Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр
(освітній рівень)

на тему: "Використання методів машинного навчання в задачах виявлення спам−листів "

Виконав: студент (ка)

Спеціальності:
125 «Кібербезпека та захист інформації»
(шифр і назва напряму підготовки, спеціальності)

Дячун Всеволод Петрович
підпис (прізвище та ініціали)

Керівник Загородна Н.В.
підпис (прізвище та ініціали)

Нормоконтроль Тимощук Д. І.
підпис (прізвище та ініціали)

Завідувач кафедри Загородна Н.В.
підпис (прізвище та ініціали)

Рецензент
підпис (прізвище та ініціали)

м. Тернопіль – 2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
«__» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

за спеціальністю 125 Кібербезпека та захист інформації
(шифр і назва спеціальності)

Студенту Дячуну Всеволоду Петровичу
(прізвище, ім'я, по батькові)

1. Тема роботи ″Використання методів машинного навчання в задачах в задачах виявлення спам-листів

Керівник роботи Загородна Наталія Володимирівна, к.т.н., доц. завідувачка кафедри КБ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «22» 11 2024 року № 4/7-1119

2. Термін подання студентом завершеної роботи 20.12.2024

3. Вихідні дані до роботи

Літературні джерела, набір даних 2007 TREC Public Spam Corpus

4. Зміст роботи (перелік питань, які потрібно розробити)

Аналіз проблеми виявлення спам-листів

Теоретичні основи розробки систем виявлення спаму на основі методів машинного навчання

Практична реалізація методів машинного навчання для виявлення спаму

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Осухівська Г.М., к.т.н., доцент		
Безпека в надзвичайних ситуаціях	Клепчик В.М., проректор з адміністративно-господарської роботи та будівництва		

7. Дата видачі завдання 23.09.2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	23.09 – 25.09	*Виконано*
2.	Підбір наукових джерел про спам та способи його виявлення	26.09-13.10	*Виконано*
3.	Переклад та опрацювання наукових джерел про теоретичні підходи виявлення спаму	14.10-25.10	*Виконано*
4.	Виконання дослідження щодо пошуку шляхів підвищення ефективності виявлення спаму	26.10-10.11	*Виконано*
5.	Виконання експериментальних досліджень	11.11-20.11	*Виконано*
6.	Оформлення розділу «Аналіз проблеми виявлення спам-листів»	21.11-25.11	*Виконано*
7.	Оформлення розділу «Теоретичні основи розробки систем виявлення спаму на основі методів машинного навчання»	26.11-30.11	*Виконано*
8.	Оформлення розділу «Практична реалізація методів машинного навчання для виявлення спаму»	01.12-05.12	*Виконано*
9.	Виконання завдання до підрозділу «Охорона праці»	04.12-08.12	*Виконано*
10.	Виконання завдання до підрозділу «Безпека в надзвичай них ситуаціях»	04.12-08.12	*Виконано*
11.	Оформлення кваліфікаційної роботи	01.12-08.12	*Виконано*
12.	Нормоконтроль	09.12 – 11.12	*Виконано*
13.	Перевірка на плагіат	12.12 – 15.12	*Виконано*
14.	Попередній захист кваліфікаційної роботи	16.12 – 20.12	*Виконано*
15.	Захист кваліфікаційної роботи	23.12.2024	

Студент _____ Дячун В.П.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Загородна Н.В.

(підпис)	(прізвище та ініціали)

АНОТАЦІЯ

Використання методів машинного навчання в задачах виявлення спам–листів // ОР «Магістр» // Дячун Всеволод Петрович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБм-61 // Тернопіль, 2024 // С. 68, рис. – 15, табл. – 1, кресл. – __, додат. – 2.

Ключові слова: SPAM, HAM, МАШИННЕ НАВЧАННЯ, SVM, наївний Байєс.

Кваліфікаційна робота присвячена оцінки ефективності методів машинного навчання в задачах виявлення спаму

В першому розділі здійснено огляд проблеми виявлення спаму, як такої, наведено класифікацію спаму та проаналізовано основні методи фільтрації спаму.

В другому розділі кваліфікаційної роботи висвітлено особливості попередньої обробки текстових даних та описано моделі машинного навчання, які, згідно огляду джерел, найчастіше використовують для виявлення спаму.

В третьому розділі описано деталі практичної реалізації методів машинного навчання для виявлення спаму, проведено оцінку точності побудованих моделей.

ABSTRACT

Using Machine Learning Methods for Spam Email Detection // Thesis of educational level "Master"// Vsevolod Diachun // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Cybersecurity, group СБм-61 // Ternopil, 2024// P. 68, figs. 15, tables 1, drawings __ added. – 2.

Keywords: SPAM, HAM, MACHINE LEARNING, SVM, NAÏVE BAYES

The qualification paper is devoted to the evaluation of the effectiveness of machine learning methods in spam detection tasks

The first section provides an overview of the problem of spam detection, classifying spam and analyzing the main methods of spam filtering.

The second section of the qualification work highlights the peculiarities of text data pre-processing and describes machine learning models that, according to the review of sources, are most often used for spam detection.

The third section describes the details of the practical implementation of machine learning methods for spam detection, and evaluates the accuracy of the built models.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ML	—	Machine Learning
kNN	—	k Nearest Neighbors
SVM	—	Support Vector Machine
DNS	—	Domain Name System
IT	—	Inforamtion Technologies
PCA	—	Principal Component Analysis
LSTM	—	Long Short-Term Memory
GRU	—	Gated Recurrent Unit
RNN	—	Recurrent Neural Networks
NLP	—	Natural Language Processing
BoW	—	Bag of Words
TF-IDF	—	Term Frequency-Inverse Document Frequency
CNN	—	Convolutional Neural Network
NLTK	—	Natural Language Toolkit

ВСТУП

Актуальність теми. Сучасний цифровий світ характеризується величезним обсягом інформації, яка циркулює в мережах зв'язку, включаючи електронну пошту. Разом із легітимними повідомленнями значна частина трафіку припадає на спам, що представляє собою небажані повідомлення, які надсилаються масово. Ці листи часто мають на меті рекламування товарів, послуг, або ж розповсюдження зловмисного програмного забезпечення. Спам-листи стають серйозною загрозою для індивідуальних користувачів та організацій, оскільки вони спричиняють фінансові втрати, втрату часу та проблеми з безпекою даних.

Спам-листи також впливають на ефективність роботи корпоративних систем зв’язку, збільшуючи витрати на обробку даних та створюючи загрози кібербезпеці. Наприклад, спам може включати фішингові повідомлення, які намагаються отримати конфіденційну інформацію, або файли зі шкідливим програмним забезпеченням, що пошкоджує системи користувача. Тому проблема виявлення спаму є не лише питанням підвищення ефективності роботи користувачів, а й забезпеченням їх безпеки.

Виявлення спаму – це складне завдання, яке постійно ускладнюється новими хитрішими методами спамерів. Сучасні підходи до розв'язання цієї проблеми поєднують різноманітні техніки, зокрема використання білих та чорних списків, пошуку ключових слів, типових для спаму, статистичний аналіз, лінгвістичний аналіз та інші. Серед методів ідентифікації спаму особливе місце займають методи машинного навчання.

Мета кваліфікаційної роботи. Розробка, впровадження та оцінка ефективності методів машинного навчання для автоматичного виявлення спам-листів, що забезпечують високу точність класифікації, адаптивність до змін у шаблонах спаму та можливість інтеграції у реальні системи фільтрації електронної пошти.

Досягнення мети передбачало виконання наступних *завдань*:

- Провести аналіз сучасного стану проблеми виявлення спаму та визначити основні виклики, пов’язані з класифікацією текстових даних.

- Знайти та проаналізувати доступні набори даних для навчання моделей виявлення спам-листів.

- Розробити підхід до попередньої обробки тексту, включаючи токенізацію, лематизацію, видалення стоп-слів, а також представлення тексту у числовому форматі.

- Дослідити ефективність різних алгоритмів машинного навчання, таких як наївний Байєс, метод опорних векторів та дерева рішень, для задачі класифікації спаму.

- Провести експериментальну оцінку моделей на тестових даних та порівняти їх ефективність за ключовими метриками, такими як точність, повнота, F1-міра.

- Розробити ансамблевий підхід для об’єднання результатів кількох моделей з метою підвищення загальної точності виявлення спаму.

- Сформулювати рекомендації щодо вдосконалення моделей виявлення спаму для подальших досліджень.

Об’єкт дослідження: процес автоматичного виявлення та класифікації спам-листів у електронній пошті

Предмет дослідження: Методи машинного навчання, алгоритми класифікації текстових даних, підходи до попередньої обробки тексту, а також ансамблеві техніки для підвищення точності виявлення спаму.

Наукова новизна одержаних результатів кваліфікаційної роботи полягає в порівнянні точності виявлення спаму на основі різних реальних наборів даних. Також розроблено ансамблеву модель, яка використовує метод голосування більшістю для комбінування результатів різних класифікаторів, що дозволило досягти більшої стійкості та узгодженості рішень у порівнянні з індивідуальними моделями.

Практичне значення одержаних результатів. Результати дослідження спрямовані на підвищення ефективності систем виявлення спам-листів у реальних умовах. Запропоновані алгоритми та підходи до обробки текстових даних можуть бути інтегровані в сучасні платформи електронної пошти для автоматичного блокування небажаних повідомлень, що сприятиме зменшенню

часу та ресурсів, необхідних для роботи з електронною кореспонденцією. Розроблені методи з урахуванням їх адаптивності до нових типів спаму можуть бути корисними для розробників систем кібербезпеки, адміністраторів корпоративних мереж, а також для створення навчальних матеріалів у галузі машинного навчання та обробки тексту.

Апробація результатів магістерської роботи. Основні результати проведених досліджень обговорювались на XII науковій-технічній конференції «Інформаційні моделі, системи та технології» (м.Тернопіль, Україна).

Публікації. Окремі результати кваліфікаційної роботи опубліковано у працях конференції (див. Додаток А).

РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМИ ВИЯВЛЕННЯ СПАМ-ЛИСТІВ

Хоча електронна пошта стала невід'ємною частиною нашого життя завдяки швидкості та доступності, вона зіштовхнулася з серйозною проблемою – спамом. Масові розсилки небажаних повідомлень не лише захаращують наші поштові скриньки, але й завдають значних економічних збитків компаніям та користувачам. Спамери постійно вдосконалюють свої методи, що ускладнює боротьбу з цим явищем. Незважаючи на розробку складних алгоритмів для фільтрації спаму, проблема залишається актуальною і, схоже, не зникне доти, доки існує безкоштовний доступ до інтернету.

Компанія SpamCon Inc [1] підрахувала, що кожне спам-повідомлення коштує підприємству 1-2 долари через втрату продуктивності працівників. З огляду на мільярди спам-листів, що надсилаються щодня, загальні збитки від спаму становлять мільярди доларів. Великі інтернет-провайдери, такі як UUNet і Netcom, витрачають значні кошти на боротьбу зі спамом, виділяючи для цього цілі відділи та бюджети в мільйони доларів. Причина такої популярності спаму полягає в його низькій вартості та ефективності для рекламодавців. Спамери легко знаходять електронні адреси в інтернеті за допомогою спеціальних програм – спам-ботів.

1.1 Історія виникнення спаму

Історія спаму починається ще з ранніх етапів розвитку електронних комунікацій. Перші прояви спаму з'явилися задовго до інтернету в сучасному розумінні. Зокрема, його витоки можна простежити до епохи телеграфів і ранніх телефонних систем, коли розсилки рекламних повідомлень або шахрайських пропозицій відбувалися через ці канали зв'язку. Це заклало основу для розуміння спаму як небажаних повідомлень, що розсилаються масово з метою досягнення певної вигоди.

Перший випадок "цифрового" спаму був зафіксований 3 травня 1978 року []. У цей день співробітник компанії DEC (Digital Equipment Corporation), Гері

Туерк, надіслав рекламний лист про презентацію нового комп'ютера компанії через мережу ARPANET — попередника сучасного інтернету. Повідомлення отримали близько 400 користувачів, що на той час було значною кількістю. Незважаючи на те, що ініціатива Туерка викликала значне обурення серед користувачів, цей інцидент став першою спробою використання електронних мереж для масової розсилки небажаних повідомлень.

Термін "спам" з’явився трохи пізніше, у 1980-х роках, завдяки популярному комедійному скетчу Monty Python's Flying Circus. У сцені відвідувачі ресторану намагаються зробити замовлення, однак кожна страва в меню включає консервоване м'ясо SPAM, і слово "SPAM" постійно повторюється. Це асоціювалося з нав'язливістю та небажаністю інформації, що стало метафорою для масових рекламних повідомлень.

У 1990-х роках із популяризацією електронної пошти та глобальної мережі інтернет спам став справжньою проблемою. Тоді користувачі почали отримувати величезну кількість небажаних листів, у яких рекламувалися товари, послуги, фінансові схеми та навіть шахрайські пропозиції. Оскільки на той час фільтрів для автоматичної обробки пошти майже не було, спамери отримали практично необмежений доступ до мільйонів поштових скриньок.

1994 рік став переломним моментом в історії електронної пошти. Саме тоді, двоє юристів з Арізони, Кантер і Сігел, відправили перше комерційне спам-повідомлення [3]. Вони розіслали масовий лист до тисяч користувачів Інтернету з пропозицією допомоги в отриманні американської "зеленої картки". Вибір теми для першого спам-повідомлення був не випадковим. Отримання "зеленої картки" – це актуальна проблема для багатьох людей, особливо іммігрантів. Тому така пропозиція могла зацікавити широку аудиторію. Хоча Кантер і Сігел, можливо, не усвідомлювали масштабів того, що вони започаткували, їхній експеримент мав далекосяжні наслідки для всього Інтернету. Спам став серйозною проблемою, яка досі не повністю вирішена. Він перевантажує поштові сервери, відволікає користувачів від важливої інформації і створює сприятливе середовище для розповсюдження шкідливого програмного забезпечення.

Початок 2000-х років став переломним моментом у розвитку спаму. У цей час з'явилися не лише масові рекламні розсилки, але й фішингові листи та шкідливий спам, який містив віруси та троянські програми. Спамери почали використовувати складні методи для обходу примітивних фільтрів, зокрема підроблені заголовки листів, маскування тексту та вкладені зображення замість текстової реклами. Це спричинило значні виклики для інтернет-безпеки та підштовхнуло до розробки більш досконалих методів боротьби зі спамом.

Із розвитком соціальних мереж та месенджерів у 2010-х роках спам вийшов за межі електронної пошти. Спамери почали активно використовувати платформи, такі як Facebook, Twitter, WhatsApp та інші месенджери, для масового поширення небажаних повідомлень. Зокрема, популярності набули автоматизовані "боти", які залишають рекламні коментарі, надсилають фішингові посилання або поширюють шкідливий контент.

На сучасному етапі спам значно еволюціонував завдяки розвитку технологій штучного інтелекту та машинного навчання. Спамери застосовують методи генерації динамічних текстів, які змінюються для обходу фільтрів, а також техніки соціальної інженерії, що маніпулюють психологією користувачів. З іншого боку, методи боротьби зі спамом також вдосконалюються, включаючи системи машинного навчання для автоматичної класифікації листів та фільтрації небажаного контенту.

Таким чином, історія спаму є прикладом постійного протистояння між спамерами та розробниками систем захисту. З кожним новим етапом розвитку технологій спам стає дедалі складнішим і адаптивнішим, однак сучасні методи виявлення, засновані на машинному навчанні, забезпечують ефективний захист користувачів від небажаної інформації.

Незважаючи на те, що з моменту відправлення першого спам-повідомлення минуло вже багато років, проблема спаму залишається актуальною. Розуміння її історичних коренів допомагає нам краще оцінити масштаби цієї проблеми і шукати ефективні способи боротьби з нею.

1.2 Суть та ознаки спаму

Спам — це небажані або непотрібні електронні повідомлення, що розсилаються масово без згоди отримувачів. Його основна мета — донести інформацію до максимальної кількості людей, незалежно від їхніх потреб чи інтересів. Спам є поширеним явищем у цифровій комунікації, включаючи електронну пошту, соціальні мережі, SMS, форуми та месенджери. У багатьох випадках він використовується для реклами, шахрайства, розповсюдження шкідливого програмного забезпечення або збору особистих даних.

