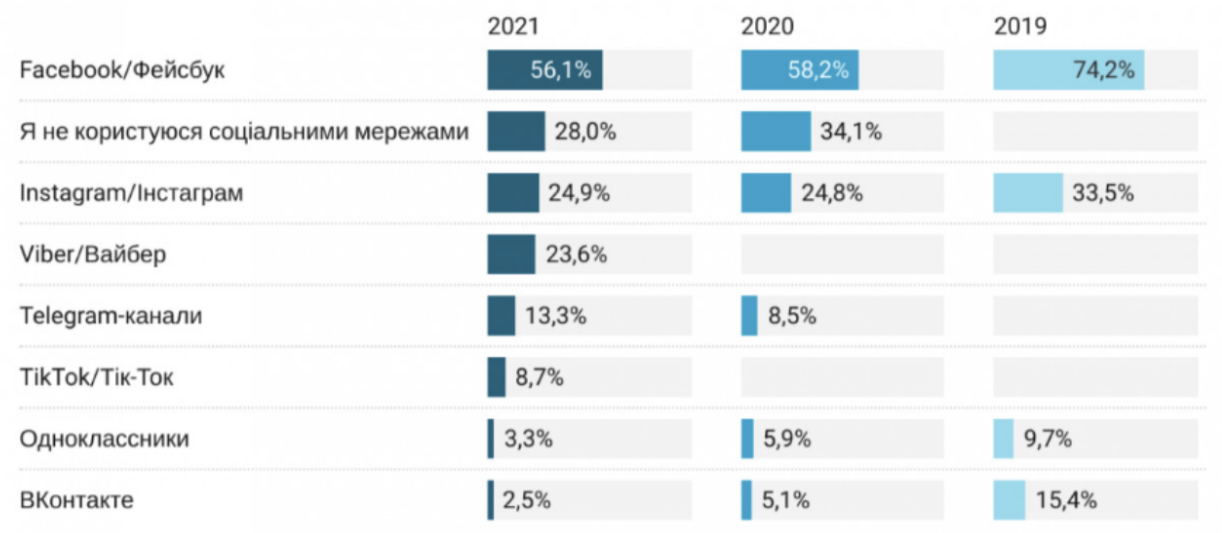


Рис 1.7. Загальнонаціональне дослідження щодо медіаспоживання

Також згідно з опитуванням [10] 79% людей вказали, що спам-повідомлення в СМ містять фейкові новини (рис 1.8). Отже спам також може бути джерелом поширення дезінформації, пропаганди.

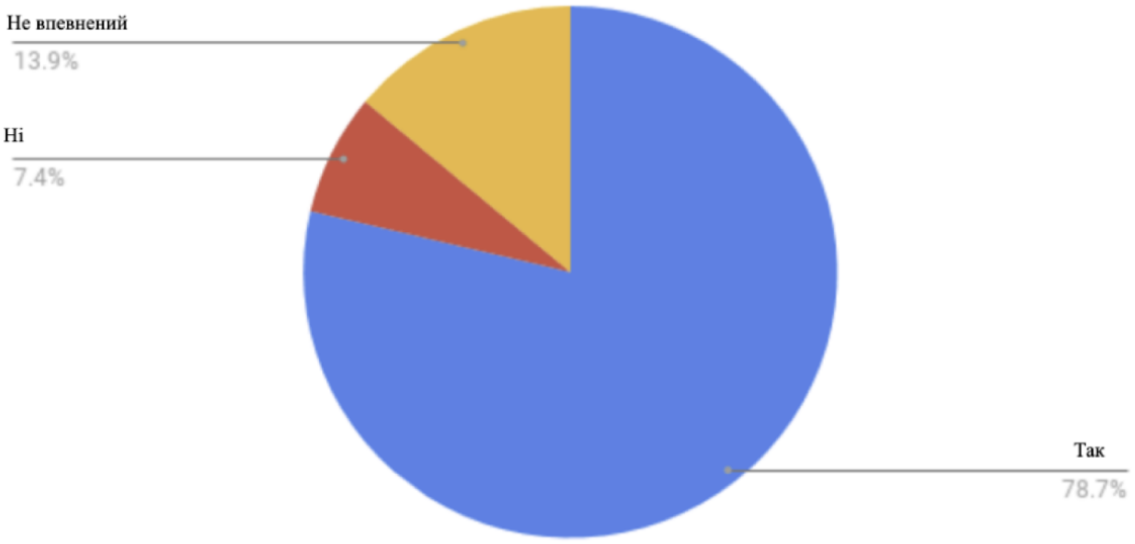


Рис. 1.8. Кількість фейкових новин в спам-повідомленях у СМ

Отже, спам може бути не тільки комерційного чи рекламного характеру, також спам може використовуватись як спосіб поширення фейкових новин або навіть пропаганди.

1.3. Спам як спосіб поширення пропаганди

В випадку програмного модуля, що буде розроблятись ми будемо класифікувати спам політичного характеру, а саме коментарі пропагандистського характеру.

Пропаганда - широке поняття, для якого різні науковці приводять різні визначення. За істориком Аланом Нельсоном пропаганда має такі ознаки умисні та систематичні спроби змінити сприйняття, емоції, важливості контролю передачі односторонніх повідомлень, скеровування поведінки цільової аудиторії [14].

Пропаганда відрізняється від звичайного спілкування і вільного обміну ідеями чи інформацією умисністю і акцентом на маніпуляції. У пропагандиста є конкретна мета, щоб досягти її, він навмисно відбирає факти, аргументи і

символи та подає їх так, щоб досягти найбільшого ефекту. Щоб максимізувати ефект, він може оминати істотні факти чи спотворювати їх, а також може відвертати увагу аудиторії від інших джерел інформації [13].

В умовах інформаційної війни в яких наразі знаходиться наша країна дуже важливо контролювати та запобігати розвитку пропаганди в усіх суспільних сферах, в тому числи в СМ. Зважаючи на популярність СМ також складним завданням є фільтрація медійного простору від небезпечних та пропагандистських риторик.

В СМ практика використання спаму в коментарях, приватних повідомленнях чи у блогах є поширеним явищем в кількох причин (рис. 1.9):

Рис. 1.9. Причини поширення пропаганди в СМ

1. Доступ до широкої аудиторії: СМ залучають мільйони користувачів з усього світу. Це надає можливість пропагандистам доставляти своє повідомлення аудиторії різного масштабу. Крім того, серед великої кількості людей пропагандистське повідомлення може неконтрольовано поширюватись навіть без участі першоджерела, оскільки маніпулятивні риторики, що використовуються в спам-повідомленнях часто спонукають користувача поширювати ці риторики.

2. Зручність поширення інформації: У СМ дуже просто ділитися постами, відео, зображеннями та іншими типами контенту. Це робить їх ідеальними для поширення пропаганди. Також СМ дають різноманіття шляхів для поширення спаму: повідомлення, коментарі, блоги, поширення фотографій.

Особливості поширення інформації в соцмережах - швидкість комунікаційного обміну, відсутність просторово-часових обмежень, позірна масштабність аудиторії, фільтрація інформації – посилюють вплив на користувача. Повідомлення в соцмережах поширюються на сторінках медіаресурсів, за якими користувач обрав стежити; шляхом розсилки новин від спільноти, до якої користувач долучився за власним вибором, або ж безпосередньо від користувача до користувача, який є другом або фоловером. З психологічної точки зору користувач сприймає свою сторінку як особистий простір, що обумовлює апріорі вищий рівень довіри до одержуваної в мережі інформації, ніж з інших джерел [15].

3. Анонімність і прихованість: Деякі пропагандисти можуть діяти анонімно або використовувати фейкові профілі, що ускладнює відстеження їхньої діяльності та виявлення. Деякі СМ не вимагають верифікації профілю для реєстрації, відслідкувати такі профілі - складне завдання.

1.4. Висновки до розділу 1

В сучасному світі разом із розвитком інформаційних технологій, засобів поширення інформації проблема спаму також не стоїть на місці. Спам є глобальною проблемою, і статистика показує, що близько 84% всіх email повідомлень є спамом.

У цьому розділі було досліджено ключові аспекти спаму та його ознаки. Спам існує в різних формах і видах, і кожен з них має свої особливості та методи боротьби. Для ефективної боротьби зі спамом перш за все важливо виявляти його, і це можливо завдяки розпізнаванню ряду загальних ознак спаму.

Дослідження форм, ознак та видів спаму дає можливість підібрати найбільш оптимальні шляхи та методи боротьби зі спамом, які будуть використовувати у програмному модулі, що буде розроблено.

СМ стають ідеальною платформою для поширення спаму через їхню популярність та доступність для великої аудиторії користувачів. Спам в СМ з кожним роком стає все більш поширеним явищем. СМ мають велику аудиторію, що робить їх привабливими для поширення спаму.

З кожним роком кількість користувачів СМ зростає, що робить їхню аудиторію ще більшою та різноманітною, тож боротьба зі спамом у СМ - важливе та актуальне завдання.

Функціонал СМ, такий як спільноти та групи, створює можливість спамерам надсилати небажані повідомлення великим групам людей зі спільними інтересами. Анонімність, яку можуть використовувати деякі спамери, ускладнює виявлення та блокування їхніх дій.

Особливо небезпечним є поширення недостовірної інформації, пропаганди шляхом поширення спаму.

Пропаганда характеризується умисністю та акцентом на маніпуляціях з метою досягнення конкретних цілей. Пропагандисти намагаються навмисно відбирати факти, аргументи і символи для досягнення найбільшого впливу на аудиторію. Вони можуть використовувати різні методи, включаючи спотворення фактів і відвертання уваги від інших джерел інформації.

Пропаганда в сучасному світі має особливу актуальність через інформаційну війну та впливовість СМ. Особливо небезпечним це явище є в умовах інформаційної війни, тож було окремо розглянуто спам як спосіб поширення пропаганди.

Процес поширення пропаганди в СМ включає в себе такі фактори, як доступ до широкої аудиторії, зручність поширення інформації та анонімність і прихованість дій. Це робить СМ ідеальними платформами для поширення пропаганди.

У зв'язку з цим, контроль та запобігання розвитку пропаганди в СМ стає дуже важливим завданням в умовах інформаційної війни, тож програмний модуль, що буде розроблено націлений на виявлення пропагандистських повідомлень, що поширюються СМ.

У другому розділі кваліфікаційної роботи, базуючись на досліджених поняттях, будуть розглянуті методи та технології боротьби з явищем спаму пропагандистського характеру у СМ.

РОЗДІЛ 2.

ТЕХНОЛОГІЇ ВИЯВЛЕННЯ СПАМУ. МЕТОДИ МАШИННОГО НАВЧАННЯ.

2.1. Технології виявлення спам-повідомлень

В першому розділі роботи ми розглядаємо види та спаму та його ознаки. Технології виявлення спаму ідентифікують небажані повідомлення на основі цих видів та ознак та включають в себе різноманітні методи та алгоритми.

Виявлення спам-повідомлень — це важлива задача для забезпечення безпеки та зручності користувачів в сучасному інтернет-середовищі. Існує кілька технологій та методів для виявлення спаму у різних контекстах, таких як електронна пошта, текстові повідомлення, коментарі в СМ та інші.

Наразі існує багато підходів та технологій виявлення спаму (рис. 2.1):

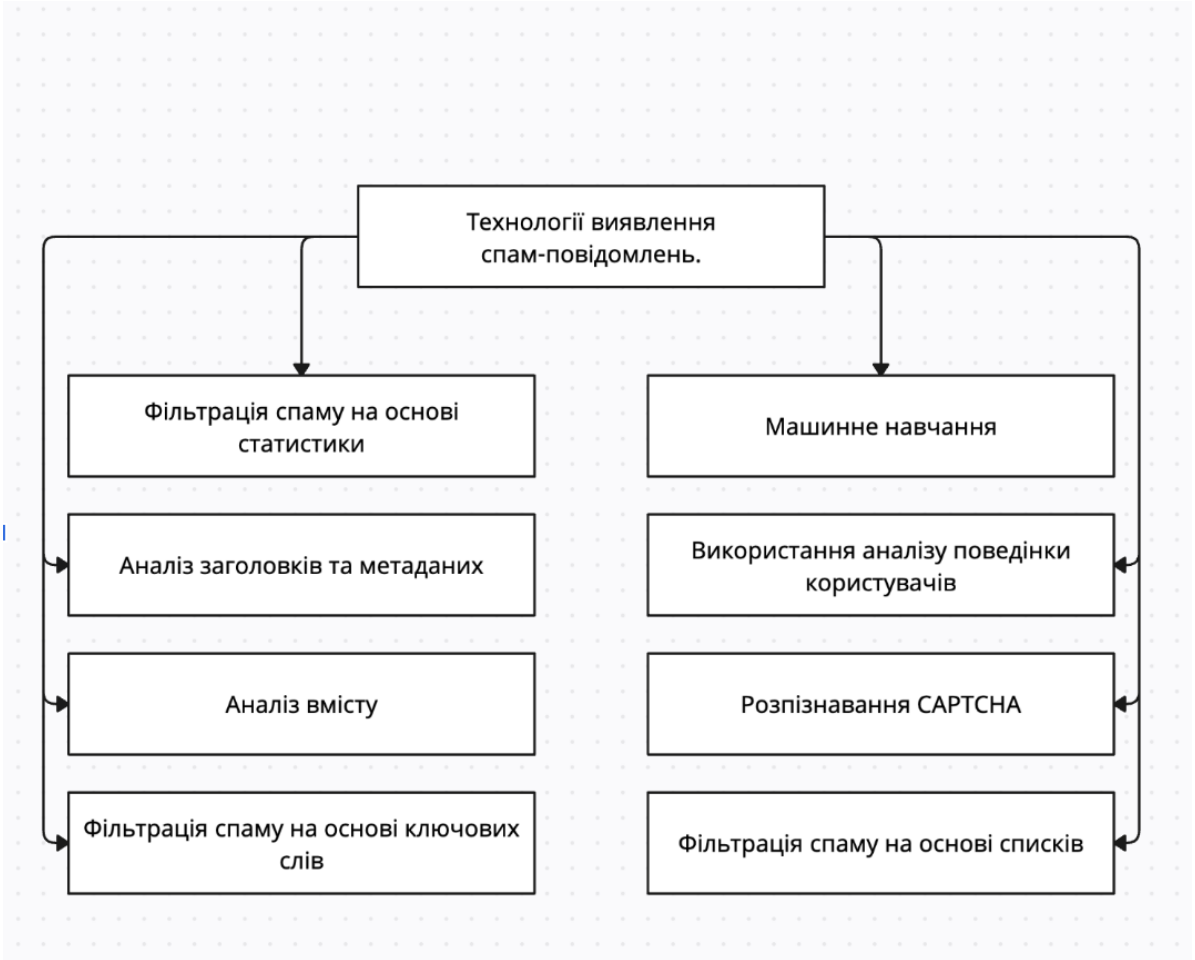


Рис. 2.1. Технології виявлення спам-повідомлень

1. Фільтрація спаму на основі ключових слів (Keyword-Based Filtering): Цей метод виявлення спаму використовує словник ключових слів, які часто зустрічаються в спам-повідомленнях. Якщо повідомлення містить багато таких слів, воно вважається спамом.

Щоб протистояти небажаним повідомленням електронної пошти, постачальники послуг почали сканувати вміст, спочатку шукаючи дайджести, пов'язані з попередніми повідомленнями про спам, а пізніше розглядаючи ключові слова. Коли постачальники послуг почали блокувати велику кількість повідомлень, що містять ключові слова зі спамом, спамери швидко почали протистояти шляхом перетворювання ключових слів [13].

2. Фільтрація спаму на основі статистики (Bayesian Filtering): Цей метод використовує статистичні моделі для виявлення спаму. Алгоритми

навчаються на основі великого обсягу повідомлень і визначають ймовірність того, що конкретне повідомлення є спамом або не спамом.

Класифікатор наївного Баєса використовує функції місць слів для ідентифікації електронної пошти зі спамом , що є підходом, який зазвичай використовується при класифікації тексту. Наївні класифікатори Байєса працюють шляхом співвіднесення використання маркерів (як правило, слів, а іноді й інших речей) зі спамом та не спамом, а потім використовують теорему Баєса для обчислення ймовірності того, що електронне повідомлення є спамом чи ні [16].

3. Машинне навчання (Machine Learning): Використання алгоритмів машинного навчання, таких як нейронні мережі або дерева рішень, для класифікації повідомлень на спам і не спам. Машинне навчання може використовувати різні ознаки, включаючи текст, заголовок, адреси електронної пошти та інші.

4. Аналіз заголовків та метаданих (Header and Metadata Analysis): Виявлення спаму може включати аналіз інформації в заголовках повідомлень та метаданих. Наприклад, велика кількість адрес електронної пошти одного відправника може бути ознакою спаму.

5. Аналіз вмісту (Content Analysis): Технології виявлення спаму можуть аналізувати вміст повідомлення, включаючи текст, зображення та посилання. Деякі алгоритми можуть виявляти нелегітимний або фішинговий вміст.

Аналіз вмісту повідомлень електронної пошти за ключовими словами та виразами дозволяє виявити й вчасно запобігти витоку конфіденційної інформації, наявність забороненого змісту, зупинити розсилку SPAM. Аналіз тексту повідомлень виконуються на основі семантичного аналізу, враховуючи лексикограматичні граматичні конструкції слова [16].

6. Фільтрація спаму на основі списків (Blacklists and Whitelists): Використання списків IP-адрес, відправників чи доменів, які відомо належать спамерам (чорні списки) або легітимним відправникам (білі списки).

Повідомлення від адрес, що перебувають на чорних списках, може бути відфільтровано як спам.

7. Оцінка поведінки користувачів (User Behavior Analysis): Використання аналізу поведінки користувачів для виявлення надзвичайно активних або незвичайних дій, які можуть бути ознакою спаму.

8. Розпізнавання CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart): Вимагання від користувачів вирішити завдання CAPTCHA, які зазвичай легко вирішуються людиною, але складно або неможливо для комп'ютерів, щоб підтвердити, що вони не є спамерами. Ці технології можуть використовуватися окремо чи в поєднанні для ефективного виявлення та фільтрації спаму в різних типах комунікації, таких як електронна пошта, СМ, чати тощо.

Машинне навчання є потужним інструментом для виявлення спаму з численними перевагами. Воно автоматизує виявлення, зменшуючи необхідність в ручному аналізі. Моделі машинного навчання можуть швидко адаптуватися до нових форм спаму, забезпечуючи актуальність захисту.

Аналіз різноманітних даних, включаючи текст, зображення та інші, дозволяє ефективно виявляти спам у різних форматах. Постійне вдосконалення моделей дозволяє системам машинного навчання ефективно протистояти еволюції методів спамерів. Здатність враховувати контекст і специфіку різних платформ підвищує точність виявлення спаму. Використання різноманітних ознак дозволяє точно визначати спамові патерни.

В розроблюваному програмному методі для розпізнавання спам-повідомлень використовуються фільтрація спаму на основі ключових слів та методи машинного навчання, тож розглянемо їх детальніше.

2.2. Методи машинного навчання

Машинне навчання представляє собою галузь штучного інтелекту, яка дозволяє системам автоматично вчитися та покращувати свою продуктивність з досвідом.

Технологія машинного навчання, заснована на аналізі даних, своє коріння має ще в минулому столітті, коли розпочалася розробка перших програм для гри в шашки. За останні десятиліття загальний принцип залишається незмінним, але завдяки вибуховому росту обчислювальної потужності комп'ютерів, закономірності та прогнози, створювані цією технологією, стали значно більш складними. Розширилося також коло проблем і завдань, які вирішуються за допомогою машинного навчання.

Завдяки машинному навчанню програміст не зобов'язаний писати інструкції, що враховують всі можливі проблеми і містять всі варіанти рішення проблем. Замість цього в програму закладають алгоритм самостійного знаходження рішень шляхом комплексного використання статистичних даних, з яких виводяться закономірності і на основі яких робляться прогнози [17].

Існує декілька шляхів навчання алгоритму (рис. 2.2):

Рис. 2.2. Методи навчання алгоритмів машинного навчання

Навчання з учителем. У цьому типі машинного навчання модель навчається на основі вхідних даних та відповідних міток (класифікація) чи значень (регресія). Для розробки модуля пошуку пропагандистських повідомлень можливо використання класифікаційних алгоритмів для визначення, чи належить текст до категорії пропаганди.

Навчання без учителя. У цьому випадку модель навчається на невідмічених даних, без міток чи значень. Оскільки програмний модуль, що буде розроблятися в даній роботі має шукати коментарі із специфічним вмістом цей метод не підходить для використання.

Навчання з підкріпленням. Цей метод передбачає взаємодію моделі з оточенням, де модель отримує винагороду чи штраф в залежності від дій, які вона здійснює. Цей метод навчання використовується більше для оптимізації роботи вже існуючих методів, але не для пошуку специфічних даних.

Методи машинного навчання з учителем відзначаються декількома важливими перевагами. Вони забезпечують точні результати, оскільки моделі тренуються на парах вхідних та відповідних вихідних даних. Ці методи дозволяють моделям вивчати складні структури в даних і розпізнавати взаємозв'язки між різними параметрами.

Отже, обраний метод навчання - *навчання з учителем*. Машинне навчання з учителем є однією з основних гілок машинного навчання і використовується тоді, коли у наборі даних є чіткі мітки або відповіді, які модель намагається вивчити. Основна ідея полягає в тому, щоб навчити алгоритм передбачати відповіді на нових, раніше небачених даних на основі вивчених закономірностей.

Машинне навчання з учителем використовується для розв'язання різноманітних завдань, від класифікації тексту до прогнозування цін на фондовому ринку. Ці методи дозволяють моделям ефективно вивчати складні залежності та робити точні прогнози на основі навчальних даних.

Для того щоб почати навчання методу необхідно спочатку надати йому набір даних для тренування. В нашому випадку ми шукаємо коментарі з пропагандистським вмістом, для цього ми створюємо набір коментарів з позначкою, яка визначає чи містить цей коментар пропаганду. Після навчання метод вміє розпізнавати коментарі. Таким чином чим більший набір навчальних даних - тим вища ефективність методу.

Основні методи машинного навчання машинного з учителем поділяються на класифікацію та регресію.

Регресія передбачає числовий вихід для певного вхідного значення. Модель навчається знаходити функціональні залежності між ознаками та вихідним значенням.

Класифікація визначає, до якої категорії чи класу належить об'єкт. Модель навчається на основі прикладів з відомими класами. Для вирішення цілей розроблюваного програмного модуля класифікація як метод підходить більше, оскільки нам треба відомі класи для класифікації чи є коментар пропагандою. Метою класифікації є створення алгоритму, який може автоматично призначати новим об'єктам один з кількох передбачених класів. Нижче представлений алгоритм процесу класифікації у машинному навчанні (рис. 2.3):

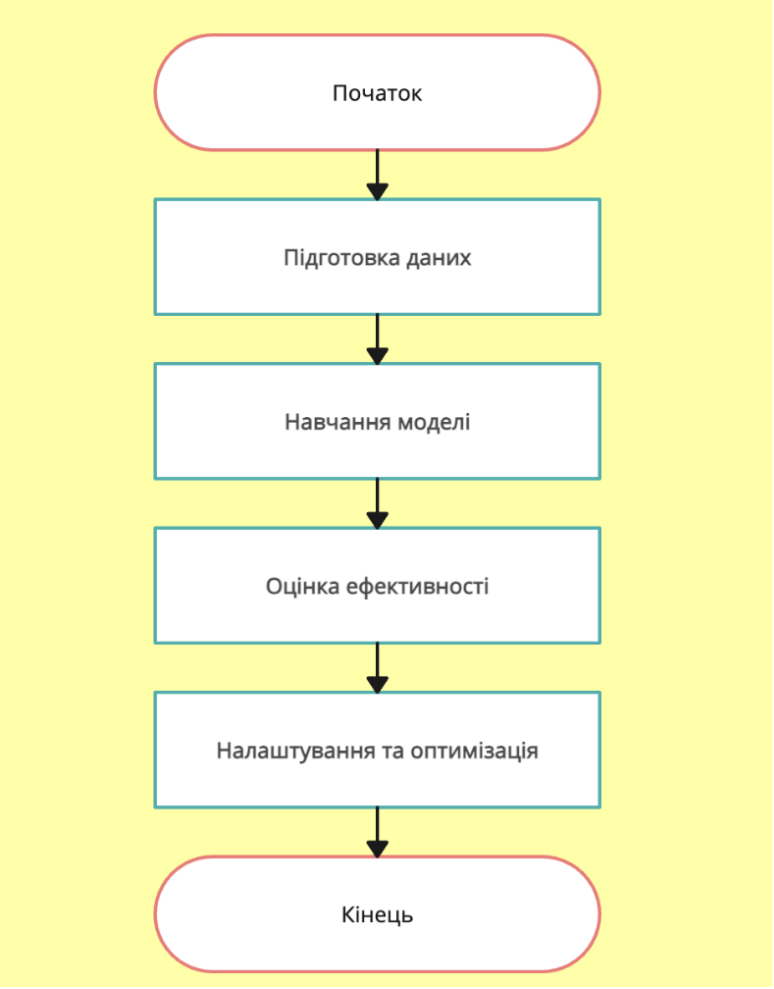


Рис. 2.3. Алгоритм процесу класифікації в машинному навчанні

Підготовка даних. Вибір ознак, що описують об'єкти. Розділення навчального та тестового наборів. Якісні та правильно підготовлені дані допомагають моделі працювати ефективно та точно.

Навчання моделі - це процес, під час якого модель "вчиться" на основі тренувальних даних і адаптується до внутрішніх закономірностей цих даних. Тренування моделі на навчальному наборі.

Оцінка ефективності є важливим етапом, щоб зрозуміти, наскільки добре модель виконує своє завдання. Використання тестового набору для визначення точності та інших метрик ефективності моделі.

Налаштування та оптимізація Перегляд та оптимізація моделі для поліпшення її результатів.

Основні методи класифікації (рис. 2.4):

Рис 2.4. Основні методи кластифікації

Для вирішення поставленого завдання важливо обрати методи, які будуть найбільш ефективні для пошуку коментарів пропагандистського вмісту. Для вибору найбільш ефективних методів вирішення поставленого завдання необхідно порівняти методи класифікації.

Таблиця 2.1

Порівняння методів класифікації

Назва методу	**Переваги**	**Недоліки**
Логістична регресія	Простота та інтерпретованість Можливість управління надмірністю Широке застосування	Обмежена здатність моделі Відсутність обробки втрат даних Вразливість до високо взаємокорельованих ознак
Наївний Баєсівський класифікатор	Простота та швидкість навчання Ефективність при невеликих обсягах даних	Сильне припущення про незалежність ознак Неефективність при великому обсязі даних Врахування тільки локальної інформації
Метод опорних векторів	Ефективність у високовимірних просторах Добра працездатність при обробці рідкісних даних Модель може уникнути перенавчання	Високі обчислювальні витрати Чутливість до шуму та викидів Складність інтерпретації результатів
k-найближчих сусідів	Простота та інтуїтивність Добре пристосовується до змін в даних Застосовується до різноманітних типів даних	Потребує стабільних та відмінно розділених класів Важливість вірної настройки параметру k

	Не обов’язковий тренувальний етап	
Дерево рішень	Простота і інтерпретованість Висока ефективність для простих завдань Автоматичне врахування важливості ознак Можливість роботи з пропущеними значеннями	Тенденція до перенавчання Чутливість до шуму
Нейронні мережі	Здатність вивчати складні залежності Можливість автоматичного визначення ознак Застосування до різних завдань	Потреба в великій кількості даних Велика обчислювальна складність Високі вимоги до обладнання та ресурсів

Програмний модуль, що буде розроблятися в даній роботі має навчатись з невеликої кількості даних, пристосовуватись до змін даних та не потребувати великих обчислювальних потужностей для спрощення його використання для користувачів Facebook.

Оскільки перевагами *методу k-найближчих сусідів* є те, що він добре пристосовується до змін в даних та його універсальність використання для різних типів даних він був обраний як перший метод. Недоліки методу не мають значного впливу у випадку, де дані мають бути чітко класифіковані, як у випадку з пошуком пропагандистських коментарів.

Оскільки перевагами МДР є простота, інтерпретованість та автоматичне врахування важливості ознак цей метод добре підходить для поставленого завдання, оскільки інтерпретованість дозволяє зберегти швидкість виконання розрахунків зі збереженістю ефективності роботи, а автоматичне врахування важливості ознак підвищує точність класифікації.

2.3. Метод k-найближчих сусідів

Метод найближчого сусіда входить до числа алгоритмів машинного навчання та базується на порівнянні елементів множин та їх класифікації з урахуванням ступеня схожості між різними об'єктами. Починаючи з початкової точки, цей алгоритм обстежує кожен найближчий об'єкт, послідовно відвідуючи їх всі.

У медицині цей метод широко використовується для вибору найоптимальнішого діагнозу, у плануванні будівництва - для визначення найвигіднішого місця розташування будівлі громадського призначення. Також його використовують у сфері транспорту, наприклад, при замовленні таксі, де застосовується для визначення найближчої точки до клієнта.

Загальний алгоритм роботи МНС представлений на рисунку 2.5:

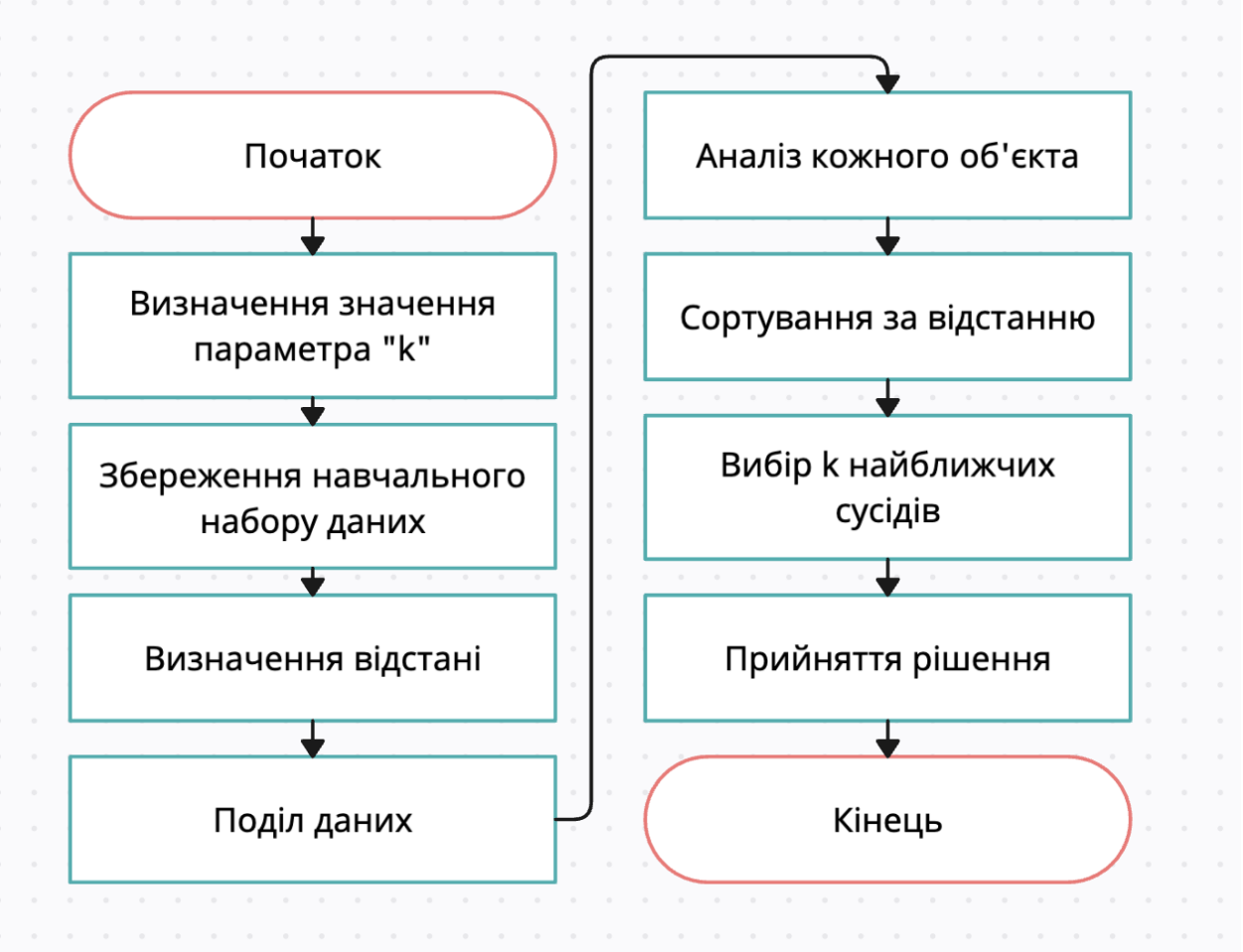


Рис 2.5. Загальний алгоритм роботи МНС

Для оптимальної роботи цього методу важливо правильно обрати значення "k". Значення "k" представляє собою кількість найближчих сусідів, яку ви вибираєте для врахування при класифікації або передбаченні значень для нового об'єкта.