Ключовими характеристики спаму є:

- Масовість. Спам розсилається у величезних обсягах до тисяч або навіть мільйонів адресатів. Основна мета таких розсилок — охопити якомога більше людей, навіть якщо реакція отримувачів буде мінімальною.
- Анонімність. Часто спамери приховують свою справжню ідентичність, використовуючи фейкові електронні адреси, проксі-сервери або зламані облікові записи. Це ускладнює їх виявлення та блокування.
- Рекламний або зловмисний характер. Спам здебільшого має рекламну мету, пропонуючи товари, послуги, або фінансові схеми. У багатьох випадках спам включає шахрайські повідомлення, спрямовані на крадіжку даних або встановлення шкідливих програм.
- Відсутність індивідуалізації. Спам-листи зазвичай не орієнтовані на конкретного користувача. Вони містять універсальний текст, який може бути адресований будь-кому.
- Експлуатація вразливостей. Спамери часто використовують соціальні, технічні або психологічні вразливості користувачів. Наприклад, фішингові листи можуть виглядати як офіційні повідомлення від банків чи популярних сервісів.
- Економічна вигода для спамерів. Розсилка спаму є дешевою, але потенційно прибутковою для його авторів. Навіть невелика частка успішних відповідей від отримувачів може принести значний дохід.

На рисунку 1.1 зображено класифікацію ознак спаму, які поділяються на формальні та лінгвістичні. Формальні ознаки включають технічні

характеристики електронного листа або його відправника. Серед них виділяються: поштові адреси та IP-адреси, що можуть бути підозрілими або відсутні в системі; відсутність адреси відправника, що свідчить про анонімність розсилки; велика кількість адрес одержувачів або навпаки, некоректно зазначена адреса. Також важливим є відсутність IP-адреси в системі DNS, що вказує на нелегальність джерела, а також певний розмір і формат повідомлення і шлях доставки електронної пошти, який часто використовують спамери для масової розсилки. Ці ознаки дозволяють автоматично ідентифікувати підозрілі листи за допомогою аналізу мережевих характеристик та структурних особливостей повідомлень.

Рисунок 1.1 – Формальні ознаки спам-листів

Друга група — лінгвістичні ознаки спаму, що стосуються текстового вмісту повідомлення. Тут виділяються такі підкатегорії: слова і фрази, побудовані певним чином, які часто використовують спамери для привернення уваги або обходу фільтрів; евристичні ознаки, що базуються на дослідженні текстових шаблонів і виявленні підозрілих характеристик, таких як надмірна кількість заголовних літер, специфічні слова чи посилання; та статистичні ознаки, які аналізуються за допомогою алгоритмів машинного навчання та фільтрів, що виявляють закономірності у структурі тексту спаму. Сукупність цих ознак є основою для автоматичного розпізнавання спаму та створення ефективних фільтрів у сучасних системах електронної пошти.

1.3 Класифікація спаму

Спам можна класифікувати за різними критеріями, але найчастіше його поділяють за змістом та метою [4].

Типи спаму:

1. Рекламний або комерційний спам. Найбільш поширений тип, що просуває товари, послуги, вебсайти або події. Часто використовує привабливі пропозиції, знижки або акції.
2. Фішинговий спам. Спрямований на крадіжку особистої інформації, такої як паролі, номери кредитних карток або банківські реквізити. Зазвичай виглядає як офіційний лист від банку чи іншої установи.
3. Шкідливий спам (malware spam). Містить шкідливе програмне забезпечення у вкладеннях або посиланнях, яке може пошкодити комп’ютер користувача чи викрасти персональні дані.
4. Тролінг-спам. Це різновид спаму, який поєднує в собі елементи масової розсилки небажаних повідомлень (спаму) та провокаційної поведінки в Інтернеті (тролінгу)

Крім того існує спам політичного характеру, релігійний спам та спам, пов'язаний з аферами.

Проте, до основних видів спаму відносять, все-таки, типи, зображені на рисунку 1.2

Рисунок 1.2 – Основні різновиди спаму

В залежності від способу розповсюдження спам поділяють на: [5]

1. Email спам. Небажані масові розсилки електронних листів, які зазвичай містять рекламу, шахрайські пропозиції або шкідливі посилання, і відправляються великій кількості користувачів без їхньої згоди.

На рисунку 1.3 наведено приклад емейл спаму

Рисунок 1.3 – Приклад емейл-спаму

2. Спам на форумах та в соціальних мережах. Небажані повідомлення, які розміщуються в коментарях, постах або особистих повідомленнях, часто містять рекламні посилання.

Реклама-спам, зображена на рисунку 1.4 є прикладом спаму, метою якого можуть бути клікбетінг, лайкджекінг чи шахрайсто.

Рисунок 1.4 – Приклад спаму в соцмережах

Клікбейтінг (Clickbaiting) - це розміщення сенсаційних заголовків, які спонукають користувача перейти до контенту з метою отримання доходу від онлайн-реклами. Коли користувач переходить на сторінку, контент, як правило, не існує або кардинально відрізняється від того, що було описано в заголовку.

Накрутка лайків (likejacking) - це акт обману користувачів, який полягає в тому, що вони публікують оновлення статусу Facebook для певного сайту без попереднього відома або наміру користувача. Користувач може думати, що він

просто відвідує сторінку, але клік може запустити скрипт у фоновому режимі, який поділиться посиланням у Facebook.

3. Спам у месенджерах. З появою популярних платформ для миттєвого обміну повідомленнями, таких як WhatsApp або Telegram, спам почав поширюватися і через ці канали.

На рисунку 1.5 наведено один з небагатьох прикладів повідомлень в меседжерах. Більш прикладів за посиланням [6]

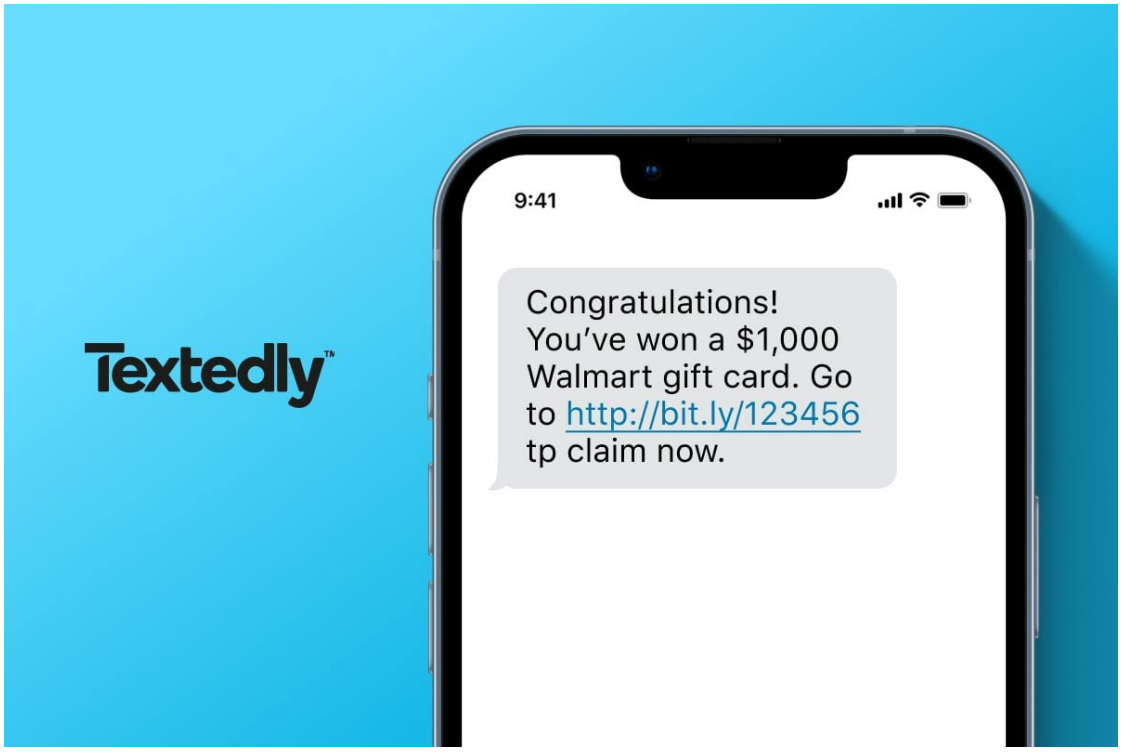

Рисунок 1.5 – Приклад спаму в меседжері з неочікуваним виграшем

4. Фейкові новини та шахрайство. Спам, що поширює фейкові новини та шахрайські схеми, становить серйозну загрозу для суспільства, оскільки підриває довіру до інформації та може призвести до фінансових втрат.

На рисунку 1.6 приклад фейкової новини. Під час українсько-російського протистояння цей вид спаму набув особливої важливості і активно використовується ворогом

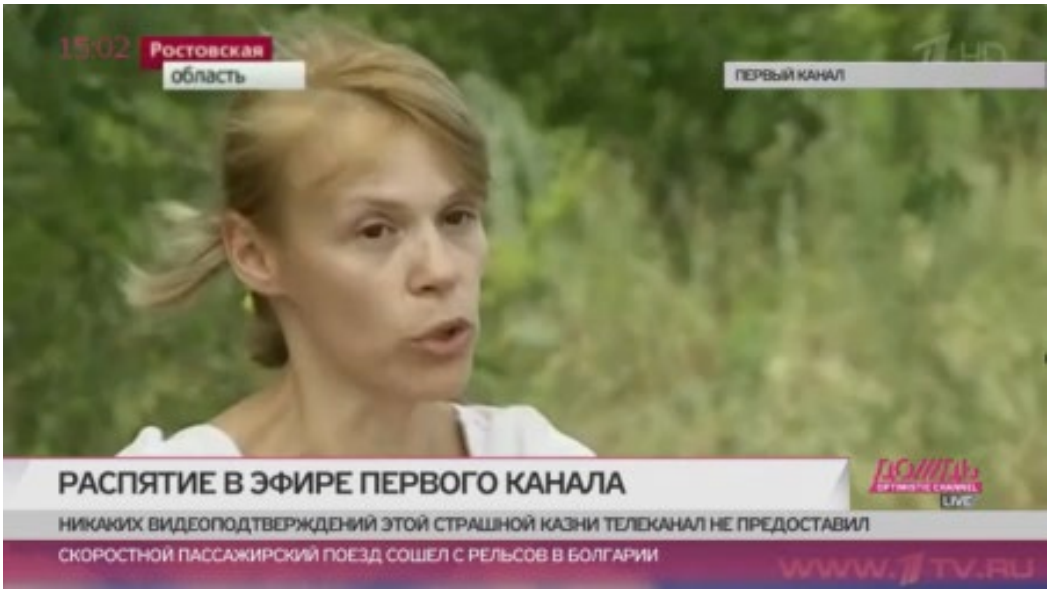

Рисунок 1.6 – Приклад фейкової новини на російському телебаченні

Фейкові новини є особливо небезпечними під час війни, оскільки дозволяють маніпулювати громадською думкою, підривають довіру до державних інституцій, розпалюють ненависть та ворожнечу.

1.4 Методи виявлення спаму: традиційні та сучасні підходи

Боротьба зі спамом є важливою з кількох причин. Спам може містити шкідливі програми, такі як віруси чи фішингові посилання, що загрожують безпеці особистих даних та фінансів. Окрім того, він забиває поштові ящики та інші комунікаційні канали, що знижує ефективність роботи, адже люди змушені витрачати час на видалення небажаних повідомлень. Спам також може призвести до зниження довіри до легітимних комунікацій, а в разі компаній — негативно вплинути на їх репутацію. Крім того, він навантажує інфраструктуру та ресурси, що створює додаткові витрати на підтримку систем безпеки. В результаті спам забруднює інформаційне середовище, спотворюючи доступ до корисної інформації і створюючи інформаційний шум.

Розглянемо детальніше основні підходи, які використовуються для автоматичної ідентифікації небажаних повідомлень.

Згідно [7] існує 4 групи основних підходів, відображених на рисунку 1.7

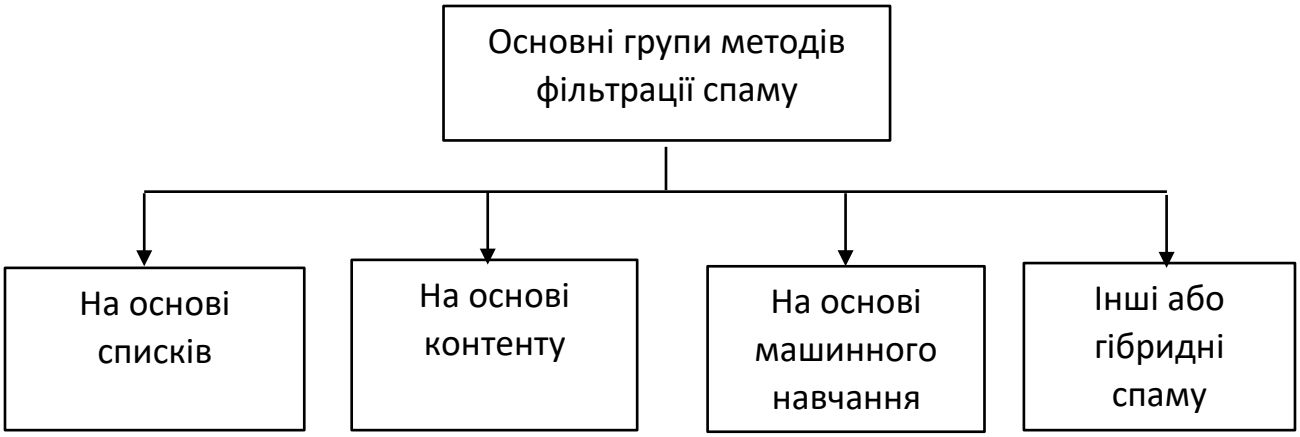

Рисунок 1.7 – Основні групи методів фільтрації спаму

1.4.1 Фільтрація спаму на основі списків

В [8] вказано, що базова фільтрація може використовувати наступні списки для ідентифікації спамових листів:

- Чорні списки. Цей популярний метод фільтрації спаму намагається зупинити небажану електронну пошту, блокуючи повідомлення з попередньо встановленого списку відправників, створеного вами або системним адміністратором вашої організації. Чорні списки - це записи адрес електронної пошти або IP-адрес, які раніше використовувалися для надсилання спаму. Коли надходить вхідне повідомлення, спам-фільтр перевіряє, чи є його IP-адреса або адреса електронної пошти в чорному списку; якщо так, повідомлення вважається спамом і відхиляється.

Хоча чорні списки гарантують, що відомі спамери не зможуть отримати доступ до поштових скриньок користувачів, вони також можуть помилково ідентифікувати легітимних відправників як спамерів. Крім того багато спритних спамерів регулярно змінюють IP-адреси та адреси електронної пошти, щоб замести сліди.

- Білі списки. Білий список блокує спам за допомогою системи, майже протилежної до чорного списку. Замість того, щоб вказувати, від яких відправників блокувати пошту, білий список дозволяє вказати, від яких відправників дозволяти отримувати пошту; ці адреси вносяться до списку довірених користувачів. Більшість спам-фільтрів дозволяють використовувати

білий список на додаток до інших засобів боротьби зі спамом, щоб зменшити кількість легальних повідомлень, які випадково позначаються як спам. Однак використання дуже суворого фільтра, який використовує лише білий список, означатиме, що кожен, хто не пройшов перевірку, буде автоматично заблокований.

- Чорні списки в режимі реального часу. Чорний список у режимі реального часу. Цей метод фільтрації спаму працює майже так само, як і традиційний чорний список, але вимагає менше зусиль для його підтримки. Це пов'язано з тим, що більшість чорних списків у режимі реального часу підтримуються третіми сторонами, які витрачають час на створення комплексних чорних списків від імені своїх підписників. Ваш фільтр просто повинен підключатися до сторонньої системи щоразу, коли надходить лист, щоб порівняти IP-адресу відправника зі списком.

Як і у випадку з чорним списком, списки «чорних дір» у реальному часі також можуть генерувати хибні спрацьовування, якщо спамери використовують легітимну IP-адресу як канал для спаму. Крім того, оскільки список, який підтримується третьою стороною, неможливо контролювати.

- Сірі списки. Відносно новий метод фільтрації спаму, сірі списки використовують той факт, що багато спамерів намагаються надіслати партію небажаної пошти лише один раз. За системою сірих списків поштовий сервер, який отримує повідомлення від невідомих користувачів, спочатку відхиляє їх і надсилає повідомлення про помилку серверу-відправнику. Якщо поштовий сервер намагається надіслати повідомлення вдруге - крок, який зробить більшість легальних серверів - сірий список вважає, що повідомлення не є спамом, і дозволяє йому потрапити до поштової скриньки одержувача. На цьому етапі фільтр сірих списків додає електронну або IP-адресу одержувача до списку дозволених відправників.

Хоча фільтри сірих списків вимагають менше системних ресурсів, ніж деякі інші типи спам-фільтрів, вони також можуть затримувати доставку пошти, що може бути незручно, коли ви очікуєте на важливі повідомлення.

1.4.2 Фільтрація на основі вмісту повідомлення

Замість того, щоб застосовувати універсальні політики до всіх повідомлень з певної електронної пошти або IP-адреси, фільтри на основі вмісту оцінюють слова або фрази, знайдені в кожному окремому повідомленні, щоб визначити, чи є лист спамом, чи легальним.

В рамках цього підходу виділяють наступні три напрями:

- Фільтрація на основі ключових слів.

Спам-фільтр на основі слів - це найпростіший тип фільтрів на основі вмісту. Загалом, фільтри на основі слів просто блокують всі листи, які містять певні терміни.

Оскільки багато спам-повідомлень містять терміни, які не часто зустрічаються в особистому або діловому спілкуванні, фільтри слів можуть бути простим, але ефективним методом боротьби зі спамом. Однак, якщо їх налаштувати на блокування повідомлень, що містять більш поширені слова, ці типи фільтрів можуть генерувати помилкові спрацьовування. Наприклад, якщо фільтр налаштовано на блокування всіх повідомлень, що містять слово «знижка», листи від легітимних відправників, які пропонують вашій неприбутковій організації обладнання або програмне забезпечення за зниженою ціною, можуть не дійти до адресата. Також спамери часто навмисно неправильно пишуть ключові слова, щоб обійти словесні фільтри, IT-персонал повинен мати час для регулярного оновлення списку заблокованих слів у фільтрі.

- Евристичні фільтри (на основі правил). Евристичні (або засновані на правилах) фільтри виходять на крок за межі простих фільтрів на основі слів. Замість того, щоб блокувати повідомлення, які містять підозріле слово, евристичні фільтри беруть до уваги кілька термінів, знайдених в електронному листі.

Евристичні фільтри сканують вміст вхідних листів і присвоюють бали словам або фразам. Підозрілі слова, які часто зустрічаються в спам-повідомленнях, такі як «Rolex» або «Viagra», отримують вищі бали, тоді як терміни, які часто зустрічаються в звичайних листах, отримують нижчі бали. Потім фільтр підсумовує всі бали і виводить загальну оцінку. Якщо

повідомлення отримує певну кількість балів або вище (визначену адміністратором антиспаму), фільтр ідентифікує його як спам і блокує.

- Байєсівські фільтри. Ці фільтри, які вважаються найдосконалішою формою фільтрації на основі вмісту, використовують закони математичної ймовірності для визначення того, які повідомлення є легітимними, а які - спамом. Щоб байєсівський фільтр ефективно блокував спам, кінцевий користувач повинен спочатку «навчити» його, вручну позначаючи кожне повідомлення як небажане або легальне. З часом фільтр бере слова і фрази, знайдені в легальних листах, і додає їх до списку; те саме він робить і з термінами, знайденими в спамі.

Щоб визначити, які вхідні повідомлення класифікуються як спам, байєсівський фільтр сканує вміст листа, а потім порівнює текст зі своїми списками з двох слів, щоб обчислити ймовірність того, що повідомлення є спамом. Наприклад, якщо слово «валіум» з'явилося 62 рази в списку спаму, але лише тричі в легальних листах, існує 95-відсоткова ймовірність того, що вхідний лист, який містить слово «валіум», є спамом.

Оскільки байєсівський фільтр постійно формує свій список слів на основі повідомлень, які отримує окремий користувач, він теоретично стає ефективнішим, чим довше його використовують. Однак, оскільки цей метод вимагає певного періоду навчання, перш ніж він почне працювати належним чином, вам потрібно буде набратися терпіння і, можливо, доведеться вручну видалити кілька небажаних повідомлень, принаймні на початку.

1.4.3 Фільтри на основі методів машинного навчання.

Машинне навчання є одним з найважливіших і найпоширеніших підходів до виявлення спаму і не лише спаму, а й будь-якої аномальної активності [9] і зазвичай поділяється на контрольоване і неконтрольоване навчання.

- Контрольоване навчання. Це метод машинного навчання, при якому модель навчається на основі вхідних даних, що супроводжуються правильними відповідями або мітками. Метою є створення моделі, яка здатна передбачати або класифікувати нові, раніше невідомі дані на основі шаблонів, виявлених під час навчання. У процесі навчання модель порівнює свої прогнози з реальними

результатами і поступово коригує свої параметри, щоб зменшити помилки та покращити точність передбачень.

Спочатку збирається велика кількість даних, які містять як вхідні дані (наприклад, текст, зображення), так і правильні відповіді (мітки). Наприклад, для класифікації спаму, це можуть бути тисячі листів, кожен з яких позначений як "спам" або "не спам". Далі алгоритм машинного навчання використовує ці дані для навчання. Він намагається знайти закономірності між вхідними даними та відповідними мітками. Після навчання моделі необхідно оцінити її якість шляхом проведення тестування: модель перевіряється на нових даних, яких вона раніше не бачила. Це дозволяє зрозуміти, наскільки точно модель може передбачити правильну відповідь. Якщо модель показала достатню точність, її можна використовувати для прогнозування результатів на нових, невідомих даних.

До методів контрольованого навчання належать: метод k-найближчих сусідів (K-Nearest Neighbors - KNN), метод опорних векторів (Support Vector Machine - SVM), метод Наївного Байеса та інші.

- Неконтрольоване навчання — це метод машинного навчання, при якому модель навчається на даних без супроводу з мітками або правильними відповідями. Вона повинна самостійно знаходити структуру в даних, виявляти закономірності, зв'язки або класифікації, не маючи заздалегідь визначених категорій. Це дозволяє моделі виявляти приховані патерни або групи об'єктів, що можуть бути корисними для аналізу, наприклад, через кластеризацію або зниження розмірності даних, без явних вказівок на те, що потрібно шукати.

Неконтрольоване навчання для виявлення спаму працює за принципом виявлення прихованих патернів у великих обсягах даних без використання міток чи позначень, таких як "спам" або "не спам". Модель аналізує вхідні дані, зокрема текст повідомлень або їх характеристики, і намагається знайти спільні ознаки чи структури серед повідомлень, що можуть свідчити про спам. Це дозволяє виявляти групи схожих повідомлень, які можуть бути спамом, навіть якщо ці групи не мали чітких позначень під час навчання.

Для виявлення спаму можуть використовуватись різноманітні методи неконтрольованого навчання. Одним із таких є кластеризація, коли алгоритм групує повідомлення, схожі за певними ознаками, в окремі класи. Наприклад, повідомлення з подібними шаблонами або словами можуть бути об'єднані в одну групу, яка, ймовірно, буде спамом. Також застосовуються методи зниження розмірності, наприклад, метод головних компонент (PCA), для виявлення важливих характеристик даних, що допомагають розпізнати спам. Такий підхід дозволяє автоматично знаходити спам у великих обсягах даних без потреби в точних мітках або ручному маркуванні кожного повідомлення.

Окремим видом машинного навчання є так зване глибоке навчання (deep learning)

Глибоке навчання — це підхід в машинному навчанні, який використовує нейронні мережі з багатьма шарами для автоматичного видобутку характеристик і патернів з даних. Воно відрізняється від традиційних методів машинного навчання здатністю самостійно вивчати складні особливості даних, не потребуючи ручного вибору ознак. Глибокі нейронні мережі можуть ефективно працювати з великими обсягами даних і вирішувати завдання, що вимагають складної обробки, такі як розпізнавання образів, обробка тексту та аналіз аудіо.

У випадку виявлення спаму, глибоке навчання може бути використане для автоматичної обробки текстових повідомлень і визначення, чи є вони спамом. Нейронні мережі можуть навчатися на великих наборах даних, що містять як спам, так і не спам повідомлення, і виявляти складні патерни, такі як специфічні комбінації слів, стиль письма або інші контекстуальні фактори, які часто зустрічаються у спам-повідомленнях. Це дозволяє моделі ефективно класифікувати нові повідомлення, навіть якщо вони мають нові або невідомі шаблони.

Для виявлення спаму найчастіше використовуються рекурентні нейронні мережі (RNN), які ефективно працюють з послідовностями даних, такими як текст. Зокрема, LSTM (Long Short-Term Memory) і GRU (Gated Recurrent Unit) моделі добре справляються з обробкою довгих текстів та запам'ятовуванням контексту, що є важливим для точного розпізнавання спаму. Крім того, для

обробки текстів можуть застосовуватись трансформери, такі як BERT або GPT, які показують відмінні результати у розумінні контексту та виявленні важливих факторів для класифікації повідомлень як спам чи не спам.

1.4.4 Інші методи фільтрації спаму

До інших методів виявлення спаму відносять широкий спектр методів, до яких належать:

- Лінгвістичний аналіз. Аналіз синтаксису, семантики та стилю повідомлення для виявлення ознак спаму.

- Аналіз зображень. Для виявлення спаму, який містить зображення з текстом.

- Аналіз IP-адрес. Відстеження IP-адрес, пов'язаних з великою кількістю спаму.

- Аналіз часу відправлення. Виявлення пікових періодів відправлення спаму.

- Аналіз заголовків. Перевіряється інформація про відправника, маршрут проходження листа та інші технічні деталі.

- Аналіз доменних імен (Системи пошуку DNS). Хоча сам по собі цей метод не є особливо надійним, деякі методи боротьби зі спамом використовують систему доменних імен (DNS), за допомогою якої ідентифікуються всі поштові сервери в Інтернеті, для виявлення та відсікання спамерів.

Сучасні системи виявлення спаму часто використовують комбінацію цих підходів, а також постійно навчаються на нових даних, щоб адаптуватися до нових тактик спамерів.

Важливо зазначити: жоден метод не є ідеальним, і завжди існує ймовірність, що деякі легітимні повідомлення можуть бути помилково відфільтровані як спам.

РОЗДІЛ 2 ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ СИСТЕМ ВИЯВЛЕННЯ СПАМУ НА ОСНОВІ МЕТОДІВ МАШИННОГО НАВЧАННЯ

2.1 Попередня обробка даних для задачі класифікації спаму

Попередня обробка даних перед класифікацією є необхідним кроком, оскільки вона підготовляє дані до подальшого аналізу та підвищує точність моделей машинного навчання. Традиційно вона включає наступні кроки: очищення даних від шуму та виявлення пропущених значень, нормалізацію чи стандартизацію даних, кодування категоріальних змінних та вибір релевантних ознак. Проте попередня обробка текстових даних має свою специфіку. До методів нормалізації текстових даних відносять:

- приведення текстових даних до одного регістру;
- видалення розділових знаків;
- видалення слів, що містять цифри;
- токенізація
- видалення стоп-слів;
- стемінг;
- лематизація.

Якщо з першою частиною методів попередньої обробки все зрозуміло, потребує додаткових роз’яснень.

2.1.1 Токенізація

Токенізація — це процес розбиття тексту на менші складові частини, які називаються токенами. Токени можуть бути словами, фразами, символами або навіть окремими реченнями, залежно від задачі та вибраного підходу. Основна мета токенізації — підготовка текстових даних до аналізу чи обробки алгоритмами машинного навчання або інших методів. Цей етап є одним із ключових у природній обробці мови (NLP), адже без структурування тексту його автоматичне опрацювання було б надзвичайно складним.

Токенізація зазвичай базується на певних правилах, таких як розділення тексту за пробілами, розділовими знаками чи іншими символами. Наприклад, у реченні "Привіт, світе!" токенами можуть бути "Привіт" і "світе", якщо орієнтуватися на слова, або "Привіт,", " ", і "світе!", якщо враховувати кожний символ. Різні підходи до токенізації можуть бути корисними для різних задач. Наприклад, аналіз настроїв може потребувати токенізації на рівні слів, а розпізнавання мови — на рівні символів.

Ілюстрація одного з підходів токенізації наведена на рисунку 2.1.

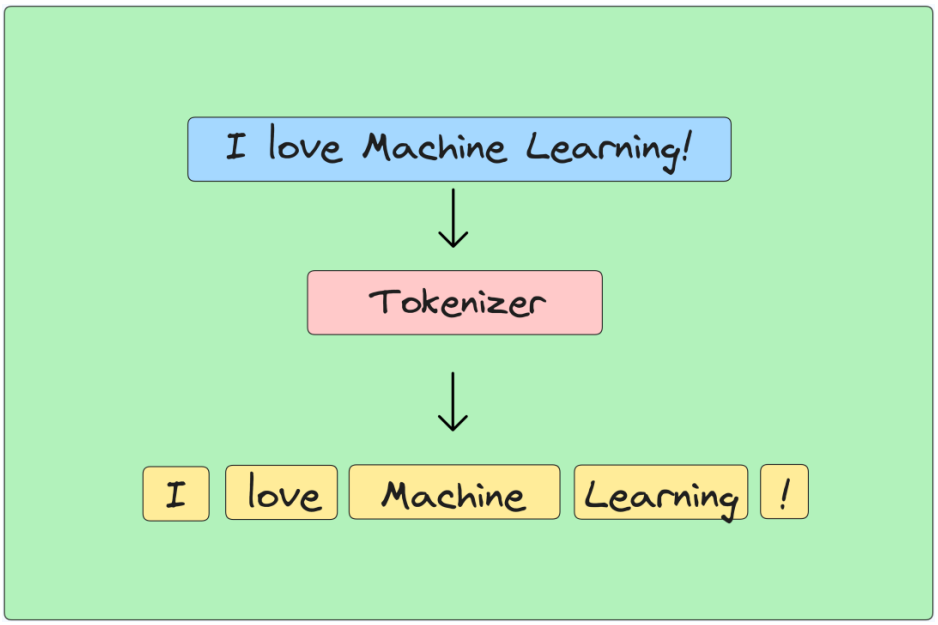

Рисунок 2.1 – Ілюстрація токенізації

Токенізація важлива, оскільки вона допомагає структурувати текст для подальшого використання в алгоритмах. У випадку великих текстових масивів, токенізація дозволяє виділити ключові одиниці, які несуть зміст. Для мов із складною морфологією (наприклад, українська чи фінська) токенізація може включати додаткові кроки, наприклад, врахування контексту або лексичних правил, щоб уникнути помилок у розділенні слів.

2.1.2 Стоп-слова

Стоп-слова — це загальновживані слова, які часто зустрічаються в текстах, але зазвичай не несуть вагомого змістового навантаження. Приклади таких слів

в українській мові — "і", "в", "на", "це", "та". У більшості мов вони виконують граматичну чи службову функцію, допомагаючи пов'язувати слова у реченнях, але не додають унікальності чи смислової цінності тексту для аналізу. У контексті обробки природної мови (NLP), їх зазвичай розглядають як зайві для задач машинного навчання, спрямованих на розпізнавання основних тем, кластеризацію чи аналіз тексту.

Видалення стоп-слів під час попередньої обробки тексту дозволяє зменшити обсяг даних, які обробляються моделлю, і зосередитися на словах, що мають більшу інформативність. Це покращує ефективність алгоритмів, знижує обчислювальну складність і допомагає уникнути шуму в даних. Наприклад, при побудові моделі для аналізу настрою тексту слова типу "він" або "але" зазвичай не допомагають визначити емоційне забарвлення, тому їх можна безпечно ігнорувати.

2.1.3 Стеммінг

Стеммінг – це процес спрощення слів шляхом видалення кінцівок, або афіксів, таких як закінчення або префікси. Цей процес дозволяє звести різні форми одного слова до спільного кореня. Наприклад, слова "бігає", "бігала", "бігати" будуть зведені до кореня "біг".

Це потрібно зокрема для удосконалення пошукової систему. Якщо потрібно знайти інформацію про "бігання", то ми б хотіли отримати результати, які містять не лише слово "бігання", але й його різні форми. Стеммінг допомагає об'єднати всі ці форми під одним коренем, покращуючи точність пошуку.