Важливо визначити оптимальне значення "k" для конкретної задачі. Велике значення "k" може зробити алгоритм менш чутливим до шуму, але виносити усю вибірку може призвести до втрати чутливості до локальних патернів. З іншого боку, мале значення "k" може зробити модель чутливішою до випадкових аномалій в даних.

Якщо прийняти k = 1, то алгоритм втратить узагальнюючу здатність (тобто здатність видавати правильний результат для даних, що не зустрічалися раніше в алгоритмі), оскільки новому запису буде присвоєний

клас, близький до нього [19]. Якщо обрати парне значення "k" в алгоритмі, це може призвести до ситуації, де виникає можливість "нічиєї" при прийнятті рішення. Це може вплинути на продуктивність та зменшити ефективність алгоритму.

Наступним кроком є збереження навчального набору даних, який складається з об'єктів і відомих класів. Це робиться для навчання алгоритму класифікувати дані.

Наступний крок - визначення відстані. Вибір метрики відстані (наприклад, евклідова відстань), яка буде використовуватися для вимірювання відстані між об'єктами в просторі ознак.

Наступний крок - поділ даних. Поділ нових даних на вибірку для навчання та тестування, щоб оцінити ефективність моделі.

Наступний крок - аналіз кожного об'єкта. Для кожного об'єкта тестового набору (або для нового об'єкта, який потрібно класифікувати) обчислити відстань від нього до всіх об'єктів навчального набору.

Наступним кроком є сортування за відстанню. Сортування об'єктів навчального набору за зростанням відстаней до тестового об'єкта.

Далі вибір k найближчих сусідів. Вибір перших k об'єктів, які мають найменші відстані до тестового об'єкта.

Фінальним кроком є прийняття рішення. Визначення класу тестового об'єкта на основі класів його k найближчих сусідів (наприклад, голосування більшості).

Ефективність алгоритму залежить від правильного вибору значень "k" і метрики відстані, а також від властивостей самого набору даних. За умови правильно підібраних значень та даних можна досягти високої швидкості та ефективності алгоритму.

2.4. Метод дерево рішень

ММН *"Дерево рішень"* — це алгоритм, який використовується для прийняття рішень на основі структурованого дерева. У цьому дереві кожен вузол представляє рішення або тестує певну ознаку, а кожний гілочка з вузла вказує на можливий результат цього тесту.

Власне, саме дерево рішень - це метод представлення вирішальних правил в ієрархічній структурі, що складається з елементів двох типів - вузлів (node) і листя (leaf). У вузлах знаходяться вирішальні правила і робиться перевірка відповідності прикладів цьому правилу по якому-небудь атрибуту навчальної множини. У простому випадку, в результаті перевірки, безліч прикладів, що потрапили у вузол, розбивається на дві підмножини, в одне з 10 яких потрапляють приклади, що задовольняють правилу, а в інше — що не задовольняють [20].

За Брейманом дерева рішень, які використовуються у обробці даних, поділяються на два основних типи:

- Класифікаційні дерева, де передбачуваний результат є класом, до якого відносяться дані.
- Регресійні дерева, де передбачуваний результат можна розглядати як дійсне число (наприклад, ціна на будинок або тривалість перебування пацієнта в лікарні).

Загальний алгоритм роботи МДР представлений на рисунку 2.6:

Рис 2.6. Загальний алгоритм роботи МДР

Для початку необхідно визначити атрибути для розбиття, тобто обрати змінну, яка найкраще розділить дані на підгрупи.

Далі йде створення вузла дерева - вузла, який представляє обране розбиття. У вузлі зберігається інформація про обраний атрибут та поріг для розбиття

Наступний кроком є розбиття даних. Розділення даних на підгрупи відповідно до обраного атрибута та порогу. Кожна підгрупа відповідає одній гілці дерева.

Далі необхідно запустити рекурсивний процес, тобто застосувати попередні кроки до кожної підгрупи. Тобто, для кожної нової підгрупи обрати новий атрибут для розбиття та повторювати процес.

Наступним необхідним кроком є зупинка. Необхідно визначити умови зупинки для рекурсії, такі як досягнення максимальної глибини дерева, мінімальна кількість прикладів у вузлі або інші критерії.

Фінальним кроком є отримання мітки, тобто в нашому випадку - отримання результату. Це відбувається, коли досягається умова зупинки алгоритму.

Ефективність алгоритму дерева рішень залежить від точного вибору атрибутів, оптимальної глибини дерева, правильних умов зупинки, достатньої кількості навчальних прикладів.

2.5. Обробка природної мови

Для підвищення ефективності роботи методів машинного навчання важливо використовувати *обробку первинної мови*. Вихідні дані для тренування методів можуть бути неповними, занадто спеціалізованими, ОПМ дозволяє отримати більш зрозумілий для навчання набір даних.

Успішне використання більшості методів машинного навчання було досягнуто завдяки наявності репрезентативних даних, вхідних ознак, та оптимізації ваг, для підвищення точності фінального передбачення [22].

Розглянемо поширені методи ОПМ (рис. 2.7):

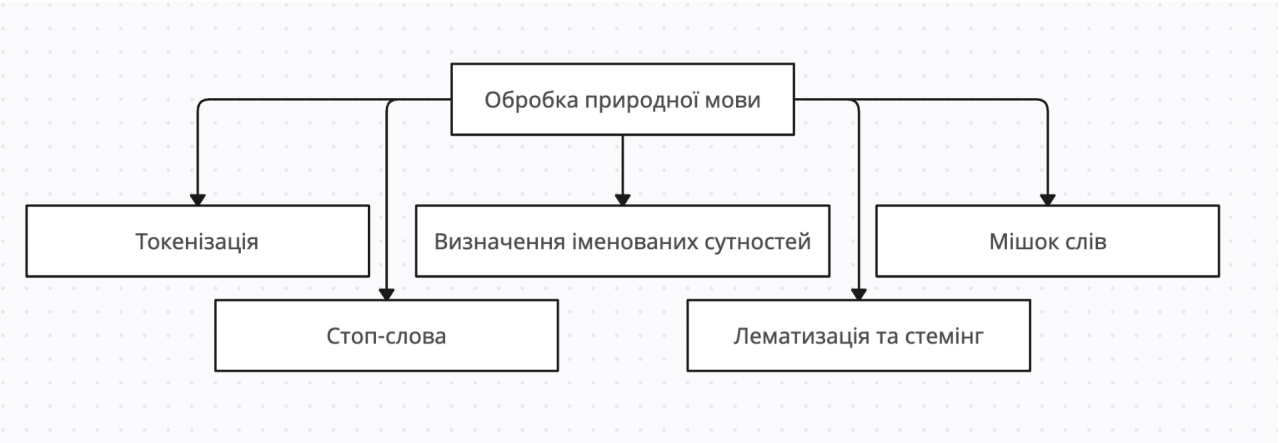


Рис 2.7. Методи ОПМ

1. Токенізація. Розбиття тексту на токени (одиниці, такі як слова або речення), щоб визначити основні елементи тексту. Розбиття тексту на токени полегшує подальший аналіз та обробку даних. Великий текст стає більш структурованим і зручним для подальших операцій, таких як векторизація для застосування алгоритмів машинного навчання.

Приклад:

"Токенізація - це важливий етап обробки природної мови." =>

["Токенізація", "-", "це", "важливий", "етап", "обробки", "природної", "мови", "."]

2. Мішок слів. Цей підхід в ОПМ перетворює текстові дані в множину слів і їхніх кількостей без врахування порядку слів у тексті чи їхніх взаємозв'язків.

Мішок слів використовується для класифікації великих текстів за їх змістом. Наприклад, визначення теми новинної статті, виявлення сентименту (позитивного, негативного, нейтрального) у відгуках або визначення категорії текстового документа.

Мішок слів може використовуватись як для окремо одного речення тексту, тобто для знаходження частоти появи певних слів у ньому, так і для порівняння двох речень, де певні вектори показуватимуть частоту появи слів з одного речення в іншому

Приклад 1:

"Це простий приклад алгоритму мішка слів." =>

Словник: {Це, простий, приклад, алгоритму, мішка, слів}

Вектор тексту: [1, 1, 1, 1, 1, 1], де 1 - частота появи слова в тексті

Приклад 2:

Текст 1: "Це простий приклад алгоритму мішка слів."

Текст 2: "Алгоритм мішка слів використовується для обробки текстової інформації." (Тут нулі вказують на те, що слова "використовується", "для", "обробки", "текстової" і "інформації" відсутні у тексті 1)

Вектор тексту 1: [1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0] (

Вектор тексту 2: [0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1] (Тут нулі вказують на те, що слова "Це", "простий", "приклад", "алгоритму" відсутні у тексті 2)

3. Стоп-слова - це слова, які зазвичай вважаються незначущими або малозначущими для розуміння сутності тексту. Метод заключається у тому, що ці слова виключаються або відкидаються під час обробки текстових даних. Це робиться для зменшення обсягу даних та покращення продуктивності алгоритмів, так як стоп-слова часто зустрічаються в тексті дуже часто, але не несуть важливої інформації про контекст.

Приклад:

"Аналіз сентименту у соціальних мережах є важливим завданням." =>

"Аналіз сентименту соціальних мережах важливим завданням."

4. Визначення іменованих сутностей. Метод спрямований на виявлення та класифікацію конкретних сутностей або об'єктів в тексті, таких як імена людей, місця, дати, організації, числові значення та інші важливі елементи. Цей процес допомагає системам ОПМ розрізняти та розуміти ключові елементи тексту, що дозволяє здійснювати більш комплексний аналіз інформації.

Приклад:

"На виставці мистецтва у Києві, яка триватиме до 15 грудня, представлені картини відомого художника Тараса Шевченка." =>

"На виставці мистецтва у [Місце: Київ], яка триватиме до [Дата: 15 грудня], представлені картини відомого художника [Особа: Тараса Шевченка]."

5. Лематизація та стемінг спрямовані на зведення слова до його базової форми. Це допомагає враховувати різні форми слова як одне слово, щоб полегшити аналіз тексту та зменшити кількість унікальних токенів, що полегшує аналіз тексту та розуміння його суті.

Приклад: "Моїй сестрі подобається читати книги кожного вечора." =>

"Моїй" - лема "мій"

"сестрі" - лема "сестра"

"подобається" - лема "подобатися"

"читати" - лема "читати"

"книги" - лема "книга"

"кожного" - лема "кожен"

"вечора" - лема "вечір"

Стемінг (відсічення афіксів у слова):

Слово: "бігають" Стем: "біг"

2.6. Аналіз існуючих методів боротьби зі спамом в соціальній мережі Facebook

Таблиця 2.2

Існуючі методи боротьби зі спамом в соціальній мережі Facebook

Метод	Характеристика методу	Неділоки методу

Використання спільноти користувачів	Facebook включає можливості звітування та позначення контенту як небажаного користувачами.	Найбільша проблема цього методу - недостатня швидкість рагування. Поки скарга користувача буде розглянута пропагандистський контент може побачити велика кількість користувачів.
Аналіз поведінки	Алгоритми аналізують поведінку користувачів, зокрема швидкість розміщення повідомлень, обсяг інтеракцій, інтервали часу між діями та інші аспекти, щоб визначити потенційно спамовий аккаунт.	Зміни в алгоритмах та політиках Facebook можуть призводити до коливань в результативності систем боротьби зі спамом. Алгоритми, створенні для поширення реклами у Facebook можуть випадково приховувати спам.
Використання CAPTCHA та інших заходів безпеки	Facebook впроваджує різноманітні інструменти безпеки, такі як CAPTCHA, для перевірки, чи користувач є людиною, а не автоматизованим ботом, що розповсюджує спам.	Не працює, якщо спам поширюють люли, а не автоматизовані боти
Аналіз взаємодії користувачів	Facebook спостерігає за взаємодією користувачів з контентом, щоб визначити, чи є певний вміст небажаним або спамом. Наприклад, велика кількість непозначених як спам скарг або відмов від підписки може вказувати на небажаний контент.	Деякі користувачі можуть спеціально взаємодіяти з спамовим контентом, щоб обходити систему виявлення.

Найголовнішою причиною неефективності існуючих методів боротьби зі спамом, який несе в собі пропагандистський вміст є специфіка наративів таких повідомлень.

Часто такі повідомлення не є спамом в його поширеному вигляді, не містять підозрілих посилань, тож розпізнати їх важче та для цього потрібен більш специфічний інструмент, такий як програмний модуль, що буде розроблено.

Також важливо проаналізувати існуючі рішення для пошуку спаму або класифікації даних в соціальних мережах. Для пошуку спам-повідомлень саме у соціальній мережі Facebook програмних рішень не було знайдено, оскільки їхня політика конфіденційності не дозволяє поширеним антиспам-додаткам зчитувати інформацію про коментарі або повідомлення користувача без спеціального дозволу. Тож до порівняння взяті аналоги, що можуть застосовуватись для соціальних мереж, зокрема Twitter.

Таблиця 2.3

Існуючі сервіси для пошуку спаму в соціальних мережах

Програмне ріщення	Використані методи машинного навчання	Види спаму, який виявляється	Ефективність роботи	Необхідна кількість навчальних даних
Кластеризація спам-доменів методами машинного навчання [24]	Класифікатор випадкового лісу	Виявлення спам-доменів	92%	513000 тисяч доменів
Виявлення зловмисних повідомлень в соціальній мережі Twitter на основі URL-посилань [25]	ММН Випадковий ліс	Фішинг, рекламний спам	86%	182000 тисячі твітів

Проєктування інформаційної системи для виявлення і запобігання масової дезінформації в Twitter [26]	Моделі глибокого навчання	Дезінформація	71%	4000 рядків

Всі програмні рішення описані в таблиці 2.3 використовуються для класифікації певних даних з СМ та використовують ММН.

Всі описані в таблиці 2.3 програмні рішення призначені для пошуку спаму в соціальних мережах та використовують для цього різні методи машинного навчання. Проте всі вони потребують величезних об’ємів тренувальних даних, що робить їх перенавчання дуже складним процесом, що є значним недоліком, оскільки для пошуку саме пропаганди важливо мати змогу постійно оновлювати дані за якими відбуватиметься класифікація.

Отже, програмне рішення для пошуку пропагандистських спам-коментарів у Facebook є актуальною задачею. Особливо такого який буде навчатись з невеликої кількості даних, оскільки спам постійно видозмінюється і простота перенавчання є перевагою для актуальності використання програмного рішення.

2.7. Висновки до розділу 2

Виявлення небажаних повідомлень є ключовою задачею для забезпечення безпеки та комфорту користувачів в інтернет-середовищі. Існує багато технологій та методів для виявлення небажаної інформації в різних контекстах, таких як електронна пошта, текстові повідомлення, коментарі в СМ та інші.

Машинне навчання ефективно виявляє спам завдяки автоматичному навчанню, здатності аналізувати великі обсяги даних, постійному вдосконаленню та здатності розпізнавати складні патерни. Цк дозволяє системі ефективно адаптуватися до нових видів спаму та забезпечувати безпеку користувачів в інтернеті, тож методи машинного навчання були випрані як технологія, що буде використовуватись у програмному модулі, який буде розроблено.

Було проаналізовано види методів машинного навчання з метою обрання найбільш підходящого для поставлених завдань. Методи машинного навчання з учителем мають кілька ключових переваг. Вони гарантують високу точність через тренування моделей на парах вхідних та відповідних вихідних даних. Ці методи дають можливість моделям освоювати складні структури в даних та розпізнавати взаємозв'язки між різними параметрами.