Алгоритми стеммінгу використовують набір правил або евристик для видалення кінцівок слів. Ці правила можуть бути досить простими, наприклад, видаляти всі літери після останньої голосної, або більш складними, що враховують морфологічні особливості мови. Важливо зазначити, що стеммінг не завжди призводить до отримання леми слова, тобто його словникової форми. Іноді він може давати неправильні результати, особливо для слів з нестандартними формами.

Приклади стеммінгу:

- "граючи" -> "гра";
- "читав" -> "чит";
- "найкращий" -> "кращ";
- "неможливо" -> "неможл".

Як бачимо, стеммінг може призводити до отримання неіснуючих слів (наприклад, "чит" замість "читати"). Це один з недоліків цього методу.

Стеммінг широко використовується в інформаційному пошуку, природній обробці мови та інших областях, де необхідно швидко обробляти великі обсяги текстових даних. Хоча він має свої обмеження, стеммінг залишається важливим інструментом для попередньої обробки текстів.

Порівняно з лематизацією, стеммінг є більш простим і швидким процесом, але менш точним. Лематизація використовує словники та правила морфології для визначення леми слова, тоді як стеммінг оперує на більш загальних правилах.

2.1.4 Лематизація

Лематизація є фундаментальним процесом у комп'ютерній лінгвістиці, спрямованим на зведення слова до його канонічної форми – леми. Лема є словниковою формою слова, яка відображає його базове значення. Наприклад, для дієслова "бігає" лемою буде "бігати". Цей процес відіграє ключову роль у багатьох завданнях обробки природної мови, таких як інформаційний пошук, машинний переклад та аналіз настроїв.

Головною метою лематизації є зменшення розмірності словника та усунення неоднозначностей, пов'язаних з різними формами одного слова. Це дозволяє алгоритмам машинного навчання більш ефективно виявляти семантичні зв'язки між словами та покращує точність подальшого аналізу тексту.

Процес лематизації є більш складним, ніж стеммінг, оскільки він вимагає глибокого розуміння морфології мови. Для виконання лематизації використовуються словники лем, які містять інформацію про різні форми слів та їх відповідні леми. Крім того, сучасні алгоритми лематизації часто використовують статистичні моделі та машинне навчання для визначення лем, особливо для слів, яких немає в словнику.

Лематизація відрізняється від стеммінгу тим, що вона намагається зберегти семантичну інформацію слова. Наприклад, стеммінг може звести слова "кращий" і "гірший" до одного кореня "кращ", що може призвести до втрати протилежного значення. Лематизація ж зведе ці слова до лем "хороший" і "поганий", відповідно, зберігши їх семантичну протилежність.

Приклади лематизації:

- "читав" -> "читати";
- "найкращий" -> "хороший";
- "бігала" -> "бігати";
- "відвідав" -> "відвідати";

Як бачимо, лематизація дозволяє звести різні форми слова до єдиної леми, що полегшує подальший аналіз тексту.

Лематизація важлива в обробці текстових даних тому, що алгоритми машинного навчання краще працюють з даними, представленими в уніфікованому вигляді. Коли ми зводимо різні форми слова до однієї леми, ми зменшуємо розмірність словника і робимо його більш однорідним. Це дозволяє алгоритмам точніше виявляти семантичні зв'язки між словами та покращує якість подальшого аналізу тексту.

Процес лематизації включає в себе кілька етапів. Спочатку текст розбивається на окремі слова. Потім для кожного слова визначається його частина мови (іменник, дієслово тощо). Далі, використовуючи словник лем, алгоритм знаходить лему для кожного слова. Словники лем можуть бути різними за розміром і складністю, від простих списків до складних моделей, які враховують контекст слова в реченні. Лематизація дозволяє покращити точність і ефективність систем аналізу текстових даних, роблячи їх більш розумними та адаптивними.

2.2 Формування простору ознак

2.2.1 Мішок слів

Bag of Words (BoW) - це один із найпростіших і широко використовуваних методів для представлення текстових даних у вигляді числових векторів у задачах обробки природних мов (NLP).

В рамках методу "Bag of Words" текст (наприклад, документ або речення) перетворюється на набір слів без урахування граматики, порядку слів або їх структури. Кожне слово розглядається як окремий елемент, і його значення в контексті не враховується. Таким чином, структура тексту втрачається. Спочатку формується словник (vocabulary) всіх унікальних слів, що зустрічаються в текстах або колекціях документів. Кожному слову в словнику надається унікальний індекс. Кожен документ або текст представлений у вигляді вектора, де кожен елемент вектора відповідає кількості входжень конкретного слова з словника у даному тексті. Якщо слово зустрічається, то ставиться кількість його входжень, якщо не зустрічається — значення цього елемента нульове. На рисунку 2.2 наведено ілюстрацію роботи методу.

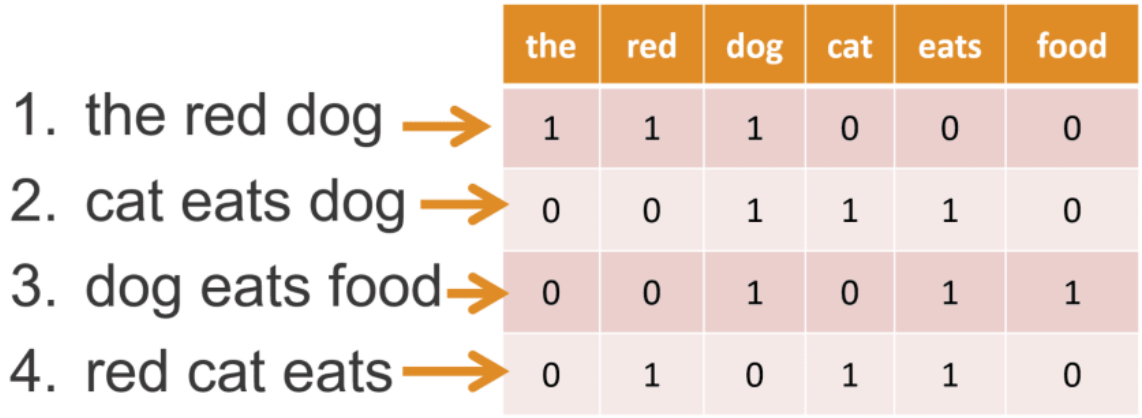

Рисунок 2.2 – Ілюстрація формування простору ознак з допомогою «мішка слів»

Особливості та обмеження методу:

- Ігнорування порядку слів. Оскільки метод не враховує порядок слів, важлива інформація, така як синтаксис і семантика, втрачається.

- Розмірність. Словник може стати дуже великим, особливо якщо текстові дані містять велику кількість унікальних слів. Це може призвести до дуже великих векторів, що є проблемою для обчислень.

- Відсутність контексту: Оскільки порядок слів не зберігається, неможливо врахувати контекст або зв'язок між словами.

2.2.2 Удосконалення методу «мішка слів» (TF-IDF)

З попереднього підрозділу очевидно, що хоч модель "мішок слів" (Bag of Words) є простим і ефективним способом представлення текстів у векторному вигляді, вона має свої обмеження. Розглянемо показник TF-IDF (Term Frequency-Inverse Document Frequency), що є вдосконаленням методу BoW, яке враховує не лише кількість входжень слова в документі, але й важливість цього слова у всій колекції документів, знижуючи вагу слів, які зустрічаються дуже часто в усіх текстах. Для цього спершу розглянемо окремі складові цього показника.

TF (Term Frequency) - це частота, з якою слово зустрічається в конкретному документі. Чим частіше слово використовується, тим вище значення TF. Даний показник розраховується за формулою:

$$TF = \frac{Number\ of\ instaces\ of\ word\ in\ the\ document}{total\ number\ of\ words\ in\ document}, \quad (2.1)$$

де в чисельнику кількість разів, яку слово зустрічається в документі, а в знаменнику – загальна кількість слів в документі.

Однак, TF не враховує того факту, що деякі слова можуть бути частими в усіх документах і, таким чином, менш корисними для розрізнення контекстів. Для усунення цього недоліку використовується показник - обернена частота документа IDF (Inverse Document Frequency), який знижує вагу частих слів, зважаючи на їх розповсюдженість у всіх документах колекції. Цей показник відображає, наскільки унікальним є слово в колекції документів. IDF обчислюється за формулою:

$$IDF = log\left(\frac{Total\ number\ of\ documents\ in\ the\ dataset}{Number\ of\ documents\ in\ the\ dataset\ which\ contain\ given\ word}\right), \quad (2.2)$$

де в чисельнику загальна кількість документів датасету, а в знаменнику – кількість документів, які містять задане слово.

Використання цього показника дозволяє знизити вагу загальних слів та підвищити вагу унікальних слів. Слова, які часто зустрічаються в багатьох документах (наприклад, "і", "а", "так"), мають низьке значення IDF. Слова, які зустрічаються рідко, мають високе значення IDF і вважаються більш важливими.

Схематично різницю між TF та IDF зображено на рисунку 2.3.

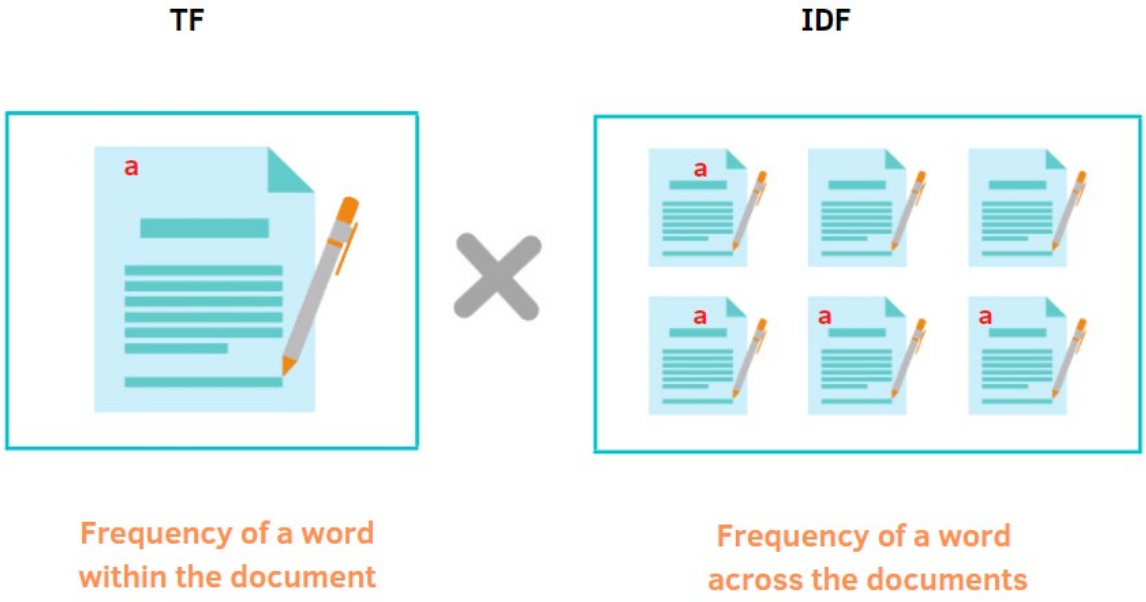

Рисунок 2.3 – Різниця між TF та IDF

TF-IDF – це комбінований показник, який поєднує в собі переваги TF та IDF. Він дозволяє оцінити важливість слова в контексті конкретного документа та всієї колекції документів:

$$TF\_IDF = TF \cdot IDF, \quad (2.3)$$

Слова з високим значенням TF-IDF є найбільш релевантними для документа і одночасно відрізняють цей документ від інших.

Поєднання TF та IDF дає більш точну оцінку важливості слів у текстах і дозволяє використовувати їх для класифікації, пошуку та інших задач в обробці текстових даних.

Хоча методи Bag of Words та TF-IDF мають свої обмеження, вони є відправною точкою для більш складних методів представлення текстів, таких як Word2Vec або BERT, які враховують контекст і семантику слів.

2.2.3 Формування простору ознак з врахуванням семантики

Word2Vec і BERT — це два потужні методи представлення текстових даних, які використовуються для моделювання семантики слів, враховуючи їх контекст. Вони є частиною новітніх підходів у обробці природних мов, які значно покращили точність моделей у порівнянні з простими методами, такими як Bag of Words чи TF-IDF.

Word2Vec — це метод, що використовує нейронні мережі для навчання векторних представлень слів. Основною ідеєю є те, що слова, які мають схожий контекст, матимуть схожі векторні представлення. Word2Vec працює за двома основними архітектурами: Skip-gram та Continuous Bag of Words (CBOW). У моделі Skip-gram кожне слово прогнозує його контекст (сусідні слова), а в CBOW — навпаки, контекст передбачає центральне слово. Це дозволяє створити щільні вектори, які відображають подібність між словами на основі їх вживання в контексті.

BERT (Bidirectional Encoder Representations from Transformers) — це значно складніша модель, яка була представлена компанією Google. Вона використовує архітектуру трансформерів і працює двосторонньо, тобто вона враховує не лише лівий, але й правий контекст слова в реченні. BERT навчається за допомогою двох основних задач: Masked Language Modeling (MLM) і Next Sentence Prediction (NSP). У MLM випадкові слова в реченні заміняються на спеціальний маркер, і модель має передбачити ці слова. У NSP модель вчиться передбачати, чи є два речення взаємно пов’язаними.

Відмінність між Word2Vec і BERT полягає в тому, що Word2Vec генерує одне фіксоване векторне представлення для кожного слова незалежно від

контексту, у якому воно вживається. Тобто слово "банк" буде мати один вектор, навіть якщо воно використовується у контексті фінансів чи річки. У BERT, натомість, слово "банк" матиме різні векторні представлення в залежності від того, чи йдеться про фінансову установу чи про річку.

BERT продемонстрував значні переваги в багатьох задачах обробки тексту, таких як класифікація текстів, аналіз настроїв, пошук інформації та питання-відповіді, оскільки враховує багатший контекст слів. Крім того, BERT використовує попередньо навчені моделі, які можна адаптувати до конкретних задач, що дозволяє значно зменшити час і ресурси, необхідні для навчання моделей.

У результаті, Word2Vec та BERT є важливими етапами в розвитку методів представлення тексту, де Word2Vec дозволяє ефективно працювати з великими корпусами текстів, а BERT дає набагато точніші та контекстуально обґрунтовані результати завдяки використанню двостороннього контексту та складніших архітектур нейронних мереж.

2.3 Методи балансування даних

Збалансований датасет — це набір даних, в якому кількість прикладів для кожної з категорій (класів) є приблизно рівною. Це важливо для задач машинного навчання, де неправильне балансування класів може призвести до того, що модель буде більше схилятися до більш представленого класу, ігноруючи менш представлений. У разі класифікації, де один клас значно більший за інший, модель може мати високі показники точності, але на практиці вона буде погано працювати на менш представленому класі.

Основні методи балансування датасету:

1. Over-sampling (надмірне вибіркове зразковування)

- SMOTE (Synthetic Minority Over-sampling Technique): Створення нових синтетичних прикладів для меншості, генеруючи їх між існуючими прикладами цього класу.
- Random Over-sampling: Просте випадкове копіювання зразків з меншості, щоб збільшити їх кількість.

2. Under-sampling (зменшення вибірки)

- Random Under-sampling: Випадкове зменшення кількості зразків з більшості класу, щоб зробити їх кількість рівною або більш схожою на клас меншості.
- Tomek Links: Видалення зразків, які мають "зв'язки" з найближчими зразками з протилежного класу (перехресні зразки).

3. Зважування класів (Class weighting). Призначення більших ваг класам меншості під час тренування, щоб модель більше звертала увагу на ці класи. Це можна зробити в алгоритмах, які підтримують параметр ваг класів, наприклад, у логістичній регресії або дереві рішень.

4. Гібридні методи - комбінація Over-sampling та Under-sampling. Використання поєднання методів, таких як спочатку зменшення більшості (under-sampling), а потім збільшення меншості (over-sampling).

5. Аугментація даних (Data Augmentation). Генерація нових прикладів на основі наявних даних шляхом застосування різних перетворень (наприклад, обертання, масштабування, відображення для зображень), що дозволяє створювати більше варіацій зразків для меншості.

6. Змішування класів (Ensemble methods). Використання ансамблів моделей для вирішення проблеми дисбалансу. Наприклад, BalancedRandomForest або EasyEnsemble — методи, які комбінують різні стратегії вибірки для кращої роботи з дисбалансованими датасетами.

7. Cost-sensitive Learning. Алгоритми, які враховують "вартість" помилок різних класів. Це дозволяє моделі надавати більшу увагу менш представленим класам без зміни кількості зразків.

8. Cluster-based Over-sampling. Створення нових зразків меншості на основі кластеризації даних (наприклад, використовуючи K-means), а потім генерування синтетичних зразків навколо центрів кластерів.

Ці методи можна комбінувати в залежності від характеристик датасету, щоб досягти найкращих результатів у вирішенні проблеми дисбалансу класів.

2.4 Моделі машинного навчання

Виявлення спаму стало критично важливим напрямком досліджень у сфері кібербезпеки та фільтрації інформації через постійну еволюцію тактики спамерів. Традиційні системи, засновані на правилах, такі як чорні списки та евристичні підходи, виявилися недостатніми для боротьби зі складнощами сучасного спаму. Методи машинного навчання (ML), що використовують здатність адаптуватися та навчатися на основі даних, стали потужним інструментом для підвищення ефективності та точності виявлення спаму.

В [10] проведено ретельний огляд наукових джерел виявлення спаму методами машинного навчання. Одним з основних підходів до виявлення спаму на основі ML є алгоритми класифікації, такі як машини опорних векторів (SVM), наївний Байєс і випадкові ліси [11,12]. Ці методи використовують ознаки, витягнуті з контенту електронної пошти або повідомлень, такі як частота слів, метадані та структурні патерни, щоб класифікувати повідомлення як спам або легітимні.

Моделі глибокого навчання, такі як згорткові нейронні мережі (CNN) та рекурентні нейронні мережі (RNN), ще більше просунули цю галузь, усунувши обмеження в інженерії функцій. Ці моделі автоматично витягують значущі патерни з необробленого тексту, фіксуючи як локальні, так і контекстні залежності [13]. Використання CNN для вилучення ознак у поєднанні з RNN, такими як LSTM, для послідовної обробки даних призвело до значного покращення виявлення спаму у великих масивах даних.

Ансамблеві методи, які об'єднують прогнози з декількох моделей ML, також широко застосовуються для виявлення спаму. Такі методи, як Bagging, Boosting і stacking, поєднують сильні сторони окремих класифікаторів для досягнення чудової продуктивності [14]. Інтегруючи такі алгоритми, як випадкові ліси та градієнтні дерева, ці методи забезпечують стійкість до перенастроювання та високу швидкість відгуку, що є важливим для практичних систем виявлення спаму.

Розглянемо детальніше найбільш популярні моделі для виявлення спаму.

2.4.1 Модель наївного Байєса

Модель наївного Байєса (Naïve Bayes) — це один із базових та ефективних методів машинного навчання для задач класифікації. Основою моделі є теорема Байєса, яка дозволяє обчислювати ймовірність належності зразка до певного класу, враховуючи розподіл його характеристик. Попри свою назву, модель називається "наївною", оскільки припускає, що всі ознаки незалежні одна від одної, що в реальних даних часто не відповідає дійсності. В основі моделі лежить формула Байєса про перерахунок ймовірності події А (апостеріорної ймовірності) за виконання додаткової умови (події В)

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \tag{2.4}$$

де $P(B|A)$ – умовна ймовірність події В за уиови, що відбулась подія А,

$P(A), P(B)$ – апріорні ймовірності подій А і В.

Для задачі класифікації формула (2.4) має вигляд:

$$P(\mathrm{c}|X) = \frac{P(X|y)P(y)}{P(X)}, \tag{2.5}$$

де c – вихід алгоритму класифікації, змінна класу, що залежить від X

$X = (x_1, x_2, \dots, x_n)$ - вектор ознак, які приймаються взаємно-незалежними одна від одної.

Припущення про незалежність ознак дозволяє записати формулу Байєса як:

$$P(\mathrm{c}|x_1, x_2, \dots, x_n) = \frac{P(x_1|\mathrm{c})P(x_2|\mathrm{c}) \dots P(x_n|\mathrm{c})P(\mathrm{c})}{P(x_1)P(x_2) \dots P(x_n)}, \tag{2.6}$$

В класифікаторі наївного Байєса знаменник відкидається, оскільки він однаковий для всіх класів

Алгоритм методу наївного Байєса виглядає наступним чином:

1. Нехай навчальна множина містить k класів: $\mathrm{c}_1, \mathrm{c}_2, \ldots, \mathrm{c}_k$, а простір ознак X містить n значень: $x_1, x_2, \ldots, x_n$.
2. Необхідно обчислити апріорні ймовірності класів $P(\mathrm{c}_i),\ i = \overline{1,k}$ та умовні ймовірності ознак для кожного з класів $P(x_j|\mathrm{c}_i),\ j = \overline{1,n},\ i = \overline{1,k}$.
3. Для деякого нового об'єкта з набором ознак $x_1^*, x_2^*, \ldots, x_n^*$ для кожного з класів $\mathrm{c}_1, \mathrm{c}_2, \ldots, \mathrm{c}_k$ обчислити значення, пропорційні до апостеріорних ймовірностей $P(\mathrm{c}_i|x_1^*, x_2^*, \ldots, x_n^*)$:

$$P(\mathrm{c}_i)P(x_1 = x_1^*|\mathrm{c}_i)P(x_2 = x_2^*|\mathrm{c}_i) \ldots P(x_n = x_n^*|\mathrm{c}_i),\ i = \overline{1,k}$$

4. Новому об'єкту присвоїти клас з максимальним значенням обчисленої у кроці 3 величини

$$P(\mathrm{c}_i) \cdot \prod_{j=1}^{n} P(x_j = x_j^*|class = \mathrm{c}_i), \tag{2.7}$$

Залежно від типу даних, модель може бути реалізована кількома способами:

1. Гаусівський наївний Байєс - використовується для безперервних даних, припускаючи нормальний розподіл кожної ознаки.

2. Мультиноміальний наївний Байєс - ефективний для текстових даних, зокрема задач класифікації документів чи виявлення спаму, де ознаки є частотами чи кількісними значеннями.

3. Наївний Байєс на основі розподілу Бернуллі - підходить для задач з бінарними ознаками, наприклад, аналізу присутності/відсутності ключових слів у тексті.

Головною перевагою є її простота та швидкість. Модель легко впроваджувати, вона не вимагає великої кількості обчислень і добре працює навіть на невеликих наборах даних. Крім того, наївний Байєс є ефективним для високоразмірних даних (наприклад, текстових), завдяки здатності обробляти великий обсяг ознак.

Основним недоліком моделі є припущення про незалежність ознак, яке рідко відповідає реальним даним. У випадках, коли між ознаками існують сильні залежності, продуктивність моделі може значно знижуватися. Також модель не

може враховувати складні взаємозв’язки між ознаками, які часто присутні в задачах із реальними даними.

2.4.2 Модель опорних векторів (SVM)

Модель опорних векторів (SVM) — це один із потужних і ефективних методів машинного навчання, особливо популярний для задач класифікації та регресії. SVM базується на пошуку оптимальної гіперплощини, яка розділяє дані двох класів із найбільшим можливим відступом (margin) між класами. Такий підхід робить модель стійкою до шуму в даних і забезпечує високу узагальнюючу здатність.

Основна ідея SVM полягає в пошуку гіперплощини у багатовимірному просторі, яка найкраще розділяє дані. Гіперплощина визначається рівнянням:

$$w \cdot x + b = 0 \tag{2.8}$$

де w — вектор ваг, x — вектор даних, а b — зміщення. Головна мета — максимізувати відступ (margin) між гіперплощиною та найближчими точками кожного класу, які називаються опорними векторами.

На рисунку 2.4 зображено геометричну інтерпретацію роботи алгоритму опорних векторів (Support Vector Machine, SVM). Червоні та чорні точки представляють різні класи даних, які алгоритм намагається розділити. Суцільна лінія є оптимальною гіперплощиною (у двовимірному просторі - прямою), яка розділяє дані на два класи з максимальною відстанню. Ця відстань називається відступом (margin). Точки, що лежать найближче до гіперплощини і визначають відступ, називаються опорними векторами (support vectors). Їхнє розташування є критичним для побудови оптимальної гіперплощини. Штрихові лінії паралельні до гіперплощини і проходять через опорні вектори, визначаючи межі відступу. Алгоритм SVM прагне знайти гіперплощину, яка максимізує цей відступ, з метою отримання більш узагальненої моделі, яка краще класифікує нові дані.

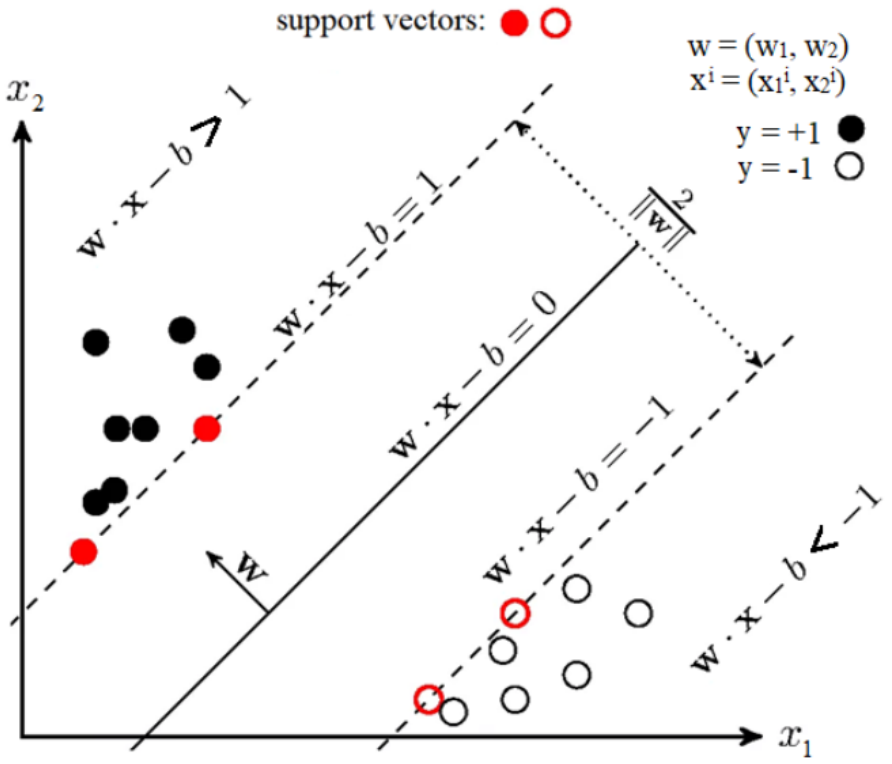

Рисунок 2.4 – Графічне представлення роботи методу опорних векторів

Для лінійно розділених даних SVM обирає одну гіперплощину з безлічі можливих, яка має максимальний відступ. Якщо дані нелінійно розділені, використовується ядровий трюк (kernel trick), який дозволяє перетворити вихідний простір ознак у простір вищої розмірності, де дані можуть стати лінійно розділеними. Популярні ядра включають:

- лінійне ядро,
- поліноміальне ядро,
- радіальне базисне функціональне (RBF) ядро,
- сигмоїдне ядро.

Основними перевагами SVM є ефективність у високорозмірних просторах, стійкість до перенавчання, особливо в задачах із обмеженою кількістю зразків, можливість використання різних ядер для адаптації до різних типів даних.

Головними недоліками SVM є обчислювальна складність, необхідність вибору ядра та гіперпараметрів, чутливість до шуму.

Модель опорних векторів є потужним інструментом для класифікації завдяки своїй здатності працювати в складних просторах ознак і забезпечувати високу узагальнюючу здатність. Незважаючи на свої обмеження, SVM

залишається актуальною у багатьох сферах, зокрема у задачах із високими вимогами до точності. Завдяки розширенням і комбінаціям із іншими методами (наприклад, ансамблевими), модель зберігає свою популярність у машинному навчанні.

2.4.3 Випадкові ліси

Випадкові ліси (Random Forests) — це потужний ансамблевий метод машинного навчання, який використовується для задач класифікації та регресії. Основою методу є концепція побудови великої кількості рішень дерев і комбінування їхніх результатів для досягнення більшої точності та стабільності. Випадкові ліси є вдосконаленням алгоритму "Bagging" (Bootstrap Aggregating), який передбачає створення різних моделей на підмножинах даних, обраних із заміною.

Ключовою особливістю випадкових лісів є додаткова випадковість у виборі розбиття вузлів дерев. На відміну від звичайних дерев рішень, які шукають найкращий розподіл для кожного вузла, випадкові ліси розглядають лише випадкову підмножину ознак на кожному вузлі. Це допомагає зменшити кореляцію між окремими деревами, роблячи ансамбль більш стійким до перенавчання. Кожне дерево в лісі працює незалежно, і його прогноз вносить внесок у фінальний результат.

Процес створення випадкових лісів починається з вибору кількості дерев (гіперпараметр), які будуть побудовані. Кожне дерево отримує свій власний набір даних для навчання шляхом вибірки з початкового набору із заміною. Далі, на кожному вузлі дерева випадковим чином вибирається підмножина ознак, з яких обирається найкращий розподіл. У підсумку результати кожного дерева агрегуються: для класифікації використовується правило голосування (більшість голосів), а для регресії — середнє значення.

Випадкові ліси мають значну перевагу у стійкості до шуму та викидів у даних. Вони можуть ефективно працювати навіть на наборах даних із пропущеними значеннями та не вимагають значної попередньої обробки.

Завдяки своїй ансамблевій природі, метод добре справляється із задачами, де окремі дерева могли б перенавчатися або демонструвати нестабільність.

Однак випадкові ліси також мають певні обмеження. Вони можуть бути повільними для навчання, особливо при великій кількості дерев або високій розмірності даних. Хоча модель і забезпечує високу точність, вона менш інтерпретована порівняно з окремими деревами рішень, оскільки агреговані результати важче пояснити. Крім того, обчислювальна складність може зрости, якщо кількість ознак чи об’єктів дуже велика.

У реальних задачах випадкові ліси знаходять застосування в багатьох галузях, включаючи медицину, фінанси, біоінформатику та маркетинг. Вони широко використовуються для аналізу великих даних завдяки своїй здатності виявляти важливі ознаки, що впливають на результати. Зокрема, метод дозволяє оцінювати важливість кожної ознаки, що є цінним для задач, де інтерпретація важлива.

Попри конкуренцію з боку сучасних глибоких моделей, випадкові ліси залишаються популярними через свою простоту у використанні, ефективність і високу точність на багатьох типах задач. Завдяки можливості паралельного обчислення та впровадженню в сучасні бібліотеки, такі як Scikit-learn, вони є доступними навіть для новачків у машинному навчанні. Це робить їх одним із найважливіших інструментів у сучасному наборі алгоритмів машинного навчання.

2.4.4 Нейронні мережі

Глибоке навчання стало революційним кроком у розвитку штучного інтелекту завдяки здатності моделювати складні залежності та виявляти структури в даних. Серед найпоширеніших моделей глибокого навчання виділяються згорткові нейронні мережі (CNN) і рекурентні нейронні мережі (RNN), які ефективно справляються з обробкою зображень та послідовних даних відповідно.

Згорткові нейронні мережі (CNN) найчастіше застосовуються для обробки зображень. Основна ідея CNN полягає у використанні згорткових шарів, які

автоматично виділяють значущі особливості з даних. Кожен згортковий шар використовує фільтри для аналізу локальних фрагментів зображень, виявляючи такі елементи, як краї, текстури чи форми. Завдяки шаруватій структурі CNN здатні будувати ієрархічні представлення, поступово переходячи від базових особливостей до високорівневих концепцій, таких як обличчя чи об'єкти.

Рекурентні нейронні мережі (RNN) зосереджені на обробці послідовних даних, таких як текст, аудіо чи часоряди. Особливістю RNN є здатність зберігати інформацію про попередні кроки, використовуючи внутрішні цикли. Це дозволяє моделі враховувати контекст і залежності між елементами послідовності. Для довгих послідовностей використовуються спеціальні модифікації RNN, такі як LSTM (довга короткострокова пам’ять) і GRU (уніфікована рекурентна одиниця), які розв’язують проблему затухання градієнта.

На рисунку 2.5 показані загальні структури згорткових нейронних мереж (CNN) і рекурентних нейронних мереж (RNN). Для CNN (ліворуч) відображено вхідний шар, шари згортки або пулінгу (позначені фіолетовими колами), а також повнозв’язні шари та вихідний шар. Ця архітектура спрямована на обробку просторових даних, наприклад, зображень. Для RNN (праворуч) зображено вхідний шар, шари із пам’яттю (зелені кола) і вихідний шар, що ілюструє здатність RNN враховувати послідовності та залежності в часі. Малюнок чітко демонструє ключові особливості кожної архітектури, підкреслюючи різницю в підході до обробки даних.