Для пошуку пропагандистських коментарів використання вже існуючих тренувальних даних - оптимальне рішення, оскільки це значно підвищить ефективність програмного модуля, що буде розроблено.

МНС (K-найближчих сусідів) визначає клас об'єкта на основі класів його найближчих сусідів у вхідному просторі. Його переваги включають високу простоту та інтуїтивність, роблячи його легким у реалізації та розумінні.

Він ефективний у виявленні шаблонів та залежностей у даних, особливо в тих випадках, коли кількість навчальних даних середньої величини. Метод дозволяє враховувати контекст та адаптуватися до змін у вхідних даних без переініціалізації моделі. Також цей метод добре підходить для роботи з текстовими даними, тож його було обрано для використання в програмному модулі, що буде розроблено.

Метод дерева рішень має кілька переваг, серед яких важливо відзначити його легку інтерпретованість, що дозволяє зрозуміти та пояснити прийняті рішення.

Дерева рішень можуть моделювати складні залежності та взаємозв'язки у даних, що робить їх ефективними для різних завдань. Цей метод дозволяє автоматично визначати важливість параметрів, спрощуючи відбір ознак, тож цей метод було обрано як другий для використання в програмному модулі, що буде розроблюбватись.

Для підвищення ефективності роботи методів машинного навчання можна використати обробку природного тексту. Така обробка допомагає краще шукати залежності в текстових даних, які опрацьовуватимуться в програмному модулі, що буде розроблено.

Після аналізу та порівняння методів машинного навчання вирішено обрати для програмного модуля, що буде розроблятись МНС та дерева рішень як найбільш оптимальні для поставленої задачі пошуку коментарів з пропагандистським вмістом. Для підвищення ефективності роботи методів будуть використані методи ОПМ.

В наступному розділі кваліфікаційної роботи будуть детально описані алгоритми ММН та методів ОПМ для їхньої ефективної практичної реалізації.

РОЗДІЛ 3.

ПРАКТИЧНЕ ЗАСТОСУВАННЯ МЕТОДІВ НАЙБЛИЖЧОГО СУСІДА ТА ДЕРЕВА РІШЕНЬ. АЛГОРИТМ РОБОТИ РОЗРОБЛЕНОГО ПРОГРАМНОГО МОДУЛЯ.

3.1. Підготовка навчальних даних, ключових слів для пошуку

Однією з найважливіших складових оптимальної роботи методів машинного навчання є правильна підготовка навчальних даних. Крім навчальних даних для розробленого програмного модуля використовуються ключові слова від підбору яких також залежить ефективність створеного рішення.

Для навчання МНС необхідно створити набір тренувальних даних. Кожен навчальний приклад повинен включати вхідні дані, які описують об'єкт чи ситуацію, яку слід класифікувати. В нашому випадку цим об’єктом є коментар.

Кожен навчальний об’єкт повинен містити класифікаційну мітку. Це інформація, яку алгоритм спробує вивчити та застосувати при класифікації нових прикладів. В нашому випадку це мітки про те, чи є даний коментар спамом.

Як тренувальні дані були використані реальні коментарі з СМ Facebook. Коментарі мають позначки TRUE/FALSE, які позначають чи є даний коментар спамом (рис. 3.1).

```
commend_id,date,message,bot
1,5/14/2023,8 лет молчали!,TRUE
2,5/14/2023,За Донбасс,TRUE
3,5/14/2023,8 лет молчали!,TRUE
4,5/14/2023,Потрібно обов'язково придбати,FALSE
5,5/13/2023,"Нагодуйте, тортика з чаєм дайте.",FALSE
6,5/12/2023,Слава захисникам,FALSE
7,5/12/2023,Бандери,TRUE
8,5/14/2023,Молодці!!!!,FALSE
9,5/14/2023,Чего ж вы 8 лет молчали???,TRUE
10,5/14/2023,"Найкращі,ми Вами пишаємося і щиро Вам Дякуємо, нехай Господь Бог Вас Усіх і Кожного Оберігає",FALSE
11,5/10/2023,Нелюди нечисть.....,FALSE
```

Рис. 3.1. Набір навчальних даних для алгоритмів машинного навчання

Набір ключових слів для пошуку пропагандистських коментарів було створено базуючись на дослідженні VoxUkraine щодо інформаційних атак пов'язаних з російським «Агентством інтернет-досліджень» [19] та словах, що поширені в коментарях з явно пропагандистським вмістом.

Для підвищення ефективності роботи програмного модуля ключові слова було здійснено обробку первинної мови, а саме видалення розділових знаків та лематизація (приведення слів до початкової форми) (рис 3.2).

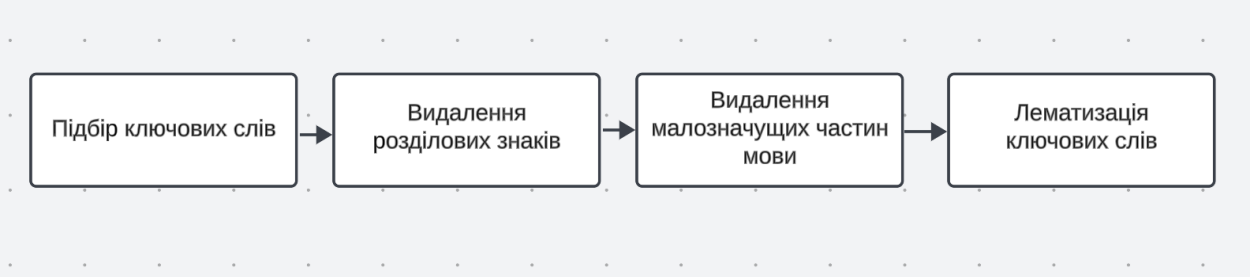


Рис. 3.2. Алгоритм обробки ключових слів

Видалення розділових знаків було виконано вбудованими інструментами Python. Лематизація виконується бібліотекою pymorphy2, яка містить український та російський словники, оскільки ми шукаємо пропагандистські коментарі обома мовами, ця бібліотека добре нам підходить.

Для видалення розділових знаків та незначущих частин мови текст розбивається на слова і потім видаляються всі символи, довжина яких менша ніж 3.

Бібліотека pymorphy2 для виконання лематизації використовує алгоритм знаходження нормальної форми, який працює найшвидше (береться перша форма в лексемі) - тому, наприклад, усі дієприкметники нормалізуються в інфінітиви. Бібліотека працює, порівнюючи вхідне слово зі словниками.

Обробка первинної мови в програмному коді (рис. 3.3):

```
def lemmatize(text):
    words = text.split() # розбиття тексту на слова
    res = list()
    for word in words:
        p = morph.parse(word)[0]
        res.append(p.normal_form) # створення нового масиву зі словами в початковій формі

    return res

for element in firstGroup:
    filteredFirstGroup = filteredFirstGroup + lemmatize(re.sub(r'[^\w\s]'," ",element)) # приведення слів до початкової форми

for element in filteredFirstGroup:
    if len(element)< 3:
      filteredFirstGroup.remove(element) # видалення розділових знаків та слів довжиною менше 3 букв
```

Рис 3.3. Функції лематизації та видалення розділових знаків

Таким чином ми отримуємо додатково до вже вказаних ключових слів масив слів у початковій формі (рис. 3.4).

```
До обробки: ['8 лет', 'За Донбасс', 'Бандер', 'Хохлостан', 'хохлы', 'нацисты', 'укроп', 'хахлов', 'МВФ'
, 'Донецк', 'Луганск', 'Россия-Украина', 'Госдеп', 'конфликт на Украине', 'МН-17']
Після обробки: ['8 лет', 'За Донбасс', 'Бандер', 'Хохлостан', 'хохлы', 'нацисты', 'укроп', 'хахлов', 'М
ВФ', 'Донецк', 'Луганск', 'Россия-Украина', 'Госдеп', 'конфликт на Украине', 'МН-17', 'год', 'донбасс',
 'бандёр', 'хохлостан', 'хохол', 'нацист', 'укроп', 'хахлов', 'мвф', 'донецк', 'луганск', 'россия', 'ук
раина', 'госдеп', 'конфликт', 'украина', '17']
```

Рис 3.4. Приклад ключових слів до обробки та після

3.2. Алгоритм роботи методу найближчого сусіда в розробленому програмному модулі

Розроблений програмний модуль опрацьовує коментарі, що були отримані з СМ Facebook двома методами машинного навчання одним з яких є МНС (рис. 3.5).

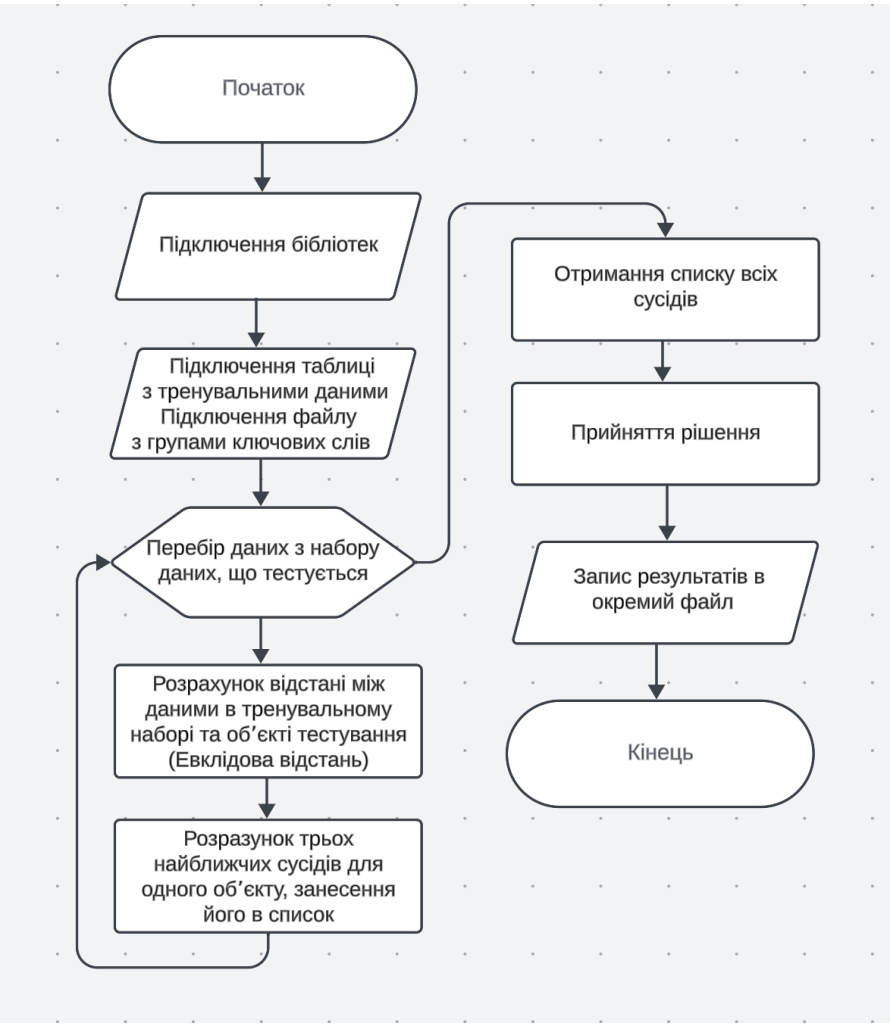


Рис 3.5. Блок-схема застосованого методу найближчого сусіда

Крок 1, вибір числа k. Важливою складовою оптимальної роботи МНС - правильний вибір числа k. Це число визначає кількість найближчих сусідів, які використовуються для прийняття рішення щодо класифікації нового об'єкта.

Алгоритм порівнює новий об'єкт з k найближчими об'єктами у навчальному наборі. Об'єкт класифікується на основі більшості класів серед його k найближчих сусідів.

Отже, важливо обрати k непарним числом, оскільки в випадку коли воно парне є можливість "нічиєї" при класифікації. Також не варто приймати k надто великим числом, оскільки тоді збільшується обчислювальна складність алгоритму. Детальний опис тестування різних значень числа k описано в розділі 4, як оптимальне значення цього числа обрано 3.

Отже, ми використовуємо три групи ключових слів, які зберігаються у файлі keywords (рис. 3.6).

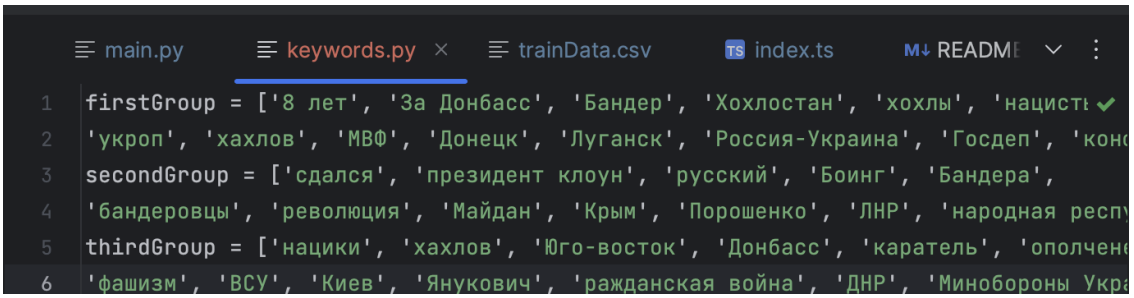


Рис 3.6. Файл з трьома групами ключових слів

Для пошуку сусідів дані з тренувального набору та коментарі розділяються на групи відповідно до ключових слів (рис. 3.7).

```
# Split the data into training and test data
def loadDataset():
    lst = []
    test = []

    for i in range(len(csv)):
        lst.append([isFirstGroup(csv[i][1]), isSecondGroup(csv[i][1]), isThirdGroup(csv[i][1]), parseBot(str(csv[i][2]))])

    for i in range(len(csvTest)):
        test.append([isFirstGroup(csvTest[i][0]), isSecondGroup(csvTest[i][0]), isThirdGroup(csvTest[i][0])])

    return lst, test
```

Рис 3.7. Розділення наборів даних на групи

Крок 2, розрахунок Евклідової відстані. перебираючи кожен зчитаний з CM Facebook коментар ми розраховуємо для нього Евклідову відстань до записів, які знаходяться в навчальному наборі даних.

$$D_E = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{3.1}$$

де n — кількість атрибутів.

Програмна реалізація розрахунку Евклідової відстані (рис. 3.8):

```
def euclideanDistance(instance1, instance2, length):
    distance = 0
    for x in range(length):
        distance += pow((instance1[x] - instance2[x]), 2)
    return math.sqrt(distance)
```

Рис 3.9. Розрахунок Евклідової відстані

Крок 3, вибір найближчих сусідів. Наступним кроком є вибір трьох найближчих сусідів зі всіх знайдених варіантів, порівнюються Евклідові відстані всіх знайдених для атрибуту сусідів (рис 3.8):

```
def getNeighbors(trainingSet, testInstance, k):
    distances = []
    length = len(testInstance)-1
    for x in range(len(trainingSet)):
        dist = euclideanDistance(testInstance, trainingSet[x], length)
        distances.append((trainingSet[x], dist))
    distances.sort(key = operator.itemgetter(1))
    neighbors = []
    for x in range(k):
        neighbors.append(distances[x][0])
    return neighbors
```

Рис. 3.9. Вибір трьох найближчих сусідів

Крок 4, прийняття рішення. Фінальним кроком алгоритму найближчого сусіда є прийняття рішення або ж зважене голосування, якщо більшість найближчих сусідів серед тренувальних даних є спам-повідомленнями буде прийняте рішення, що цей коментар є спамом, зважене голосування виконується за формулою:

$$votes(class) = \sum_{i=1}^{n} \frac{1}{d^2(X,Yi)} \qquad (3.2)$$

де

d^2 (X,Y) - квадрат відстаней від відомого запису до невідомого ,

n - кількість відомих записів, для яких розраховуються голоси.

Програмна реалізація зваженого голосування представлена на рис 3.10:

```
def getResponse(neighbors):
    # Creating a list with all the possible neighbors
    classVotes = {}
    for x in range(len(neighbors)):
        response = neighbors[x][-1]
        if response in classVotes:
            classVotes[response] += 1
        else:
            classVotes[response] = 1
    sortedVotes = sorted(classVotes.items(), key=operator.itemgetter(1), reverse=True)
    return sortedVotes[0][0]

def getAccuracy(testSet, predictions):
    correct = 0
    for x in range(len(testSet)):
        if testSet[x][-1] == predictions[x]:
            correct += 1
    return (correct/float(len(testSet))) * 100.0
```

Рис. 3.10. Прийняття рішення чи є коментар спамом

3.3. Алгоритм роботи методу "Дерево рішень" в розробленому програмному модулі.