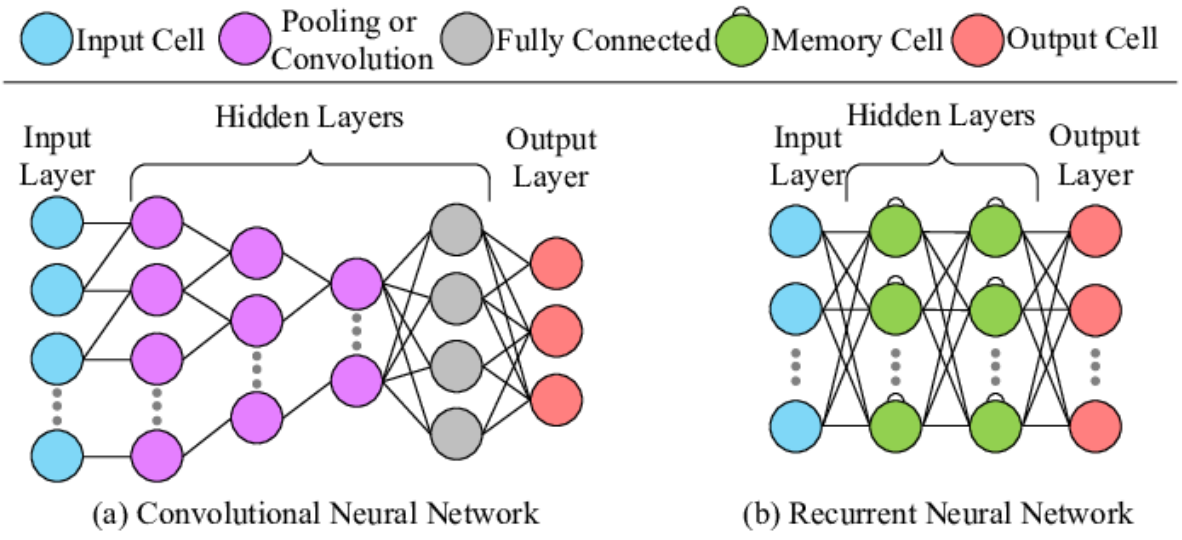

Рисунок 2.5 – Різниця між CNN та RNN

CNN знайшли широке застосування у задачах розпізнавання зображень, медичної діагностики за візуальними даними та автоматичного водіння. Наприклад, у медичній сфері вони використовуються для виявлення ракових клітин на рентгенівських знімках. Водночас RNN активно застосовуються для машинного перекладу, розпізнавання мовлення, аналізу настроїв у текстах та прогнозування часових рядів, наприклад, фінансових даних.

Обидва типи нейронних мереж працюють з великим обсягом даних і потребують значних обчислювальних ресурсів. Їхній успіх значною мірою залежить від якісної архітектури та гіперпараметрів. Хоча CNN і RNN мають свої сфери домінування, сучасні підходи часто комбінують їх у гібридних архітектурах, наприклад, для автоматичного опису зображень текстом.

Глибокі нейронні мережі, такі як CNN і RNN, продовжують змінювати багато галузей завдяки своїй здатності адаптуватися до складних і різноманітних задач. Їхній потенціал лише зростає, особливо з розвитком потужних обчислювальних платформ і оптимізованих алгоритмів навчання.

Методи машинного навчання є важливими для задачі виявлення спаму, оскільки вони дозволяють автоматично виявляти складні шаблони та закономірності в текстах листів, адаптуватися до нових тактик спамерів і обробляти великі об'єми даних. Завдяки здатності навчатися на великих наборах даних, ці методи забезпечують високу точність, знижують кількість помилкових спрацьовувань і дозволяють ефективно обробляти нові види спаму. Вони також можуть враховувати персональні уподобання користувача, створюючи адаптовані моделі для кожного індивідуального випадку, що робить системи виявлення спаму більш ефективними і гнучкими.

РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ СПАМУ

3.1 Вибір середовища

Python визнаний одним із найкращих інструментів для реалізації методів машинного навчання (ML) завдяки своїй простоті, багатству бібліотек і широкій спільноті користувачів. Його синтаксис легкий для освоєння, що дозволяє дослідникам і розробникам швидко адаптуватися до вирішення складних задач. Ця мова особливо популярна у сферах обробки даних, аналізу та побудови моделей штучного інтелекту.

Одна з ключових переваг Python полягає у наявності потужних бібліотек для ML, таких як TensorFlow, PyTorch, scikit-learn, Keras та NumPy.

На рисунку 3.1 зображено взаємозв'язки між популярними бібліотеками для машинного навчання та обробки даних у мові програмування Python.

Рисунок 3.1 – Популярні бібліотеки Python для машинного навчання

В центрі розташовані TensorFlow та Keras, які є одними з найпоширеніших фреймворків для глибокого навчання. Інші бібліотеки, такі як Scikit-learn, Numpy, Pandas, PyTorch, забезпечують додаткові функціональні можливості для роботи з даними, статистичного аналізу та побудови моделей машинного

навчання. NLTK спеціалізується на обробці природної мови, а Spark - на обробці великих даних. Ця екосистема бібліотек пропонує широкий спектр інструментів для вчених даних та розробників, що працюють у галузі штучного інтелекту та машинного навчання.

Ці інструменти забезпечують широкий спектр функціональності, починаючи від попередньої обробки даних і закінчуючи навчанням складних глибоких нейронних мереж. Крім того, Python підтримує інтеграцію з іншими мовами, такими як C++ чи R, що дозволяє поєднувати швидкість обчислень із зручністю програмування.

На противагу Python, мови, як-от R, Julia або Java, мають свої сильні сторони, але вони не досягають такого рівня універсальності. R, наприклад, широко використовується в статистиці та аналізі даних, проте її можливості для глибокого навчання обмежені порівняно з Python. Julia є перспективною мовою, яка забезпечує високу швидкість обчислень, але її екосистема бібліотек ще не настільки розвинена. Java, хоч і використовується для корпоративних додатків, має більш громіздкий синтаксис і не така гнучка для швидкої розробки моделей.

Python також виграє завдяки інтеграції з інтерактивними середовищами, такими як Jupyter Notebook. Це дозволяє дослідникам комбінувати код, візуалізації та текстову документацію в одному файлі, що зручно для аналізу даних і презентації результатів. Інші середовища, наприклад MATLAB, мають схожу функціональність, але їхня комерційна ліцензія робить їх менш доступними для широкого використання.

За даними опитування Stack Overflow 2023, Python займає перше місце серед мов програмування, які найчастіше використовуються в машинному навчанні та аналізі даних. Близько 50% розробників обирають Python для своїх проектів у галузі ML, тоді як найближчий конкурент R використовується лише в 6% випадків. Крім того, популярність Python серед початківців робить його ідеальним вибором для навчання та розвитку нових спеціалістів.

Екосистема Python активно підтримується глобальною спільнотою. Вона включає тисячі ресурсів, таких як онлайн-курси, форуми та документація, що значно спрощує процес навчання та вирішення технічних питань. Багато

компаній і дослідницьких центрів також створюють власні інструменти на базі Python, розширюючи його можливості.

Незважаючи на деякі недоліки, як-от відносно нижча швидкість обчислень порівняно з C++ або Julia, Python компенсує це своєю універсальністю та простотою інтеграції зі швидшими мовами. Завдяки цьому він залишається лідером у сфері ML, забезпечуючи розробників усіма необхідними інструментами для ефективної роботи.

Таким чином, Python є ідеальним середовищем для реалізації методів машинного навчання завдяки своїй зручності, багатству бібліотек та підтримці спільноти. Його популярність і універсальність роблять цю мову першим вибором для більшості розробників та дослідників у галузі штучного інтелекту.

3.2 Огляд бібліотек та деяких основних функцій реалізації

На першому етапі нам потрібно завантажити необхідні бібліотек. В лістингу 3.1 наведено приклад завантаження усіх необхідних бібліотек для попередньої обробки, а в лістингу 3.2 для – створення моделей

Лістинг 3.1 – Імпорт необхідних бібліотек для попередньої обробки

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
# data preprocessing
import nltk
nltk.download('stopwords')
nltk.download('punkt')
import string
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.tokenize import sent_tokenize, word_tokenize
import re
from collections import Counter
from wordcloud import WordCloud
from sklearn.preprocessing import LabelEncoder
```

Лістинг 3.2 – Імпорт необхідних бібліотек для створення моделей

```
    # Model Building
    from sklearn.feature_extraction.text import TfidfVectorizer
    from sklearn.model_selection import train_test_split
    from sklearn.metrics import
accuracy_score,confusion_matrix,precision_score
    from sklearn.svm import SVC
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.naive_bayes import MultinomialNB
```

Коротко опишемо основні з цих бібліотек:

- pandas - використовується для роботи з табличними даними (DataFrame), має зручний інтерфейс для маніпуляцій з даними, таких як завантаження, фільтрація, агрегація та перетворення даних;
- matplotlib.pyplot - бібліотека для створення графіків і візуалізацій даних, таких як діаграми, гістограми та лінійні графіки;
- - seaborn - розширення для matplotlib, яке спрощує створення більш складних і стильних візуалізацій, особливо для статистичних графіків;
- nltk - бібліотека для обробки природної мови, яка включає інструменти для токенізації, стемінгу, видалення стоп-слів та багато іншого;
- string - бібліотека для роботи з рядками, зокрема для доступу до стандартних символів, таких як букви, цифри і пунктуація;
- re - бібліотека для роботи з регулярними виразами, використовується для пошуку і заміни патернів у тексті;
- sklearn (Scikit-learn) - в Python надає набір простих у використанні інструментів для побудови та оцінки моделей машинного навчання, включаючи класифікацію, регресію, кластеризацію, зниження розмірності та попередню обробку даних;
-
- sklearn.preprocessing – бібліотека в Python надає інструменти для підготовки та масштабування даних, такі як нормалізація, стандартизація,

кодування категоріальних змінних та інші методи попередньої обробки для моделей машинного навчання;

- sklearn.feature_extraction – бібліотека в Python надає інструменти для перетворення сирих даних, таких як текст або зображення, у числові ознаки, які можуть бути використані в моделях машинного навчання;
- sklearn.metrics — це бібліотека в Python, яка надає функції для оцінки якості моделей машинного навчання, зокрема для обчислення таких метрик, як точність, відновлення, F1-міра та інші.

Після завантаження всіх бібліотек необхідно завантажити датасет та виконати попередню обробку даних. В лістингу 3.3 продемонстровано функцію для очищення тексту від пунктуації, видалення стоп-слів та лематизації

Лістинг 3.3 – Функція, що виконує попередню обробку тексту

```
    def preprocess_text(text):
        # Remove punctuation
        no_punctuation = ''.join([char for char in text if char
not in string.punctuation])
        # Lowercase the text
        no_punctuation_lower = no_punctuation.lower()
        # Tokenize the text into words
        words = nltk.word_tokenize(no_punctuation_lower)
        # Remove stopwords and non-alphabetic characters, and
lemmatize
        lemmatizer = WordNetLemmatizer()
        lemmatized_words = [lemmatizer.lemmatize(word) for word in
words if word.lower() not in stopwords.words('english') and
word.isalpha()]
        # Join the lemmatized words back into a sentence
        lemmatized_text = ' '.join(lemmatized_words)
        return lemmatized_text
```

Лістинг 3.4 демонструє код для поділу датасету на навчальну та тестову вибірки:

Лістинг 3.4 – Поділ датасету на навчальну та тестові вибірки

```
# Split the dataset to train and test sets
x_train, x_test, y_train, y_test = train_test_split(
    text_tfidf, spam_df["label"], test_size=0.2)
```

В лістингу 3.5 наведено функцію для оцінки різних параметрів точності класифікаційної моделі

Лістинг 3.5 – Оцінка точності класифікаційної моделі

```
def evaluate_model(model, x_test, y_test, model_name="Model"):
    # Predict labels on testing data
    y_pred = model.predict(x_test)
    # Calculate confusion matrix
    conf_matrix = confusion_matrix(y_test, y_pred)
    # Extract TP, TN, FP, FN
    TN, FP, FN, TP = conf_matrix.ravel()
    # Calculate various performance metrics
    accuracy = accuracy_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)
    # Calculate mean accuracy using cross-validation
    mean_accuracy = cross_val_predict(model, x_train, y_train,
cv=10).mean()
    classification_rep = classification_report(y_test, y_pred,
output_dict=True)
    metrics = {
        "Accuracy": accuracy,
        "Recall": recall,
        "Precision": precision,
        "F1-score": f1,
        "Mean Accuracy": mean_accuracy,
        "Classification Report": classification_rep
    }
  return metrics,y_pred
```

В лістингу 3.6 наведено приклад застосування класифікаційної моделі та оцінки її точності:

Лістинг 3.6 – Приклад використання Random Forest

```
# Train Random Forest model
randomForest_model = RandomForestClassifier()
randomForest_model.fit(x_train, y_train)
# Evaluate Random Forest model
rf_metrics,y_pred_rf = evaluate_model(randomForest_model,
x_test, y_test, model_name="Random Forest")
```

Звісно, що повний лістинг програми – набагато складніший, в даному підрозділі наведені лише основні функції.

3.3 Опис наборів даних

Для дослідження спамових листів використовуються наступні набори відкритих даних: TREC 2007 spam Dataset [15] та Enron-Spam dataset [16].

Перший згаданий датасет містить 50199 спамових (spam) і 25220 нормальних листів (ham).

Датасет містить 5 стовпців:

- мітку;
- тему листа;
- відправника;
- отримувача;
- текст листа.

Повідомлення датасету отримані з певного сервера, що були доставлені між датами: Sun, 8 Apr 2007 13:07:21 -0400 та Fri, 6 Jul 2007 07:04:53 -0400

На рисунку 3.2 наведено виконання команди data.head() для згаданого датасету

	label	subject	email_to	email_from	message
0	1	Generic Cialis, branded quality@	the00@speedy.uwaterloo.ca	"Tomas Jacobs" <RickyAmes@aol.com>	Content-Type: text/html; \nContent-Transfer-Enc...
1	0	Typo in /debian/ README	debian-mirrors@lists.debian.org	Yan Morin <yan.morin@savoirfairelinux.com>	Hi, i've just updated from the gulus and I che...
2	1	authentic viagra	<the00@plg.uwaterloo.ca>	"Sheila Crenshaw" <7stocknews@tractionmarketin...	Content-Type: text/plain; \n\tcharset="iso-8859...
3	1	Nice talking with ya	opt4@speedy.uwaterloo.ca	"Stormy Dempsey" <vqucsmdfgvsg@ruraltek.com>	Hey Billy, \n\nit was really fun going out the...
4	1	or trembling; stomach cramps; trouble in sleep...	ktwarwic@speedy.uwaterloo.ca	"Christi T. Jernigan" <dcube@totalink.net>	Content-Type: multipart/ alternative;\n ...

Рисунок 3.2 – Перші 5 записів датасету TREC 2007 Spam Dataset

Інший набір даних Enron Spam Dataset містить електронні листи, що походять з корпоративної електронної пошти компанії Enron. Цей датасет містить 17171 спамових (spam) і 16545 нормальних листів (ham).

Датасет містить 5 стовпців:

- мітку;
- тему листа;
- дату;
- текст листа.

Оскільки оригінальні набори даних цього датасету записані таким чином, що кожен лист знаходиться в окремому txt-файлі, розподіленому в декількох каталогах. В даному випадку читання даних є громіздким та ускладненим необхідністю формування структури датафрейму з багатьох файлів [17]. Лістинг коду наведено в додатку Б.

Обидва датасети є важливим ресурсом для розробки та тестування технологій фільтрації спаму. Зібрані на основі реальних електронних листів, ці дані дозволяють працювати з складними текстами, включаючи рекламу, фішинг та інші небажані елементи, що часто зустрічаються в спамі.

3.4 Результати тестування

Загальний підхід до проведення експериментальних досліджень стосовно використання методів машинного для задач виявлення спаму було використано підхід наведений на рисунку 3.3.

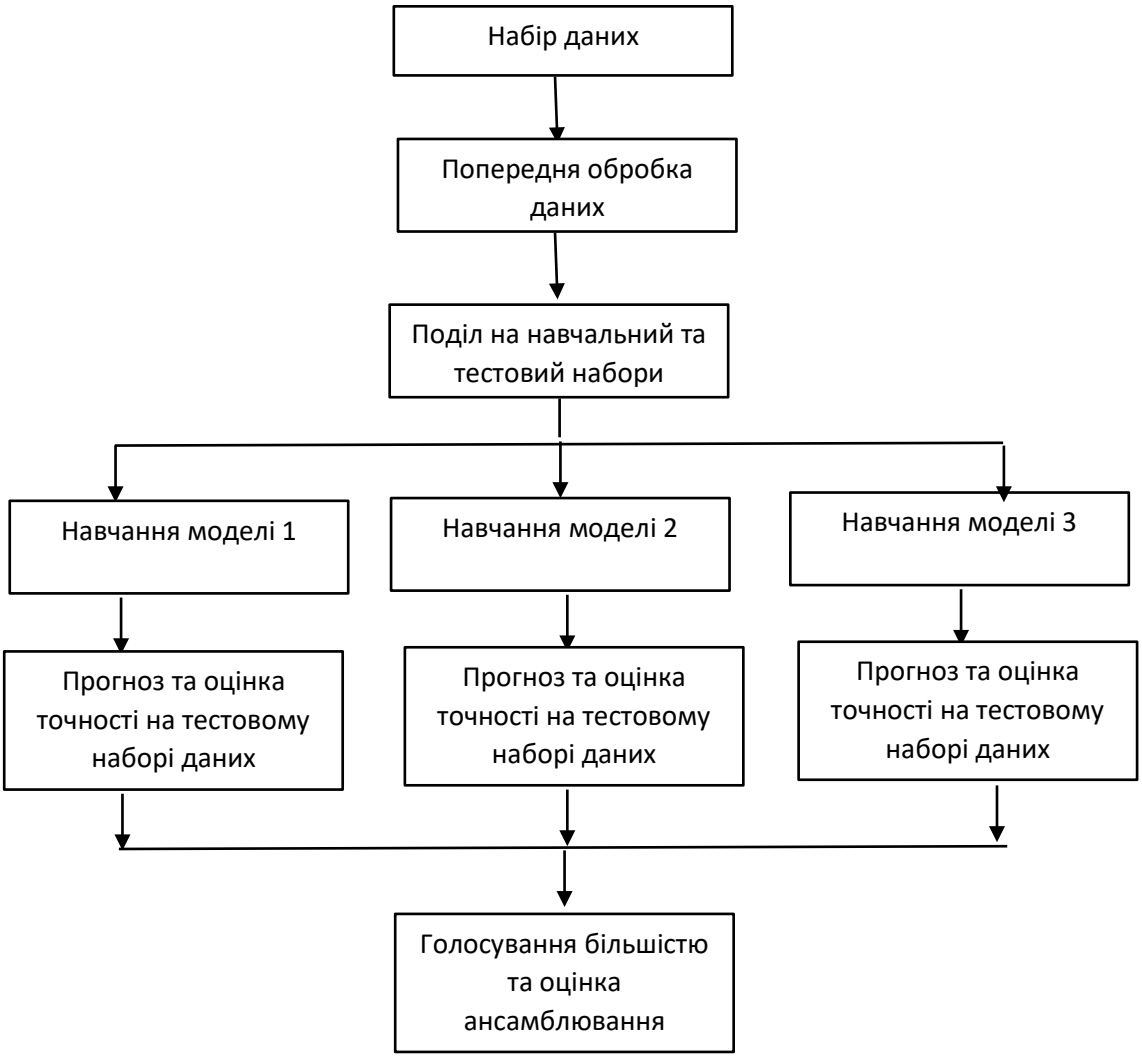

Рисунок 3.3 -Ілюстрація підходу щодо оцінки методів машинного навчання для задач виявлення спаму

Збалансованість даних є важливою для успішної побудови класифікаційних моделей, оскільки вона значно впливає на ефективність та точність моделі. Коли класові мітки (наприклад, позитивні та негативні приклади в задачі класифікації) не розподілені рівномірно, модель може мати проблеми з навчанням і неправильно оцінювати деякі категорії. Два набори даних містять порівняну кількість спамових та нормальних листів, тому додаткові методи балансування даних не проводились.

Для тестування методів машинного навчання було обрано три моделі: модель наївного Байєса, методу опорних векторів та випадкового лісу. Ансамблювання проводилось методом більшості голосів з допомогою `VotingClassifier`.

Результати тестування моделей наведено в таблиці 3.1

Таблиця 3.1 – Результати тестування моделей для двох обраних наборів даних

Датасет	Моделі	Точність /Accuracy	Влучність/ Precision	Повнота/ Recall	F1
TREC 2007 Spam Dataset	Ансамблювання	0,94	0,94	0,93	0,93
	Модель Байєса	0,82	0,8	0,8	0,8
	SVM	0,85	0,83	0,8	0,81
	Випадковий ліс	0,93	0,93	0,92	0,92
Enron-Spam dataset	Ансамблювання	0,99	0,98	0,98	0,98
	Модель Байєса	0,97	0,97	0,96	0,96
	SVM	0,98	0,97	0,97	0,97
	Випадковий ліс	0,98	0,98	0,98	0,98

З результатів очевидно, що точність виявлення спаму залежить не лише від використаних моделей, а й від даних, на яких ми тестуємо датасет

З таблиці можна зробити наступні висновки, що ефективність ансамблювання більш помітна на TREC 2007, де різниця між моделями більша, і є можливість покращити результати.

Ансамблювання ефективне, особливо на датасетах, де окремі моделі демонструють різну продуктивність. Це підтверджує доцільність його використання для підвищення точності класифікації.

Для подальших досліджень доцільно було б сконцентруватись на удосконаленні методів класифікації шляхом використання оптимізації гіперпараметрів (RandomizedSearchCV, GridSearchCV) та застосуванням методів глибокого навчання.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці

У кваліфікаційній роботі магістра спроектовано систему для дослідження використання методів машинного навчання для виявлення спаму. Під час розв’язання задач дослідження, особливо практичної реалізації системи, враховано вимоги з охорони праці і техніки безпеки, пожежної та електробезпеки.

Виконання як теоретичної частини роботи, так і практичної, передбачає використання комп’ютерної техніки та обладнання з низькими напругами і силою струму. Зокрема, в якості блоку живлення плати ESP8266, використовувалась напруга живлення, яка становить 5 В. На платі використовуються можливі номінали напруги на рівні 5 В і 3,3 В, що не становить небезпеки для користувачів та розробника системи.

В якості регламентуючого документа з пожежної безпеки перед початком роботи над комп’ютерною системою для контролю параметрів мікроклімату теплиць використано вимоги «Типового положення про інструктажі, спеціальне навчання та перевірку знань з питань пожежної безпеки на підприємствах, в установах та організаціях України», які є діючим на даний час і затверджені постановою Кабінету міністрів України від 26 червня 2013 р. № 444.

Для організації захисту від негативного впливу екранів дотримано вимог Закону України "Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями" та НПАОП 0.00-7.15-18, який затверджений наказом Міністерства соціальної політики України 14.02.2018 N207. Робоче місце під час виконання кваліфікаційної роботи та проектування комп’ютерної системи облаштовано згідно наведених вимог та відповідає організаційним, ергономічним та вимогам з пожежної безпеки.

Електробезпеку робочого місця регламентують Правила безпечної експлуатації електроустановок споживачів, які затверджені наказом Держнаглядохоронпраці від 09.01.98 N 4, зареєстрованих у Міністерстві юстиції

України 10.02.98 за N 93/2533 (НПАОП 40.1-1.21-98). Електромережа, яка використовувалася при виконанні кваліфікаційної роботи магістра, відповідає правилам [23]:

– живлення електромережі проектовано, як окрему групову трьох провідну мережу з використанням фази, робочого «нуля» та захисного «нуля»;

– захисний «нуль» застосовано для реалізації заземлення електропристроїв;

– усі електричні та електронні пристрої мають захист від короткого замикання та непередбачуваних аварійних ситуацій;

– монтаж та експлуатація електромережі задовольняють вимогам щодо унеможливлення виникнення джерела загоряння через коротке замикання та перевантаження;

– усі лінії електроживлення виконанні не з легкозаймистого матеріалу або з негорючою ізоляцію;

– електричне устаткування підключено до мережі лише за допомогою справних штепсельних з’єднань і розеток заводського виготовлення;

– у розетках і штепселях передбачено контакти заземлення.

Вимоги електробезпеки при проектуванні компонентів комп’ютерної системи для контролю параметрів мікроклімату теплиць дотримано двома шляхами: використання безпровідних технологій передавання даних і напруги живлення в діапазоні 3,3В і 5 В, що дозволяє зменшити можливість ураження струмом при виникненні контакту з мережею чи в аварійних ситуаціях.

Щодо пожежної безпеки будівлі, де виконувався проект, то дотримано вимоги державних будівельних норм "Пожежна безпека об'єктів будівництва", які затверджені наказом Держбуду України від 03.12.2002 N 88, а також вимоги правил пожежної безпеки України, затвердженими наказом Міністерства України з питань надзвичайних ситуацій від 19.10.2004 N 126.

У приміщеннях, де розташовуються робочі місця користувачів ПК потрібно забезпечити відповідність вимогам санітарних норм і правил наведених у ДСанПіН 3.3.2-007-98. Крім цього, на робочих місцях, обладнаних комп’ютерами і периферійною технікою забезпечено оптимальні значення

параметрів мікроклімату: температури, руху повітря та відносної вологості, у відповідності до вимог нормативних документів.

Щодо освітлення, то приміщення де експлуатуються ПК, повинно бути обладнаним джерелами штучного освітлення та мати природне освітлення. Нормативний документ, який регламентує вимоги до рівнів природного і штучного освітлення – ДБН В.2.5-28-2018. Природне освітлення забезпечують прозорі вікна та інші світлові прорізи, що знаходяться на півночі або північному сході. У приміщеннях коефіцієнт природного освітлення повинен бути не нижче ніж 1,5 %. Розрахунок коефіцієнта природного освітлення виконують відповідно до методики, яка наведена у ДБН В.2.5-28-2018.

Штучне освітлення у приміщеннях з ПК забезпечується за допомогою системи загального освітлення, переважно рівномірного. В якості штучного джерела світла застосовуються люмінесцентні лампи типу ЛБ.

При використанні ПК для розробки проекту комп’ютерної системи для контролю параметрів мікроклімату теплиць на основі технологій інтернету речей було дотримано наступних вимог з техніки безпеки:

– не виконувався самостійний ремонт ПК і периферійних пристроїв;

– не вносились конструктивні чи інші зміни в апаратне забезпечення комп’ютера;

– використовувались тільки ті матеріали та предмети, які стосувались розробки комп’ютерної системи для контролю параметрів мікроклімату теплиць.

Для забезпечення вимог щодо безпечної експлуатації інформаційних технологій та мереж дотримано вимог СТУ EN 60950-1:2015 «Обладнання інформаційних технологій. Безпека. Частина 1. Загальні вимоги» (ДСТУ EN 60950- 1:2015).

4.2 Організація оповіщення і зв’язку у надзвичайних ситуаціях техногенного та природного характеру

Одним із головних заходів захисту населення від надзвичайних ситуацій (НС) є його своєчасне оповіщення про небезпеку, обстановку, яка склалася внаслідок її реалізації, а також інформування про порядок і правила поведінки в

умовах НС. Під час організації оповіщення і доведення інформації до населення України необхідно керуватися вимогами Положення про організацію оповіщення і зв'язку у надзвичайних ситуаціях, затвердженого постановою Кабінету Міністрів України від 15 лютого 1999 року № 192. Кожний громадянин України повинен знати порядок подавання сигналу “Увага всім!”, діяти за ним та іншими сигналами цивільного захисту (ЦЗ) в умовах НС та особливого періоду.

Встановлено, що система оповіщення та інформування у сфері ЦЗ України включає:

– оперативне доведення до відома населення інформації про виникнення або можливу загрозу виникнення НС, у тому числі через загальнодержавну, територіальні і локальні автоматизовані системи централізованого оповіщення;

– завчасне створення та організаційно-технічне поєднання постійно діючих локальних систем оповіщення та інформування населення із спеціальними системами спостереження і контролю (включаючи державну мережу спостереження і лабораторного контролю) в зонах можливого ураження;

– централізоване використання мереж зв'язку, радіомовлення, телебачення та інших технічних засобів передачі інформації незалежно від форми власності та підпорядкування в разі виникнення НС.

Системи оповіщення населення України мають державний, регіональний, місцевий і об'єктовий рівні. Управління системою оповіщення кожного рівня організовується безпосередньо відповідними органами повсякденного управління системи ЦЗ. Рішення на застосування системи оповіщення приймає відповідний голова державної адміністрації (начальник територіальної підсистеми Єдиної системи цивільного захисту). Відповідальність за організацію і практичне здійснення оповіщення несуть керівники органів виконавчої влади, місцевого самоврядування, підприємств, установ і організацій. Тому керівник об'єкта господарської діяльності і кожний громадянин повинні знати сигнали ЦЗ і уміти правильно за ними діяти.

В результаті наукової розвідки встановлено, що в Єдиній системі ЦЗ України оповіщення населення передбачає спочатку, за будь-якого характеру

небезпеки, включення електричних сирен, переривчастий звук яких означає єдиний сигнал небезпеки "Увага всім!". Для вирішення завдань оповіщення на всіх рівнях Єдиної системи ЦЗ створюються спеціальні системи централізованого оповіщення (СЦО). Системою оповіщення будь-якого рівня є організаційно-технічне об'єднання оперативно чергових служб органів управління ЦЗ, спеціальної апаратури управління і засобів оповіщення, а також каналів (ліній зв'язку), які забезпечують передачу команд управління і мовної інформації у НС.

СЦО регіонального рівня є основною ланкою системи оповіщення в цілому. Саме з цього рівня планується організація централізованого оповіщення. Завданням СЦО регіонального рівня є оповіщення посадових осіб і сил даного рівня, органів управління, сил місцевого і об'єктового рівнів та їх посадових осіб, а також населення, яке проживає на території, на яку поширюється дія СЦО цього рівня. Інформація, яка доводиться до органів управління і посадових осіб, має оперативний характер, а до населення доводиться інформація про характер і масштаби загрози та про дії в умовах НС, які склалися.

Дослідженням встановлено, що основним способом оповіщення населення про НС в умовах мирного та воєнного часу є передача інформації з використанням державних мереж проводового, радіо і телевізійного мовлення. Для зосередження уваги населення перед передачею інформації вмикаються сирени, виробничі гудки та інші сигнальні засоби, що буде означати подання попереджувального сигналу "Увага всім!", після якого негайно приводяться в готовність радіотрансляційні вузли, радіомовні і телевізійні станції, вмикаються мережі зовнішньої звукофікації. За сигналом населення зобов'язане увімкнути радіотрансляційні та телевізійні приймачі для прослуховування нагального повідомлення. У всіх випадках використання систем оповіщення, з увімкненням сирен, негайно доводиться до населення відповідне повідомлення засобами проводового, радіо та телевізійного мовлення. Тексти повідомлень передаються протягом 5 хвилин державною мовою і мовою, якою користується більшість населення в регіоні з припиненням іншої передачі. Тексти звернень записуються на магнітних стрічках на весь обсяг касети з обох сторін. Фонограми і друковані

тексти звернень зберігаються в запечатаних конвертах в оперативних чергових з питань НС, які в необхідних випадках доводяться до населення. Дублікати фонограм і друкованих текстів звернень зберігаються в запечатаних конвертах на радіотрансляційних вузлах, в апаратних радіомовлення, студіях телебачення і використовуються в разі виходу з ладу апаратури оповіщення або аварії на з'єднувальній лінії зв'язку.

У разі повітряної тривоги: "Увага! Говорить Головне управління (управління, відділ) з питань НС облдержадміністрації (міськвиконкому, райдержадміністрації). Громадяни! Повітряна тривога! Відключіть світло, газ, погасіть вогонь у печах. Візьміть засоби індивідуального захисту, документи, запас харчів та води. Попередьте сусідів і допоможіть хворим та людям похилого віку вийти на вулицю. Якнайшвидше дістаньтеся захисної споруди або заховайтеся на місцевості. Дотримуйтеся спокою та порядку. Уважно слухайте повідомлення Головного управління (управління, відділу) з питань НС облдержадміністрації (міськвиконкому, райдержадміністрації)".

Після повітряної тривоги: "Увага! Говорить Головне управління (управління, відділ) з питань НС облдержадміністрації (міськвиконкому, райдержадміністрації). Відбій повітряної тривоги! Усім повернутися до місць роботи або проживання. Допоможіть у цьому хворим та людям похилого віку. Будьте готові до можливого повторного нападу противника. Завжди майте з собою засоби індивідуального захисту. Уважно слухайте повідомлення Головного управління (управління, відділу) з питань НС облдержадміністрації (міськвиконкому, райдержадміністрації)".

У разі загрози хімічного зараження: "Увага! Говорить Головне управління (управління, відділ) з питань НС облдержадміністрації (міськвиконкому, райдержадміністрації). Громадяни! Виникла безпосередня загроза хімічного зараження. Одягніть протигази, сховайте дітей у дитячих захисних камерах. Для захисту поверхні тіла використовуйте захисний одяг, комбінезони та чоботи. При собі майте плівкові (полімерні) накидки, куртки або плащі. Перевірте герметизацію житлових приміщень, стан вікон та дверей. Загерметизуйте продукти харчування і запасіться водою. Укрийте сільськогосподарських тварин

і корми. Допоможіть хворим та людям похилого віку. Сповістіть сусідів про одержану інформацію. Відключіть електронагрівальні прилади. Надалі дійте відповідно до вказівок Головного управління (управління, відділу) з питань НС облдержадміністрації (міськвиконкому, райдержадміністрації)".

У разі загрози радіоактивного зараження: "Увага! Говорить Головне управління (управління, відділ) з питань НС облдержадміністрації (міськвиконкому, райдержадміністрації). Громадяни! Виникла безпосередня загроза радіоактивного зараження. Приведіть у готовність засоби індивідуального захисту та постійно майте їх із собою. Після команди управління (відділу) з питань НС та у справах захисту населення від наслідків Чорнобильської катастрофи надягніть їх. Для захисту поверхні тіла від забруднення радіоактивними речовинами використовуйте захисний одяг, комбінезони та чоботи. При собі майте плівкові (полімерні) накидки, куртки або плащі. Перевірте герметизацію житлових приміщень, стан вікон та дверей. Загерметизуйте продукти харчування і запасіться водою. Укрийте сільськогосподарських тварин і корми. Сповістіть сусідів про одержану інформацію. Допоможіть хворим та людям похилого віку. Надалі дійте відповідно до вказівок Головного управління (управління, відділу) з питань НС облдержадміністрації (міськвиконкому, райдержадміністрації)".

Уразі загрози біологічного зараження: "Увага! Говорить Головне управління (управління, відділ) з питань НС облдержадміністрації (міськвиконкому, райдержадміністрації). Громадяни! Виникла безпосередня загроза біологічного зараження. Для захисту поверхні тіла використовуйте захисний одяг, комбінезони та чоботи. Із собою майте плівкові (полімерні) накидки, куртки або плащі. Перевірте герметизацію житлових приміщень, стан вікон та дверей. Загерметизуйте продукти харчування і запасіться водою. Укрийте сільськогосподарських тварин і корми. Допоможіть хворим та людям похилого віку. Сповістіть сусідів про одержану інформацію. Відключіть електронагрівальні прилади. Надалі дійте відповідно до вказівок Головного управління (управління, відділу) з питань НС облдержадміністрації (міськвиконкому, райдержадміністрації)".

ВИСНОВКИ

Під час виконання кваліфікаційної роботи освітнього рівня «Магістр» було досліджено використання класифікаційних моделей машинного навчання для задач виявлення спаму. Для дослідження використано 2 відкриті набори даних даних TREC 2007 spam Dataset та Enron-Spam dataset.

Було досягнуто виконання наступних завдань:

- проведено аналіз сучасного стану проблеми виявлення спаму та визначено основні виклики, пов’язані з класифікацією текстових даних.
- Проведено огляд літературних джерел стосовно використання методів машинного навчання для задач виявлення спаму;
- проаналізовано доступні набори даних для навчання моделей виявлення спам-листів та вибрано два набори для дослідження;
- проведено попередню обробку тексту, включаючи токенізацію, лематизацію, видалення стоп-слів, а також формування простору ознак у числовому форматі.
- встановлено ефективність різних алгоритмів машинного навчання, таких як наївний Байєса, метод опорних векторів та дерева рішень на обраних набарах даних.
- запропоновано підхід ансамблювання результатів кількох класифікаційних моделей, що дозволило підвищити точність виявлення спаму;
- розглянуто окремі питання з охорони праці і безпеки в надзвичайних ситуаціях.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. The Radicati Group. Email Statistics Report, 2017-2021. [Електронний ресурс]. URL: https://www.radicati.com/wp/wp-content/uploads/2017/01/Email-Statistics-Report-2017-2021-Executive-Summary.pdf.
2. Tymoshchuk, D., Yasniy, O., Mytnyk, M., Zagorodna, N. & Tymoshchuk, V.(2024). Detection and classification of DDoS flooding attacks by machine learning method. CEUR Workshop Proceedings, 3842, 184–195.
3. Hossein Siadati, Sima (Tahereh) Jafarikhah, Markus Jakobsson. Traditional Countermeasures to Unwanted Emails. Understanding Social Engineering Based Scams. 2016. P. 51–62. DOI: http://dx.doi.org/10.1007/978-1-4939-6457-4_5.
4. Lypa, B., Horyn, I., Zagorodna, N., Tymoshchuk, D., Lechachenko T., (2024). Comparison of feature extraction tools for network traffic data. CEUR Workshop Proceedings, 3896, pp. 1-11.
5. Ghozali, Nurul & NOZARI, NUR & Zamzuri, Nur. (2023). Types and Methods of Managing SPAM Messages: A Review. 10.36227/techrxiv.170327076.62289830/v1.
6. 10 Spam Text Message Examples [Електронний ресурс] URL: https://blog.textedly.com/spam-text-message-examples
7. ТИМОЩУК, Д., ЯЦКІВ, В., ТИМОЩУК, В., & ЯЦКІВ, Н. (2024). INTERACTIVE CYBERSECURITY TRAINING SYSTEM BASED ON SIMULATION ENVIRONMENTS. MEASURING AND COMPUTING DEVICES IN TECHNOLOGICAL PROCESSES, (4), 215-220.
8. Ten Spam-Filtering Methods Explained [Електронний ресурс] URL: https://www.techsoupbrasil.org.br/10_sfm_explained
9. N Zagorodna, M Stadnyk, B Lypa, M Gavrylov, R Kozak. 2022. Network Attack Detection Using Machine Learning Methods. Challenges to national defence in contemporary geopolitical situation, No 1, P. 55-61

10. Borotić, Gordana & Granoša, Lara & Kovačević, Jurica & Bagic Babac, Marina. (2024). Effective Spam Detection with Machine Learning. Croatian Regional Development Journal. 4. 43-64. 10.2478/crdj-2023-0007.
11. Ahmed, N., Amin, R., Aldabbas, H., Koundal, D., Alouffi, B., & Shah, T. (2022). Machine Learning Techniques for Spam Detection in Email and IoT Platforms: Analysis and Research Challenges. Security and Communication Networks, 1862888. https://doi.org/10.1155/2022/1862888
12. ТИМОЩУК, Д., & ЯЦКІВ, В. (2024). USING HYPERVISORS TO CREATE A CYBER POLYGON. MEASURING AND COMPUTING DEVICES IN TECHNOLOGICAL PROCESSES, (3), 52-56.
13. Siddique, Z. B., Khan, M. A., Din, I. U., Almogren, A., Mohiuddin, I., & Nazir, S. (2021). Machine Learning-Based Detection of Spam Emails. Scientific Programming, 2021, 6508784. https://doi.org/10.1155/2021/6508784
14. Sinha, A., & Singh, S. (2020). A Detailed study on email spam filtering techniques. International Journal of Data Science and Analytics, 10(3), 1-34
15. Tymoshchuk, D., & Yatskiv, V. (2024). Slowloris ddos detection and prevention in real-time. Collection of scientific papers «ΛΌГОΣ», (August 16, 2024; Oxford, UK), 171-176.
16. Enron-Spam dataset [Електронний ресурс] URL: http://nlp.cs.aueb.gr/software_and_datasets/Enron-Spam/index.html
17. Enron Spam Dataset [Електронний ресурс] URL: https://github.com/mwiechmann/enron_spam_data
18. Emmanuel Gbenga Dada, Joseph Stephen Bassi, Haruna Chiroma, Shafi'i Muhammad Abdulhamid, Adebayo Olusola Adetunmbi, Opeyemi Emmanuel Ajibuwa, Machine learning for email spam filtering: review, approaches and open research problems,Heliyon,Volume 5, Issue 6, 2019, https://doi.org/10.1016/j.heliyon.2019.e01802.
19. Методичний посібник для здобувачів освітнього ступеня «магістр» всіх спеціальностей денної та заочної (дистанційної) форм навчання «БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ» / В.С. Стручок –

Тернопіль: ФОП Паляниця В.А., – 156 с. Отримано з https://elartu.tntu.edu.ua/handle/lib/39196.

Додаток А Публікація

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
імені ІВАНА ПУЛЮЯ
НАУКОВЕ ТОВАРИСТВО ім. ШЕВЧЕНКА

М А Т Е Р І А Л И

XII НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»

18-19 грудня 2024 року

ТЕРНОПІЛЬ
2024

УДК 004.056

Всеволод Дячун, студент гр.СБм-51, Вікторія Ваврічен, студентка гр. СБ-31

Тернопільський національний технічний університет імені Івана Пулюя

МЕТОДИ ВИЯВЛЕННЯ СПАМУ

Vsevolod Diachun, student of gr. СБм-51, Viktoria Vavrichen, student of gr. СБ-31

METHODS OF SPAM DETECTION

Спам — це непотрібна, часто рекламна або шахрайська інформація, що надходить через електронну пошту, соціальні мережі або інші канали комунікації. Спам часто містить шкідливі посилання або вкладення, що можуть привести до зараження комп'ютерів вірусами, шкідливим програмним забезпеченням чи фішингом. Велика кількість спам-повідомлень засмічує поштові скриньки, що може ускладнювати роботу та знижувати продуктивність. Спам несе із собою нав'язливу рекламу, що може займати важливі ресурси пристроїв, наприклад, інтернет-трафік або пам'ять.

Боротьба зі спамом є однією з найстаріших проблем комп'ютерної безпеки, і, водночас, важливою складовою забезпечення безпеки та ефективності цифрового середовища сьогодення. Існує кілька підходів до виявлення та боротьби зі спамом. Традиційні методи виявлення спаму діляться на дві великі групи:

1. Методи, що базуються на аналізі повідомлення. Один з найпоширеніших методів – це фільтрація на основі ключових слів і шаблонів. Програми автоматично аналізують текст повідомлення на наявність певних слів або фраз, характерних для спаму. Крім того, аналіз контенту листа може здійснюватись з використанням сигнатур в оновленій базі даних, із застосуванням статистичних методів на основі теореми Байєса, за допомогою використання SURBL ((Spam URL Real-time Block Lists)
2. Репутаційні схеми. Їхнє завдання - виявляти розсилки однотипних листів серед великої кількості користувачів. Цей метод оцінює репутацію відправника, базуючись на історії його діяльності. Якщо адреса відправника була раніше пов'язана з великим обсягом спаму, система може позначити його повідомлення як спам.
3. Методи, засновані на визнанні відправника спамером. Ці методи спираються на різні чорні списки IP-адрес та адрес електронної пошти. Відомі адреси або IP-адреси, з яких часто надходить спам, можуть бути внесені в чорні списки, що допомагає блокувати подібні повідомлення ще на етапі їх надходження.
4. Методи, засновані на перевірці адреси електронної пошти та доменного імені відправника.

Крім цього фільтрація спам-листів може ґрунтуватись на аналізі поведінки користувачів. Якщо користувач отримує спам-повідомлення від незнайомих адрес чи в незвичний час, система може попередити про ймовірний спам. Це свого роду задача виявлення аномалій.

Сьогодні для фільтрації електронної пошти активно розробляються методи, що навчаються, або інтелектуальні методи, засновані на алгоритмах машинного навчання та техніках глибокого навчання. Підвищення точності роботи цих методів є актуальним науковим завданням.

Література:

1. Saadat Nazirova. Survey on Spam Filtering Technique. Communications and Network, 2011, No 3, P.153-160. doi:10.4236/cn.2011.33019

2. J, Rajesh & G, Mahalakshmi & P, Sudarshan. Email Spam Detection using Machine Learning Techniques. IARJSET, 2020, No 8, P.189-193.

3. AbdulNabi, Isra’a & Yaseen, Qussai. Spam Email Detection Using Deep Learning Techniques. Procedia Computer Science. 2021, No 184(2), P. 853-858

Додаток Б - Лістинг програми, яка зчитає дані датасету Enron Spam Dataset, формує датасет і записує в csv файл