Другим методом обробки коментарів в розробленому програмному модулі є МДР(рис. 3.11).

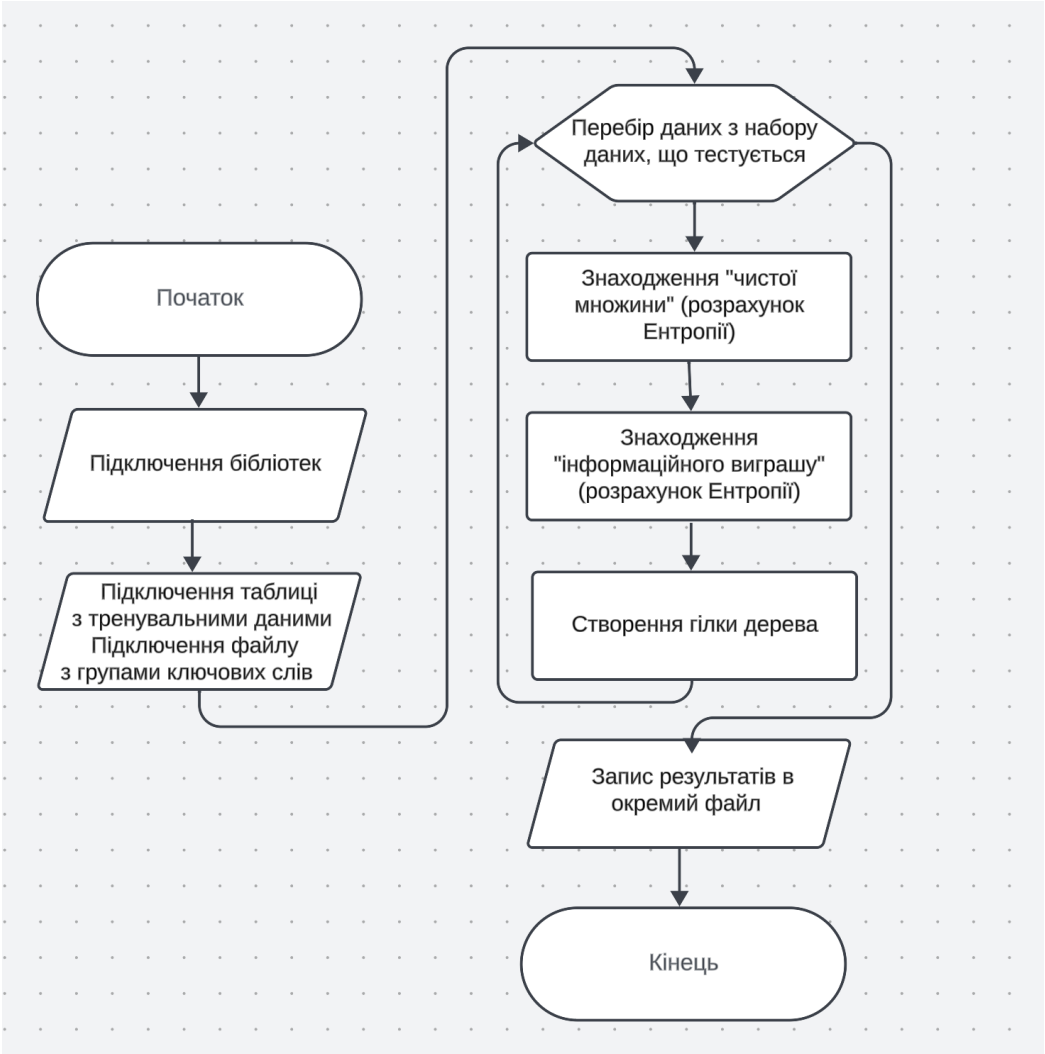


Рис 3.11. Блок-схема застосованого МДР

Для поділу дерева рішень на гілки зазвичай використовується один в двох ключових підходів: статистичний підхід та підхід приросту інформації. Статистичний підхід часто вимагає великої кількості навчальних даних, тож для розробленого програмного модуля було обрано підхід приросту інформації.

Перед виконанням безпосередньо алгоритму дерева рішень відбувається його навчання, для чого було використано бібліотеку sklearn. Це програмна бібліотека машинного навчання, створена для мови програмування Python.

Для тренування використовується команда цієї бібліотеки train_test_split, що розділяє дані в тренувальному наборі випадковим чином

для подальшого їх використання. Це випадкове розбиття може вносити незначні коливання результатів виконання цього методу машинного навчання, проте в цілому випадковість розбиття підвищує ефективність методу.

Ми вже маємо три групи ключових слів, описані в цьому розділі, які оброблені методами обробки природної мови.

Крок 1, створення гілки дерева. Ціллю розбиття даних на гілки є отримання чистої множини. Чистою множиною вважається число, Ентропія якого дорівнює нулю. Для розрахунку Ентропії використовується формула:

$$Entropy = -\sum_{i=0}^{n} \frac{N_i}{N} * log_2(\frac{N_i}{N}) \quad (3.3)$$

де

n — число класів в вихідній підмножини,

Ni — чисто прикладів і класса,

N — число прикладів у всій підмножині.

Програмна реалізація розрахунків Ентропії (рис. 3.12):

```
def entropy(rows):
    entries = class_counts(rows)
    avg_entropy = 0
    size = float(len(rows))
    for label in entries:
        prob = entries[label] / size
        avg_entropy = avg_entropy + (prob * math.log(prob, 2))
    return -1*avg_entropy
```

Рис 3.12. Розрахунок Ентропії

Крок 2, обрання параметру для наступного розбиття. Для вибору оптимального параметру використовується формула розрахунку інформаційного виграшу. Інформаційний виграш використовується для отримання максимальної інформації про класифікацію, ґрунтуючись на ентропії.

$$Gain = Entropy(\text{кореня}) - Entropy(\text{підмножини}) \quad (3.4)$$

Програмна реалізація розрахунків інформаційного виграшу (рис. 3.13):

```
def info_gain(left, right, current_uncertainty):
    p = float(len(left)) / (len(left) + len(right))

    return current_uncertainty - p * entropy(left) - (1 - p) * entropy(right)
```

Рис 3.12. Розрахунок інформаційного виграшу

Крок 3, побудова дерева рішень. Результатом виконання алгоритму дерево рішень є дерево з гілками, які позначені мітками TRUE/FALSE, тобто прийняте рішення щодо того чи є коментар спамом.

3.4. Висновки до розділу 3

Одним з найважливіших аспектів оптимальної роботи методів машинного навчання є підготовка вихідних та навчальних даних.

Підготовка навчальних даних включає в себе збір актуальних та підходящих для поставленого завдання даних. В розробленому програмному модулі використовуються реальні коментарі з СМ Facebook, завдяки чому навчання машинних методів ефективне та без потреби у перенавчанні готове до застосування.

В розробленому програмному модулі є такі вихідні дані як ключові слова за якими методи машинного навчання можуть первинно класифікувати коментарі. До ключових слів застосовані методи ОПМ, що значно поліпшує ефективність програмного модуля.

Методи ОПМ дозволяють методам машинного навчання краще інтерпретувати людську мову та розширити коло співпадіння з ключовими

словами. В розробленому програмному модулі застосовані лематизація та видалення розділових знаків, незначущих слів.

Для ефективної роботи методу машинного навчання МНС було проведено експеримент для вибору оптимального числа k, від якого безпосередньо залежить ефективність алгоритму.

Для обох використаних методів машинного навчання докладно досліджено та описано алгоритми роботи, що дозволило обрати такі способи їх використання при яких забезпечено високу ефективність роботи за короткий час та без використання значних обчислювальних потужностей, що дозволить використовувати розроблений програмний модуль пересічному користувачу СМ Facebook без необхідності використання додаткового апаратного забезпечення.

В наступному розділі буде розглянуто яким чином користувач може використовувати розроблений програмний модуль та приведено результати його тестування.

РОЗДІЛ 4.

ПРАКТИЧНЕ ЗАСТОСУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО МОДУЛЯ. ДЕМОНСТРАЦІЯ РЕЗУЛЬТАТІВ ТЕСТУВАННЯ

4.1. Алгоритм роботи розробленого програмного модуля

В розділі 3 даної кваліфікаційної роботи було описано два ММН, які використовувались при розробці програмного модуля. Одним з методів є МНС. Для його оптимальної роботи необхідно обрати значення для числа k.

Для прийняття остаточного рішення було проведено експеримент з його різними значеннями - 3, 5 та 7.

Було проведено тестування використовуючи 48 коментарів, половина з яких містять проросійські наративи, а половина - ні, відсоток точності було вирахувано відштовхуючись від того, який відсоток коментарів з пропагандистським вмістом виявлено.

Експеримент було проведено з однаковою кількістю навчальних даних, а саме 100 коментарів у навчальному наборі даних та 50 ключових слів розділених по кількості груп (3, 5, 7 відповідно). Тобто кількість ключових слів для експерименту залишається незмінною, але число k (кількість найближчих сусідів, які враховуються при визначенні класу чи значення для нового зразка) змінюється.

Таблиця 4.1

Результати експерименту з різними значеннями числа k

k	*Перший експеримент*	*Другий експеримент*	*Третій експеримент*	*Четвертий експеримент*	*П’ятий експеримент*

3	Швидкість: 231.03ms Точність: 73,6%	Швидкість: 219.96ms Точність: 73,6%	Швидкість: 221.14ms Точність: 73,6%	Швидкість: 225.16ms Точність: 73,6%	Швидкість: 220.14ms Точність: 73,6%
5	Швидкість: 367.18ms Точність: 74,9%	Швидкість: 372.38ms Точність: 74,9%	Швидкість: 369.93ms Точність: 74,9%	Швидкість: 373.13ms Точність: 74,9%	Швидкість: 370.69ms Точність: 74,9%
7	Швидкість: 676.82ms Точність: 76,9%	Швидкість: 680.14ms Точність: 76,9%	Швидкість: 688.93ms Точність: 76,9%	Швидкість: 690.14ms Точність: 76,9%	Швидкість: 687.19ms Точність: 76,9%

Середня швидкість виконання алгоритму найближчого сусіда на тестових даних з числом k = 3 - 223.49 ms, точність - 73,6%.

Швидкість виконання алгоритму найближчого сусіда на тестових даних з числом k = 5 - 370.72 ms, точність - 74,9%.

Швидкість виконання алгоритму найближчого сусіда на тестових даних з числом k = 7 - 684.64 ms, точність - 76,9%.

Як ми можемо побачити результат ефективності знаходження для п’яти експериментів лишається незмінним за умови якщо число k не змінюється, незначним чином коливається час виконання алгоритму в діапазоні різниці 10ms. Час коливання алгоритму може коливатись через запущені інші процеси на ПК, наприклад.

Представимо результати проведеного експерименту щодо числа k діаграмою (рис. 4.1):

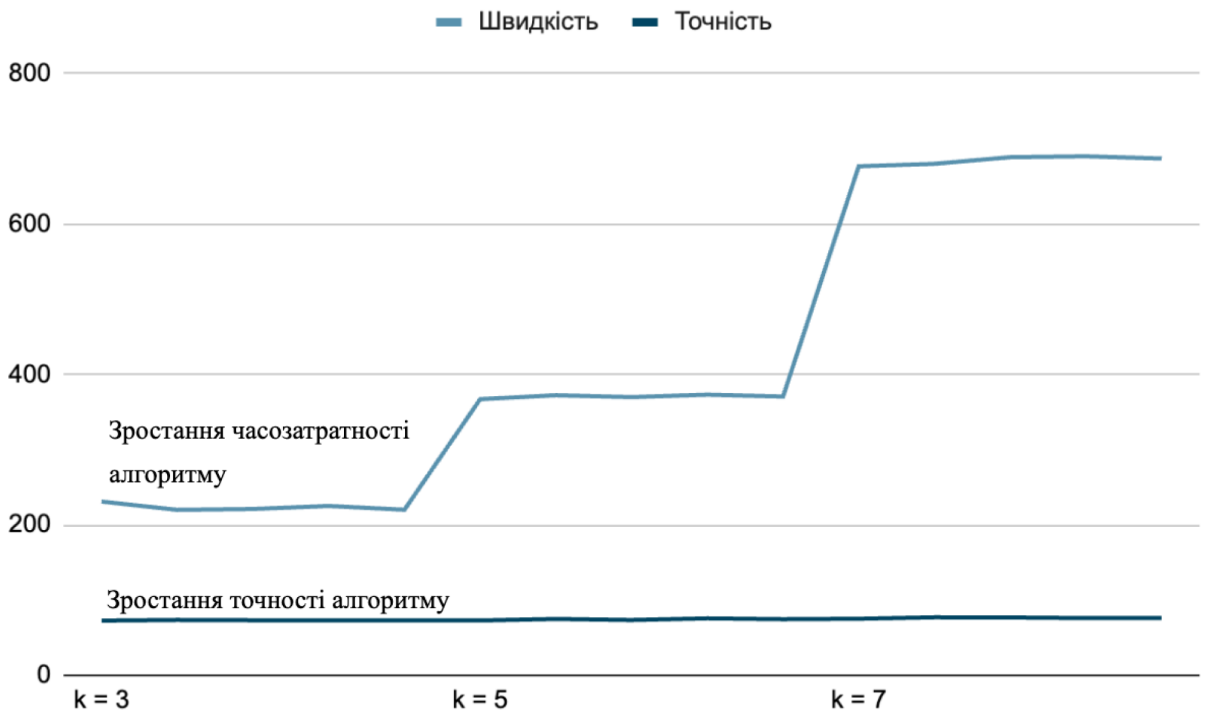


Рис 4.1. Результати експерименту з різним числом k

З діаграми видно, що зі збільшенням числа k час виконання алгоритму збільшується майже вдвічі в той час як його ефективність коливається менше ніж на 2 відсотка, тож було прийнято рішення прийняти за число k = 3.

В розділі 3 даної роботи було детально описано кожен з використаних ММН, тож розглянемо загальний алгоритм роботи розробленого програмного модуля (рис. 4.2).