```
import requests
import shutil
import os import re
import datetime as dt
import sys
import pandas as pd
# Downloading and Unpacking data
# Download original email dataset from Athens University of
Economics and Business website
print('Beginning download of email data set from
http://www.aueb.gr/users/ion/data/enron-spam...')
url_base = 'http://www.aueb.gr/users/ion/data/enron-
spam/preprocessed/'
enron_list = ["enron1", "enron2", "enron3", "enron4", "enron5",
"enron6"]
os.mkdir("raw data")
for entry in enron_list:
print("Downloading archive: " + entry + "...")
# Download current enron archive
url = url_base + entry + ".tar.gz"
r = requests.get(url)
path = "raw data/" + entry + ".tar.gz"
with open(path, 'wb') as f:
f.write(r.content)
print('...done! Archive saved to: ' + path)
# Unpack current archive this will unpack to /raw data/enron1, etc
print("Unpacking contents of " + entry + " archive...")
shutil.unpack_archive(path, "raw data/")
print("...done! Archive unpacked to: raw data/" + entry)
print("All email archieves downloaded and unpacked. Now beggining
processing of email text files.")
# Processing data
# The data is recorded in such a way, that each message is in a
seperate file.
# Therefore, we have to open each single file, parse it and add it
to a dataframe
mails_list = []
print("Processing directories...")
# go through all dirs in the list
# each dir contains a ham & spam folder
for directory in enron_list:
print("...processing " + str(directory) + "...")
ham_folder = "raw data/" + directory + "/ham"
spam_folder = "raw data/" + directory + "/spam" i = 0
# Process ham messages in directory
for entry in os.scandir(ham_folder):
```

```
# This should be encoded in Latin_1 but catch encoding errors just
to be sure
try:
file = open(entry, encoding="latin_1")
content = file.read().split("\n", 1)
except (UnicodeDecodeError):
print("COULD NOT DECODE")
print("Problem with file:" + str(entry))
print("Error message:", sys.exc_info()[1])
subject = content[0].replace("Subject: ", "")
message = content[1]
# date is contained in filename - parsed using regex pattern
pattern = r"\d+\.(\d+-\d+-\d+)"
date = re.search(pattern, str(entry)).group(1)
date = dt.datetime.strptime(date, '%Y-%m-%d') file.close()
mails_list.append([subject, message, "ham", date])
# Process spam messages in directory
for entry in os.scandir(spam_folder):
try:
file = open(entry)
file = open(entry, encoding="latin_1")
except (UnicodeDecodeError):
print("COULD NOT DECODE")
print("Problem with file:" + str(entry))
print("Error message:", sys.exc_info()[1])
subject = content[0].replace("Subject: ", "")
message = content[1]
# date is contained in filename - parsed using regex pattern
pattern = r"\d+\.(\d+-\d+-\d+)"
date = re.search(pattern, str(entry)).group(1)
date = dt.datetime.strptime(date, '%Y-%m-%d')
file.close()
mails_list.append([subject, message, "spam", date])
print(str(directory)+" processed!")
print("All directories processed. Writing to Dataframe...")
mails = pd.DataFrame(mails_list, columns=[ "Subject", "Message",
"Spam/Ham", "Date"])
print("...done!")
# Save to file
print("Saving data to file...")
mails.to_csv("enron_spam_data.csv")
print("...done! Data saved to 'enron_spam_data.csv'")

# Confirmation message and data count
print("\nData processed and saved to file.\nMails contained in
data:")
print("\nTotal:\t" + str(mails.shape[0]))
print(mails["Spam/Ham"].value_counts())
```