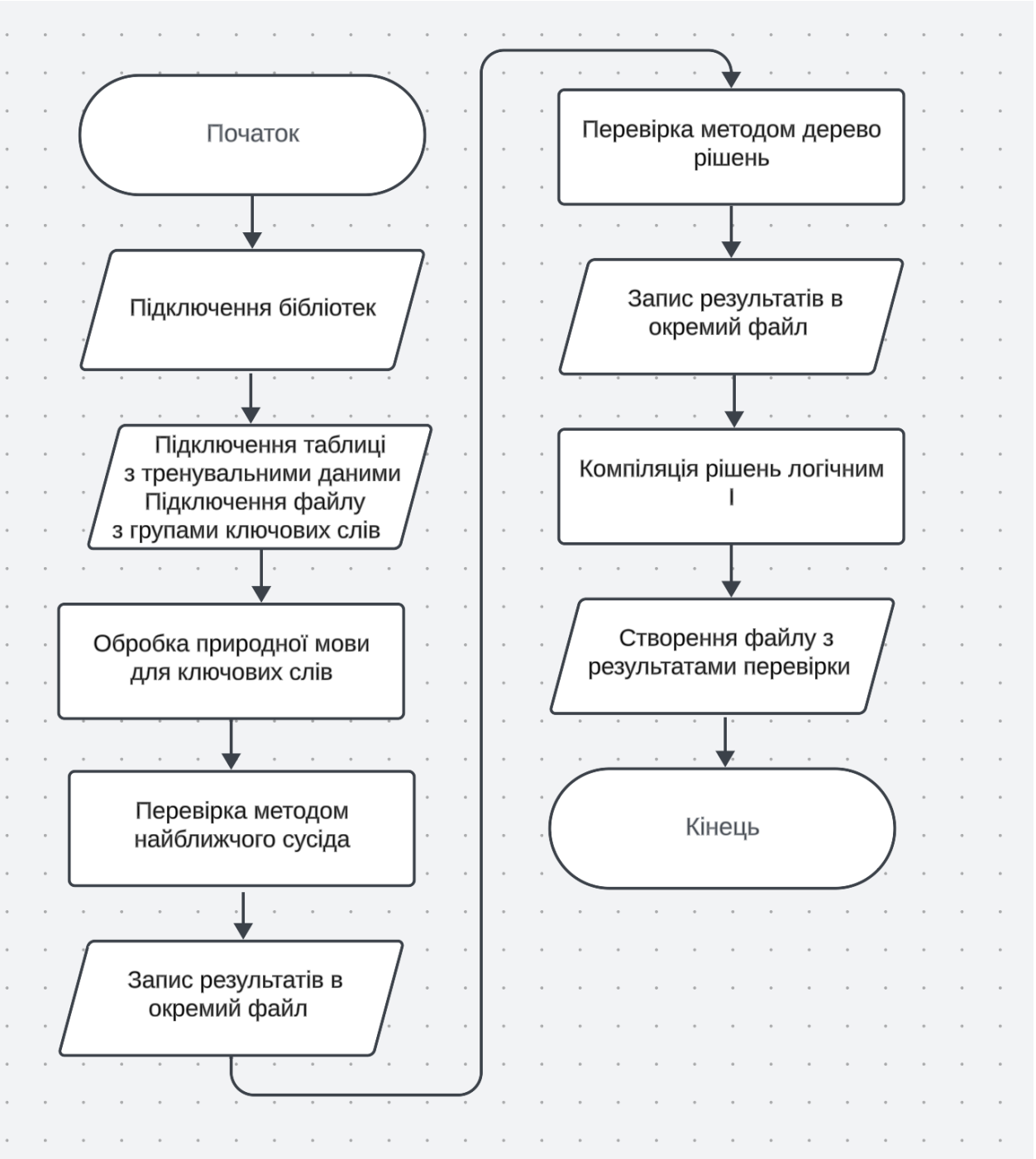


Рис. 4.2. Блок-схема розробленого програмного модуля

Результатом створеного програмного рішення є таблиця, що створена з використанням двох ММН. Результатом є скомпільований логічним І файл, що включає в себе знайдені обома ММН коментарі.

Програмний модуль написано двома мовами програмування. JavaScript використано для реалізації запиту до API Facebook та перетворення

отриманих коментарів на таблицю, яку можна ітерувати мовою програмування Python/

Python використаний для реалізації методів ММН. Python має велику кількість бібліотек, які було використано для оптимальної обробки та класифікації даних.

Використані бібліотеки:

- *datetime,* бібліотеку використано для розрахунку часу виконання класифікації програмним модулем;
- sklearn, використано для поділу вхідних даних на тренувальні дані та дані для тестування;
- *pymorphy2,* використано для лематизації навчальних даних;
- math, використано для розрахунків величин, які використовуються ММН.

4.2. Рекомендації щодо застосування програмного модуля

В цьому розділі кваліфікаційної роботи ми детально розібрали принцип та алгоритми роботи розробленого програмного модуля, перейдімо до його практичного застосування.

Політика конфіденційності Facebook передбачає, що користувач може зчитувати дані лише зі своїх сторінок, груп тощо. Також для того щоб отримати доступ до зчитування необхідно отримати спеціальний токен доступу.

Для отримання токену доступу необхідно пройти перевірку, надіславши фотографію свого паспорту команді Meta. Після чого можливо отримати токен доступу (рис. 4.3):

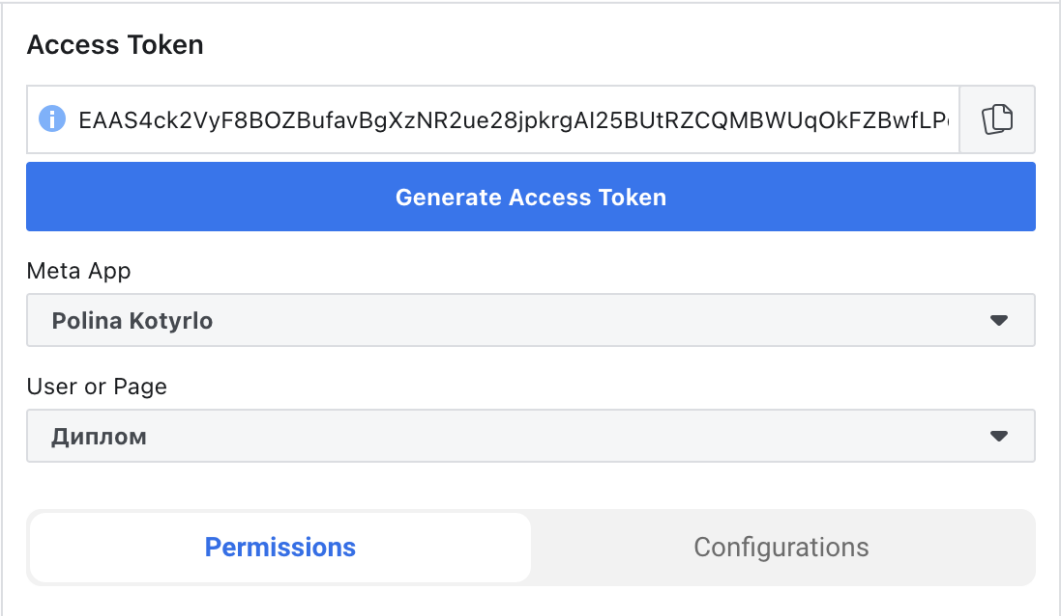


Рис. 4.3. Згенерований токен доступу

Для коректної роботи програмного модуля важливо оновлювати токен кожних 15 хвилин. Він автоматично оновлюється на веб-сайті https://developers.facebook.com. Отриманий токен доступу ми додаємо у файл .env, який знаходиться в кореневій папці програмного модуля (рис. 4.4).

Рис. 4.4. Файл зі згенерованим токеном доступу

З актуальним токеном доступу наш програмний модуль готовий до використання, для його запуску достатньо запустити файл start.sh, який знаходиться в кореневій папці розробленого програмного модуля, до виконання (рис. 4.5).

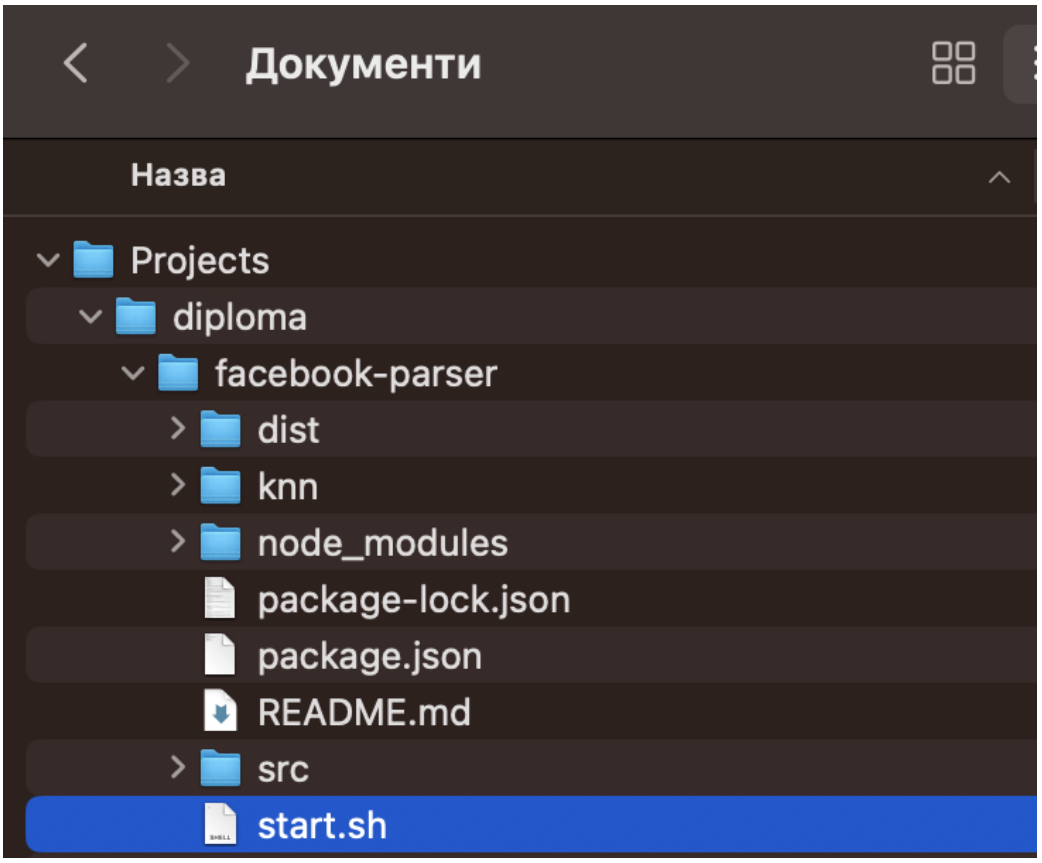


Рис. 4.5. Файл запуску програмного модуля start.sh

Як результат виконання розробленого програмного модуля користувач отримує архів з результатами (рис. 4.6).

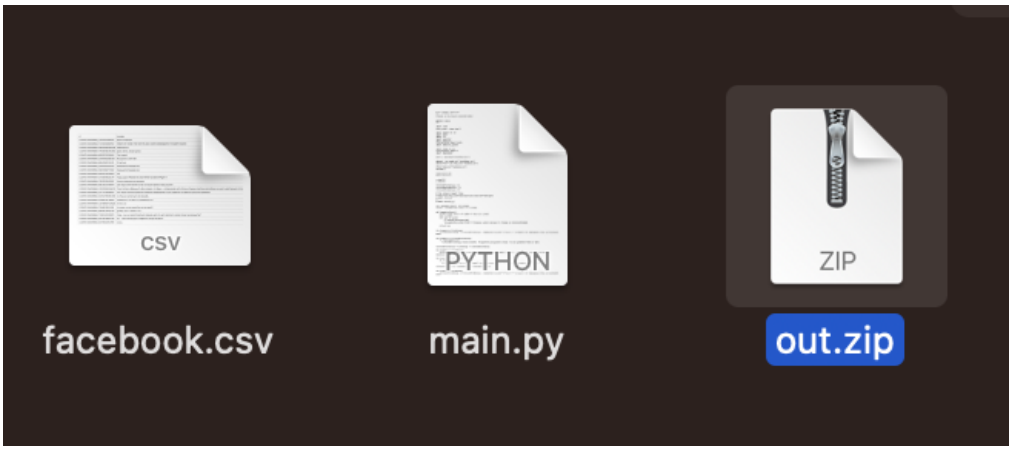


Рис. 4.6. Архів з результатами виконання програмного модуля

Результатом є таблиця, в якій описаний коментар, а також мітка, чи є даний коментар спамом (рис. 4.7)

predict	row
Not a bot	['Дуже смачно, вкотре дякую' '2023-11-13T14:53:44+0000']
Not a bot	['Дуже атмосферно' '2023-11-13T14:52:42+0000']
Not a bot	['Здравствуйте' '2023-11-23T18:59:03+0000']
Not a bot	['Удачі з дипломом!' '2023-12-06T19:10:56+0000']
Not a bot	['Такі хороші!' '2023-11-13T14:53:31+0000']
Bot	['Мы русские с нами Бог' '2023-11-23T18:59:27+0000']
Not a bot	['Я там була' '2023-11-13T14:53:20+0000']
Bot	['Оказывается Бандера жив' '2023-11-21T10:09:46+0000']

Рис. 4.7. Вихідний файл-таблиця з результатами класифікації

4.3. Тестування розробленого програмного модуля

Для тестування розробленого програмного модуля було створено групу в СМ Facebook для тестування з 4 учасниками. В групі наявні коментарі різного вмісту. Для тестування програмного модуля було зібрано реальні коментарі з Facebook різного вмісту.

Загальна кількість коментарів в тестовому наборі даних - 50, половина з них містять пропагандистські риторики, інша половина - ні (рис. 4.8).

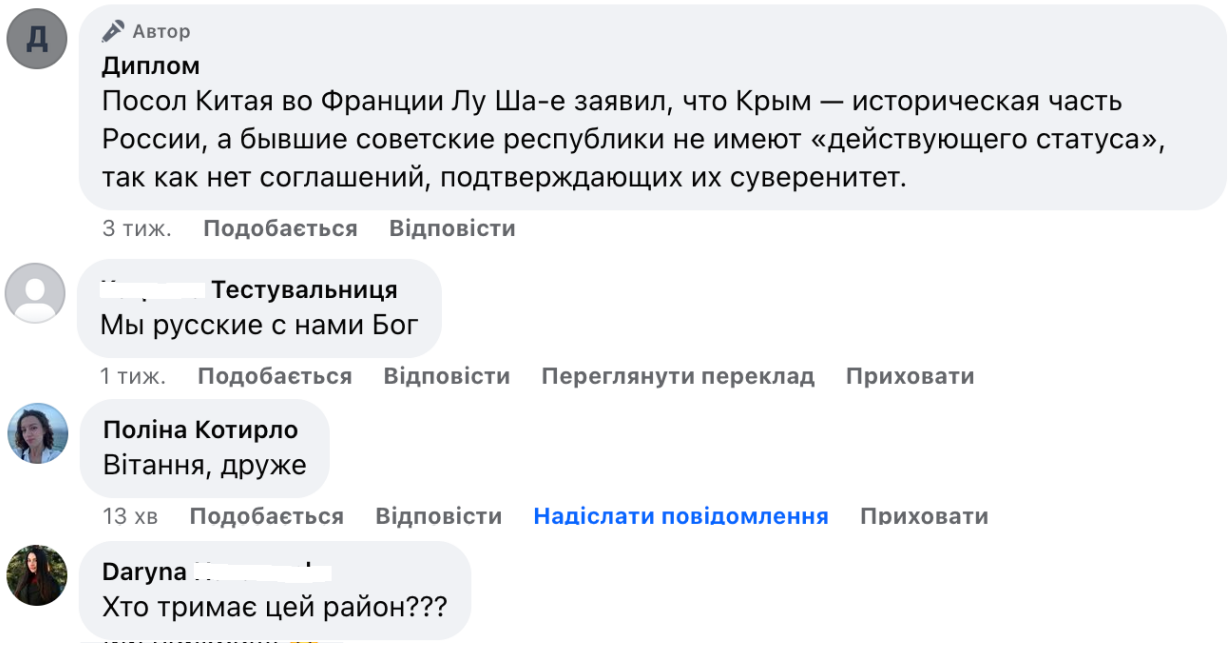


Рис. 4.8. Приклад коментарів різного вмісту

Як було описано в третьому розділі даної роботи з розробленому програму модулі використовуються два методи машинного навчання. Розглянемо ефективність роботи двох використаних методів при класифікації незалежно один від одного.

Було проведено тестування на однаковому наборі даних, експеримент повторювався 5 разів для отримання середніх результатів.

Таблиця 4.2

Результати оцінки ефективності методів ММН незалежно один від одного

	Метод k-найближчих сусідів	Метод дерево рішень
1	19 розпізнаних коментарів (76%)	17 розпізнаних коментарів (68%)
2	19 розпізнаних коментарів (76%)	19 розпізнаних коментарів (76%)
3	19 розпізнаних коментарів (76%)	18 розпізнаних коментарів (72%)
4	19 розпізнаних коментарів (76%)	19 розпізнаних коментарів (76%)
5	19 розпізнаних коментарів (76%)	21 розпізнаний коментар (84%)

Середний результат ефективності для МНС складає 77,6%, для МДР - 75,2%. Деякі коментарі, які класифікувались одним методом як спам не класифікувались іншим, тож було вирішено компілювати результати класифікації двох ММН.

Для оцінки роботи розробленого програмного модуля обраховується час його виконання. Тож як критерії оцінки ефективності було обрано час виконання та кількість виявлених пропагандистських коментарів. Було проведено 10 запусків програмного модуля з однаковими вихідними даними для визначення середніх значень ефективності та часу його роботи.

Варто зазначити, що дерево рішень показує дещо різний результат знайдених коментарів кожної ітерації. Це нормальне явище, оскільки алгоритм передбачає початково випадкове розбиття тренувальних даних для

навчання, оскільки це в подальшому виконанні алгоритму підвищує його ефективність для класифікації тестових даних.

Таблиця 4.3

Результати оцінки ефективності розробленого програмного модуля

Номер експерименту	1	2	3	4	5	6	7	8	9	10
Час виконання (ms)	658,82	785,79	834,04	761,67	814,03	548,76	658,44	811,28	566,18	593,46
Кількість визначених пропагандистських коментарів	21	22	21	22	22	20	20	22	22	23
Ефективність розпізнавання пропагандистських коментарів (%)	84	88	84	88	88	80	80	88	88	92

Важливо зазначити, що програмний модуль не позначив ні одного коментаря який не я спамом як спам, це говорить про відсутність перенавчання алгоритмів. Середнім результатом кількості визначених коментарів є 21,5, що є 86% від загальної кількості коментарів, які містять спам (рис 4.9).

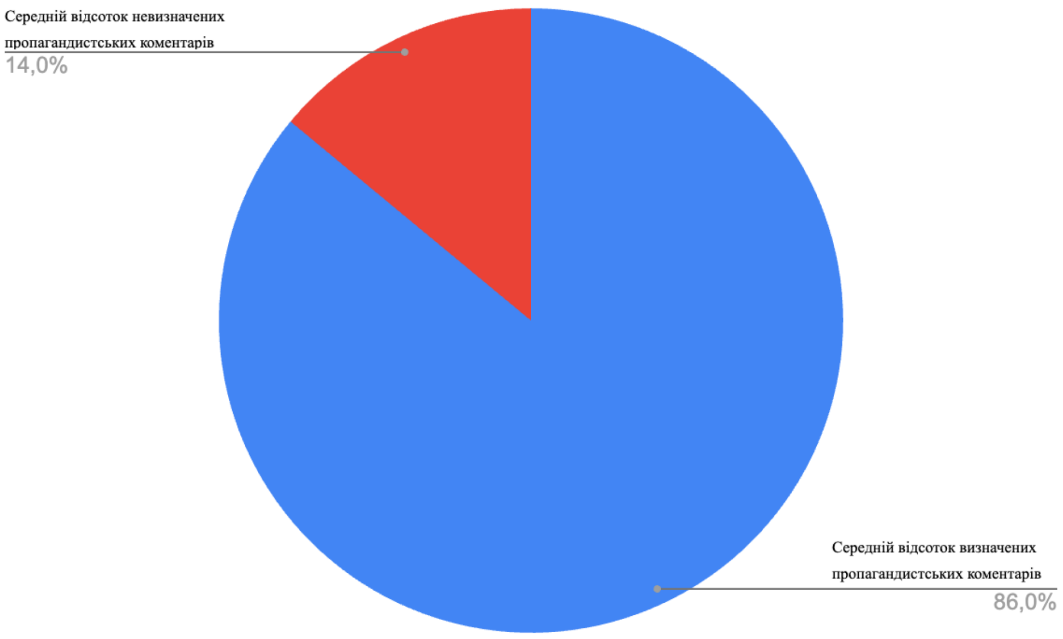


Рис. 4.9. Графічне представлення відсоткового співвідношення класифікації коментарів

Час виконання програмного модуля коливається, в діапазоні 566,18 ms - 834,04 ms (рис. 4.10).

Рис. 4.10. Діаграма часозатратності алгоритму

Через використаний МДР результати запуску розробленого програмного модуля можуть бути дещо різними за кожної ітерації, проте за результатами експериментів два методи разом безумовно мають вищу ефективність роботи в порівнянні з використанням їх окремо один від одного (таблиця 4.1).

Важливо зазначити, що розроблений програмний модуль не класифікував ні один коментар який не є спамом як спам. Це говорить про відсутність перенавчання алгоритму.

Тобто ймовірність помилок першого роду для розробленого програмного модулю близька до нуля за результатами експериментів. Щодо помилок другого роду, ймовірність таких помилок складає 14%.

Отже, для класифікації 50 коментарів розроблений програмний модуль витрачає менше секунди, при цьому забезпечуючи ефективність 86%. Порівнюючи цей результат з результатами тестування двох ММН незалежно

один від одного (таблиця 4.1), то розроблений програмний модуль на 8,4% ефективніший ніж МНС та на 10,8% ефективніший ніж МДР.

4.4. Порівняння розробленого програмного модуля з аналогами

Під час виконання дипломної роботи здобувача освітнього ступеня “бакалавр” мною було розроблено програмний модуль, який виконував аналогічну задачу пошуку пропагандистських коментарів у СМ Facebook.

Ключовою різницею між попереднім розробленим програмним модулем та поточним є підвищення ефективності роботи з 73% виявлених пропагандистських коментарів до 86%.

Такого результату було досягнуто за допомогою комбінації роботи двох окремих методів машинного навчання, які, однак, навчаються з одно набору навчальних даних, що значно ефективніше ніж їх використання незалежно одне від одного.

Також для підвищення розпізнавальних можливостей алгоритму було використано методи ОПМ, що зробило дані для пошуку більш “зрозумілими” для методів машинного навчання.

Це пришвидшило час виконання алгоритму в 10 разів (таблиця 4.1), часозатратність знизилась з 8,985 секунд до 710,14 ms (в середньому за результатами 10 експериментів, таблиця 4.2).

В другому розділі даної кваліфікаційної роботи було проаналізовано програмні рішення, що базуються та ММН та класифікують спам повідомлення/домени, зокрема з СМ (таблиця 2.3).

Таблиця 4.4

Порівняння розробленого програмного модуля з програмними рішеннями пошуку спаму

Програмне ріщення	Використані методи та технології	Види спаму, який виявляється	Ефективність роботи	Необхідна кількість навчальних даних
Кластеризація спам-доменів методами машинного навчання	Класифікатор випадкового лісу	Виявлення спам-доменів	92%	513000 тисяч доменів
Виявлення зловмисних повідомлень в соціальній мережі Twitter на основі URL-посилань	ММН Випадковий ліс	Фішинг, рекламний спам	86%	182000 тисячі твітів
Проєктування інформаційної системи для виявлення і запобігання масової дезінформації в Twitter	Моделі глибокого навчання	Дезінформація	71%	4000 рядків
Розроблений програмний модуль виявлення повідомлень пропагандистського змісту в соціальних мережах	Метод найближчого сусіда Метод дерево рішень	Спам-пропаганда	86%	150 рядків

Головною перевагою розробленого програмного рішення серед інших є те, що він легкий до перенавчання. Використані в проаналізованих програмних рішеннях ММН потребують значних об’ємів даних для

тренування в той час як розроблений програмний модуль працює з навчальним набором даних в 150 коментарів.

Швидке перенавчання алгоритму надзвичайно важливе в задачі пошуку пропаганди, оскільки в умовах інформаційної війни пропагандистські наративи постійно змінюються та еволюціонують.

Два з трьох описаних рішень мають нижчу ефективність розпізнавання спаму, ніж розроблений програмний модуль. Для розробленого програмного модуля важливо шукати саме спам-повідомлення з пропагандистськими риториками, всі описані в таблиці 4.4 програмні рішення шукають менш вузьконаправлені види спаму.

4.5. Висновки до розділу 4

Створені рекомендації по застосуванню програмного модуля дозволяють користувачу СМ Facebook використовувати його на своєму домашньому ПК.

Порівнюючи розроблений програмний модуль з аналогічним за метою модулем, що було розроблено мною при виконанні дипломної роботи здобувача освітнього ступеня “бакалавр” можна зробити такі висновки:

- Ефективність програмного модуля зросла з 73% до 86%.
- Часозатратність алгоритму була зменшена з 8,985 секунд до часу, що складає менше секунди та коливається в діапазоні 566,18 ms - 834,04 ms.
- Вдосконалена зручність використання програмного модуля, оскільки розроблений програмний модуль можна запускати за допомогою виконавчого файлу, без використання середовища розробки
- Ймовірність помилок першого роду для розробленого прогамного модуля близька до нуля.

Порівняємо розроблений програмний модуль з з іншими програмними рішеннями описаними в 2 розділі цієї роботи (таблиця 2.3).

Зроблено висновки, що ефективність програмного модулю порівняно з аналогами менша ніж у одного з трьох приведених прикладів.

Методи машинного навчання, що використані в описаних в таблиці 4.3 програмних рішеннях вимагають дуже значних об’ємів даних для навчання, що робить їх складними для перенавчання. Проте у випадку пошуку спам-повідомлень з пропагандистськими риториками дуже важливо вчасно та ефективно навчати алгоритм для його ефективної роботи.

Також в порівнянні зі всіма трьома програмними рішеннями у розробленому програмному модулі використано більш прості ММН, що значить, що він вимагає менших обчислювальних потужностей, а отже може бути використаний пересічним користувачем соціальних мереж.

РОЗДІЛ 5.

АНТРОПОГЕННІ ФАКТОРИ ТЕХНОСФЕРИ

Довкілля - навколишнє середовище людини, зумовлене в даний момент сукупністю факторів, здатних чинити пряму або непряму, негайну або віддалену дію на людину та її здоров’я і життя.

Техносфера - це регіон біосфери в минулому, перетворений людиною за допомогою прямого або непрямого впливу технічних засобів з метою найкращої відповідності своїм матеріальним і соціально-економічним потребам.

Протягом багатьох століть місце існування людини повільно змінювала свій вигляд і, як наслідок, мало змінювалися види і рівні негативних впливів. Так тривало до середини XIX ст. - початку активного росту впливу людини на середовище існування. У XX ст. в результаті масштабної антропогенної діяльності в багатьох регіонах світу відбулося глобальне забруднення довкілля і життєво необхідних джерел небезпечними і шкідливими для здоров'я людини речовинами.

На Землі виникли зони підвищеного забруднення біосфери, що призвело до її часткової, а в ряді випадків і до повної регіональної деградації. Цим змінам багато в чому сприяли високі темпи зростання чисельності населення на Землі (демографічний вибух) і його урбанізація; зростання споживання і концентрація енергетичних ресурсів; інтенсивний розвиток промислового та сільськогосподарського виробництва; масове використання засобів транспорту і ряд інших процесів.

Техносфера прийшла на зміну біосфері і в результаті на планеті залишилося мало територій з непошкодженими екосистемами. Найбільшою мірою екосистеми зруйновані в розвинених країнах - Європі, Північній Америці, Японії. Природні екосистеми збереглися тут на невеликих площах,

які оточені з усіх боків територіями, порушеними діяльністю людини. Тому що збереглися відносно невеликі плями біосфери піддаються сильному техносферной тиску.

Створюючи техносферу, людина прагнула до підвищення комфортності довкілля, до зростання комунікабельності, до забезпечення захисту від природних негативних впливів. Усе це позитивно вплинуло на умови життя і в сукупності з іншими факторами (поліпшення медичного обслуговування тощо) на тривалість життя людей.

Проте нераціональна господарська діяльність, багаторазово підсилена здобутками науково-технічного прогресу, призвела до пошкодження і вичерпання природних ресурсів, зміни регенераційних механізмів біосфери, деформації сформованого протягом багатьох мільйонів років природного кругообігу речовин та енергетичних потоків на планеті, порушення динамічної рівноваги глобальної земної соціоекосистеми.

Техносфера включає в себе регіони міста, промислової зони, виробничого і побутового середовища. До нових, техносферних, належать умови проживання людини в містах і промислових центрах, виробничі, транспортні і побутові умови життєдіяльності.

Найбільший "внесок" в забруднення навколишнього середовища вносять теплові електростанції, металургійні і хімічні заводи та ін.

Теплові електростанції викидають сполуки сірки, вуглецю та азоту, споживають велику кількість води, а стічні води теплових електростанцій забруднені і мають високу температуру, що стає причиною не лише хімічного, а і теплового забруднення.

Металургійна промисловість споживає також багато води, яка стає брудною в процесі виробництва. Різними видами виробництва характеризується хімічна промисловість.

Дуже забруднює довкілля автомобільний транспорт. У світі існує більше 780 млн. автомобілів, які дають 70-90% забруднень у містах.

Основний забруднювач атмосфери від автотранспорту - оксид нітрогену NOx (суміш оксидів нітрогену NO та NO2) та чадним газом (оксидом карбону (П) СО), що містяться у вихлопних газах. Частка транспортного забруднення повітря становить 60% за СО та 50% за NOx, від загального забруднення повітря цими газами. Підвищений вміст СО та NOx можна виявити у вихлопних газах невідрегульованого двигуна в режимі прогрівання.

Викиди шкідливих речовин автотранспортом характеризуються кількістю основних забруднювачів повітря, що потрапляють до атмосфери з вихлопних газів за певний проміжок часу.

До шкідливих речовин належать: чадний газ (0,3 - 10% об'єму вихлопних газів, вуглеводні - паливо, яке не згоріло (до 3% об'єму) оксид нітрогену 8 та сажа.

Пороговий рівень забруднювача - певна концентрація - нижче за яку не спостерігається негативного впливу. Важлива не сама наявність забруднювача, а його доза, тобто добуток концентрації на експозицію. При збільшенні експозиції пороговий рівень може знизитися в 1000 разів і більше. Кожен забруднювач має свій пороговий рівень, реакція на нього окремих організмів - індивідуальна, пороговий рівень залежить від наявності інших забруднювачів або стресових чинників.

Кожен забруднювач має свій пороговий рівень, реакція на нього окремих організмів - індивідуальна, пороговий рівень залежить від наявності інших забруднювачів або стресових чинників.

Коли забруднювачі змішуються з великим об'ємом повітря, їх концентрація може знизитися до порогового рівня. Потім, внаслідок діяльності мікроорганізмів грунту та впливу інших природних процесів відбуваються поглинання та асиміляція забруднювачів.

Деревні рослини поглинають з атмосфери 50 - 60% токсичних газів, тоді як грунти - 5 - 6%, атмосферна волога - 5 - 20%, водойми та тварини –

5%. Зелені рослини виділяють у повітря леткі речовини - фітонциди, які мають антимікробні властивості.

Академік Б П. Токін, який їх вивчав, довів що під впливом деяких речовин, що входять до складу фітонцидів, у повітрі знижується концентрація забруднювачів: СО - на 10 - 30%, сірчистого газу - на 50%, оксидів нітрогену - на 15 - 35 %.

ВИСНОВКИ

Зростання кількості користувачів Інтернету вже є ключовою тенденцією у сучасному світі. Проте з розвитком технологій також з’являються нові загрози, з якими користувачі мережі інтернет стикаються щоденно.

Однією з таких загроз є спам. Проблема спаму є глобальною та вимагає все більшої уваги у боротьбі з нею щороку. Згідно з аналізом компанії Statista [4], приблизно 84% усіх повідомлень електронної пошти є спамом.

Спам це проблема пов’язана не лише з електронною поштою, а також з соціальними мережами, месенджерами, блогами тощо. Соціальні мережі зараз продовжують продовжують набирати популярність та збільшують кількість користувачів щоденно, тож як середовище для пошуку спам-повідомлень було обрано саме соціальні мережі, зокрема Facebook. Facebook є найпопулярнішою соціальною мережею в Україні.

Для того щоб ефективно протидіяти загрозі важливо правильно її класифікувати. В нашому випадку завдання розробленого програмного модуля - класифікувати спам-повідомлення, що містять російську пропаганду, що є актуальним завданням в умовах війни.

В першому розділі роботи було виконано аналіз видів, форм та способів поширення спаму, що дозволило підібрати найбільш оптимальні шляхи та методи боротьби зі спамом, що були використані при розробці програмного модуля.

Важливим аспектом боротьби зі спамом пропагандистського характеру є вчасне пристосування до постійно змінюваних наративів. Розглянувши в другому розділі кваліфікаційної роботи методи боротьби зі спамом, інструментом для реалізації пошуку таких коментарів було обрано методи машинного навчання.

Машинне навчання представляє собою потужний інструмент для виявлення спаму, який має численні переваги. Цей метод автоматизує процес

виявлення, ефективно зменшуючи потребу у ручному аналізі. Моделі машинного навчання можуть швидко адаптуватися до нових форм спаму, що гарантує високий рівень актуальності захисту.

Обраними методами машинного навчання є метод найближчого сусіда та метод дерево рішень. Обидва методи добре справляються з задачами класифікації даних на основі тренувальних даних та ключових слів. Важливою складовою розробленого програмного модуля є простота використання та навчання, обидва методи не потребують великої кількості даних для навчання та великих обчислювальних потужностей.

Одним з шляхів підвищення ефективності методів машинного навчання є використання обробки природної мови. В другому розділі було розглянуто різновиди таких методів. Оскільки для обох методів машинного навчання. що використовуються в розробленому програмному модулі використовуються ключові слова, то було обрано такі види обробки природної мови, які допомогли б розширити коло знаходження коментарів з мінімальним ризиком “перенавчання” алгоритмів.

Отже, було обрано лематизацію - зведення ключових слів до їхньої базової форми, що розширить коло пошуку. Також застосовано метод обробки природної мови “Стоп-слова”, який прибирає малозначущі частини з речень, що дозволило підвищити швидкість виконання алгоритму за зменшило “шум” для методів машинного навчання.

Було досліджено наявні аналоги для розробленого програмного модуля та виявлено, що програми для пошуку спаму зазвичай недостатньо специфічно налаштовані для пошуку саме повідомлень пропагандистського характеру. Деякі з них не можуть бути застосовані саме для соціальної мережі Facebook через політику конфіденційності компанії-власника цієї соціальної мережі Meta.

В третьому розділі кваліфікаційної роботи було детально досліджено та описано алгоритми роботи використаних методів машинного навчання,

описано процес підготовки тренувальних даних та ключових слів у розробленому програмному модулі.

Було створено рекомендації щодо застосування розробленого програмного модуля, що дає змогу пересічному користувачеві Facebook використовувати створене програмне рішення.

Результати тестування показали ефективність програмного модуля 86% за швидкості виконання менше однієї секунди для 50 коментарів. Після проведення порівняльної характеристики з програмним модулем з аналогічним завданням пошуком пропагандистських коментарів, розробленим при виконанні дипломної роботи здобувача освітнього ступеня “бакалавр”, що ефективність нового програмного рішення не лише зросла на 13%, але при цьому часозатратність алгоритму знизилась в 10 разів.

Отже, в результаті виконання кваліфікаційної роботи отримані наступні результати:

- було проаналізовано види та форми спаму, методи боротьби зі спамом у соціальних мережах та на основі результату проведеного аналізу обрано оптимальні методи для вирішення задачі виявлення повідомлень пропагандистського змісту в соціальних мережах;
- було розроблено програмний модуль виявлення повідомлень пропагандистського змісту в соціальних мережах, в якому за рахунок поєднання двох різних методів машинного навчання, досягається підвищення ймовірності правильного розпізнавання повідомлення пропагандистського змісту в соціальних мережах.
- було проведено тестування розробленого програмного модуля виявлення повідомлень пропагандистського змісту в соціальних мережах, що дало змогу дослідити доцільність використання розробленого модулю для вирішення поставленої мети.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Аналіз методів виявлення деструктивних впливів в соціальних інтернет сервісах [Електронний ресурс] – Режим доступу: World Wibe Web. – http://eprints.zu.edu.ua/37557/1/%D0%97%D0%B1%D1%96%D1%80%D0%BD%D0%B8%D0%BA_%D0%86%D0%A2%D1%82%D0%B0%D0%9C%D0%A1_2023_%D0%B4%D1%80%D1%83%D0%BA.pdf#page=81

2. Дослідження методів Огляд основних підходів до розпізнавання спаму методами машинного навчання [Електронний ресурс] – Режим доступу: World Wibe Web. – https://ela.kpi.ua/bitstream/123456789/51499/1/%28249-252%29_Omelchenko.pdf

3. ласифікації листів українською мовою з метою виявлення спаму [Електронний ресурс] – Режим доступу: World Wibe Web. – URL: https://openarchive.nure.ua/server/api/core/bitstreams/0e7117b4-9a19-4d70-88bc-459e500e6f02/content

4. Average daily spam volume worldwide from October 2020 to September 2021 [Електронний ресурс] – Режим доступу: World Wibe Web. – URL: https://www.statista.com/statistics/1270424/daily-spam-volume-global/

5. Templeton, B.: n.d.b, Origin of the term "spam" to mean net abuse, http://www.templetons.com/brad/spamterm.html, p. 54.

6. У МВС попередили про новий фішинговий спам в Facebook [Електронний ресурс] – Режим доступу: World Wibe Web. – URL: https://suspilne.media/537129-ne-perehodte-za-posilannam-u-mvs-poperedili-pro-novij-fisingovij-spam-v-facebook/

7. Основні параметри для ідентифікації порушника інформаційної безпеки [Електронний ресурс] – Режим доступу: World Wibe Web. – URL: http://www.irbis-nbuv.gov.ua/cgi-bin/irbis_nbuv/cgiirbis_64.exe?I21DBN=LINK&

P21DBN=UJRN&Z21ID=&S21REF=10&S21CNR=20&S21STN=1&S21FMT=ASP_meta&C21COM=S&2_S21P03=FILA=&2_S21STR=Zi_2013_15_1_11

8. Blog Spam: A Review [Електронний ресурс] – Режим доступу: World Wibe Web. – URL: https://www.ceas.cc/2007/papers/paper-85.pdf

9. Digital 2023: global overview report [Електронний ресурс] – Режим доступу: World Wibe Web. – URL: https://datareportal.com/reports/digital-2023-global-overview-report

10. Social Media Spam: What Channels Are Most Spammy in 2022 [Електронний ресурс] – Режим доступу: World Wibe Web. – URL: https://foundationinc.co/lab/social-media-spam

11. 47% of Social Media Users Report Seeing More Spam in Their Feeds, Even as Networks Fight to Stop It [Електронний ресурс] – Режим доступу: World Wibe Web. – URL: https://blog.hubspot.com/marketing/social-media-users-seeing-more-spam

12. Алгоритм створення текстів масової розсилки: лінгвістичні особливості та технічний аспект [Електронний ресурс] – Режим доступу: World Wibe Web. – URL: https://jarch.donnu.edu.ua/article/view/11565/11441

13. Richard Alan Nelson, A Chronology and Glossary of Propaganda in the United States. Westport, Conn. and London: Greenwood Press, 1996. 336 pp. URL: https://www.tandfonline.com/doi/abs/10.1080/08821127.1997.10731916.

14. Шведа Ю.Р. Політичні партії. Енциклопедичний словник. Львів. Астролябія, 2005.

15. Пропаганда в соціальних мережах – загроза інформаційній безпеці держави [Електронний ресурс] – Режим доступу: World Wibe Web. – URL: https://sci.ldubgd.edu.ua/jspui/bitstream/123456789/6139/1/konf_04_04_2019.pdf#page=48

16. Дослідження методів фільтрації спаму на основі вмісту [Електронний ресурс] – Режим доступу: World Wibe Web. – URL:

https://openarchive.nure.ua/server/api/core/bitstreams/a7e41d4a-5c57-41d6-993c-8dff5132a29b/content

17. Математичні методи моделювання, розпізнавання образів та безпеки даних [Електронний ресурс] – Режим доступу: World Wibe Web. – URL: https://ela.kpi.ua/bitstream/123456789/57055/1/Omelchenko_Bakalavr.pdf

18. Система моделювання процесів спам-фільтрації [Електронний ресурс] – Режим доступу: World Wibe Web. – URL: https://krs.chmnu.edu.ua/jspui/bitstream/123456789/2363/1/401%D0%B7%20%D0%A0%D0%BE%D0%BC%D0%B0%D0%BD%D0%B5%D1%86%D1%8C%20%D0%92%D1%96%D0%BA%D1%82%D0%BE%D1%80%D1%96%D1%8F%20%D0%92%D0%B0%D0%BB%D0%B5%D0%BD%D1%82%D0%B8%D0%BD%D1%96%D0%B2%D0%BD%D0%B0.pdf

19. Програмне забезпечення виявлення спаму на основі машинного навачання [Електронний ресурс] – Режим доступу: World Wibe Web. – URL: https://ekmair.ukma.edu.ua/server/api/core/bitstreams/96b2ee74-19a9-489b-9b06-34493ccb17ee/content

20. Актуальні проблеми Data Mining [Електронний ресурс] – Режим доступу: World Wibe Web. – URL: http://csc.knu.ua/media/filer_public/38/03/3803002b-e068-4a08-8a6c-a4edc183892a/datamining20170917.pdf

21. Статистична модель прогнозування вартості автомобіля за даними автомобільного ринку України [Електронний ресурс] – Режим доступу: World Wibe Web. – URL: https://ekmair.ukma.edu.ua/server/api/core/bitstreams/590b3734-7e5c-48cc-80ec-f1704589696b/content

22. The 7 NLP Techniques That Will Change How You Communicate in the Future (Part I)‖ [Online]. Available: https://heartbeat.fritz.ai/the-7-nlp-techniques-that-will-change-how-youcommunicate-in-the-future-part-i-f0114b2f0497

23. 755 000 твітів, або Як російська «Фабрика тролів» намагалась впливати на порядок денний в Україні. Дослідження [Електронний ресурс] – Режим доступу: World Wibe Web. – URL: https://voxukraine.org//longreads/twitter-database/index.html

24. Кластеризація спам-доменів методами машинного навчання [Електронний ресурс] – Режим доступу: World Wibe Web. – URL: https://elartu.tntu.edu.ua/bitstream/lib/39760/2/Dyplom_Hrytsyuk_V_P_2022.pdf

25. Виявлення зловмисних повідомлень в соціальній мережі Twitter на основі URL-посилань [Електронний ресурс] – Режим доступу: World Wibe Web. – URL: https://ela.kpi.ua/bitstream/123456789/31419/1/Deshunina_magistr.docx

26. Проєктування інформаційної системи для виявлення і запобігання масової дезінформації в Twitter [Електронний ресурс] – Режим доступу: World Wibe Web. – URL: https://elartu.tntu.edu.ua/bitstream/lib/36810/1/%d0%9a%d0%be%d0%b7%d0%b0%d0%ba_%d0%90_%d0%92.pdf

Додаток А

ПРОГРАМНИЙ КОД

```
import time
start_time = time.time()

import pandas as pd
import math
import re
import operator
import datetime
from datetime import date
import dateutil.parser

import numpy as np
from keywords import *
import pymorphy2

morph = pymorphy2.MorphAnalyzer()

dataset = pd.read_csv('trainData.csv')
datasetTest = pd.read_csv('facebook.csv')

csv = dataset.iloc[:, 1:].values
csvTest = datasetTest.iloc[:, 1:].values

filteredFirstGroup = []
filteredSecondGroup = []
filteredThirdGroup = []

def lemmatize(text):
```

```
 words = text.split() # розбиття тексту на слова
   res = list()
   for word in words:
      p = morph.parse(word)[0]
       res.append(p.normal_form) # створення нового масиву зі словами в початковій формі

   return res

for element in firstGroup:
      filteredFirstGroup = filteredFirstGroup + lemmatize(re.sub(r'[^\w\s]'," ",element)) # приведення слів до початкової форми

for element in filteredFirstGroup:
   if len(element)< 3:
     filteredFirstGroup.remove(element) # видалення розділових знаків та слів довжиною менше 3 букв

filteredFirstGroup = firstGroup + filteredFirstGroup

for element in secondGroup:
                  filteredSecondGroup = filteredSecondGroup + lemmatize(re.sub(r'[^\w\s]'," ",element)) # приведення слів до початкової форми

for element in filteredSecondGroup:
   if len(element)< 3:
      filteredSecondGroup.remove(element) # видалення розділових знаків та слів довжиною менше 3 букв
```

Продовження додатку А

```
filteredSecondGroup = secondGroup + filteredSecondGroup

for element in thirdGroup:
    filteredThirdGroup = filteredThirdGroup + lemmatize(re.sub(r'[^\w\s]',"
",element)) # приведення слів до початкової форми

for element in filteredThirdGroup:
  if len(element)< 3:
    filteredThirdGroup.remove(element) # видалення розділових знаків та
слів довжиною менше 3 букв

filteredThirdGroup = thirdGroup + filteredThirdGroup

def isFirstGroup(message):
  for phrase in firstGroup:
    if phrase.lower() in message.lower():
      return 1
  return 0

def isSecondGroup(message):
  for phrase in secondGroup:
    if phrase.lower() in message.lower():
      return 1
  return 0

def isThirdGroup(message):
  for phrase in thirdGroup:
    if phrase.lower() in message.lower():
```

```
    return 1
  return 0

def parseDate(date_reg, facebook):
  if facebook:
    return time.mktime(dateutil.parser.isoparse(date_reg).timetuple())
  else:
                return time.mktime(datetime.datetime.strptime(date_reg,
"%m/%d/%Y").timetuple())


def parseBot(string):
  if string == 'False':
    return 'Not a bot'
  return 'Bot'


# Split the data into training and test data
def loadDataset():
  lst = []
  test = []

  for i in range(len(csv)):
          lst.append([isFirstGroup(csv[i][1]), isSecondGroup(csv[i][1]),
isThirdGroup(csv[i][1]), parseBot(str(csv[i][2]))])

  for i in range(len(csvTest)):
    test.append([isFirstGroup(csvTest[i][0]), isSecondGroup(csvTest[i][0]),
isThirdGroup(csvTest[i][0])])
```

Продовження додатку А

```
return lst, test

def euclideanDistance(instance1, instance2, length):
    distance = 0
    for x in range(length):
        distance += pow((instance1[x] - instance2[x]), 2)
    return math.sqrt(distance)

def getNeighbors(trainingSet, testInstance, k):
    distances = []
    length = len(testInstance)-1
    for x in range(len(trainingSet)):
        dist = euclideanDistance(testInstance, trainingSet[x], length)
        distances.append((trainingSet[x], dist))
    distances.sort(key = operator.itemgetter(1))
    neighbors = []
    for x in range(k):
        neighbors.append(distances[x][0])
    return neighbors

def getResponse(neighbors):
    # Creating a list with all the possible neighbors
    classVotes = {}
    for x in range(len(neighbors)):
        response = neighbors[x][-1]
        if response in classVotes:
            classVotes[response] += 1
        else:
```

```
classVotes[response] = 1
        sortedVotes = sorted(classVotes.items(), key=operator.itemgetter(1),
reverse=True)
    return sortedVotes[0][0]

def getAccuracy(testSet, predictions):
    correct = 0
    for x in range(len(testSet)):
        if testSet[x][-1] == predictions[x]:
            correct += 1
    return (correct/float(len(testSet))) * 100.0

predictions=[]

def main():
    trainDF, testDf = loadDataset()
    k = 3
    for x in range(len(testDf)):
        neighbors = getNeighbors(trainDF, testDf[x], k)
        result = getResponse(neighbors)
        predictions.append({
            "predict": result,
            "row": csvTest[x]
        })
        print('> predicted=' + repr(result))
main()

print("--- %s ms ---" % ((time.time() - start_time) * 1000))
```

```
import time

from DecisionTree import *
import pandas as pd
from sklearn import model_selection
import datetime
from datetime import date

dataset = pd.read_csv('facebook.csv')

csv = dataset.iloc[:, 1:].values

lst = []
predictionsTree = []

header = ['account age', 'message', 'avatar', 'bot']
keyPhrasesSpam = filteredFirstGroup + filteredSecondGroup + filteredThirdGroup
keyPhrasesMbSpam = thirdGroup

def parseMessageTree(message):
    is_spam = 'Not spam'
    for phrase in keyPhrasesSpam:
        if phrase.lower() in message.lower():
            is_spam = 'Spam'

    if is_spam == 'Not spam':
        for phrase in keyPhrasesMbSpam:
            if phrase.lower() in message.lower():
```

```
is_spam = 'Can be spam'

    if is_spam == 'Spam':
      row = '';
      for x in range(len(csv)):
              if csvTest[x][0] == message:
                row = csvTest[x]

      predictionsTree.append({
         "predict": 'bot',
         "row": row
      })

    return is_spam

for i in range(len(csv)):
    lst.append([parseMessageTree(csv[i][0])])

trainData, testData = model_selection.train_test_split(lst, test_size=0.2)

t = build_tree(trainData, header)

leaves = getLeafNodes(t)

innerNodes = getInnerNodes(t)

rezData = [*predictions, *predictionsTree]

df = pd.DataFrame(rezData)
```

Продовження додатку А

```
compression_opts = dict(method='zip', archive_name='out.csv')
df.to_csv('out.zip', index=False, compression=compression_opts)



print_tree(t)
print("--- %s ms ---" % ((time.time() - start_time) * 1000))
```