МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет радіоелектроніки

Факультет _____Комп'ютерних наук_____

Кафедра _____Програмної інженерії_____

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти - другий (магістерський)

Дослідження методів фільтрації спаму на основі вмісту

Виконав:

Студент _2_ курсу, групи _____ІПЗм-19-1_____

_____Шанін А.О._____

Спеціальність 121-Інженерія програмного забезпечення

Тип програми _____освітньо-наукова_____

Керівник _____доц. Вечур О.В._____

Допускається до захисту

Зав. Кафедри, проф. _____ З.В. Дудар

2021 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Програмної інженерії
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 121 – інженерія програмного забезпечення
(код і повна назва спеціальності)

Тип програми Освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Інженерія програмного забезпечення
(повна назва)

ЗАТВЕРДЖУЮ:

Зав.кафедри _____
(підпис)

« ___ »_____ 2021р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студента *Шаніна Андрія Олександровича*
(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження методів фільтрації спаму на основі вмісту*

затверджена наказом університету від *26.03.2021* № *386 Ст*
2. Термін подання роботи до екзаменаційної комісії *20 травня* 2021р.
3. Вихідні дані до роботи *методи фільтрації спаму на основі тексту, підходи нейролінгвістичного програмування та машинного навчання до первинної обробки та векторізації тексту, методи оптимізації моделей, OS Windows, Unix, мова програмування Python*
4. Перелік питань, що потрібно опрацювати в роботі *мета роботи, аналіз проблемної галузі і постановка задачі, аналіз алгоритмів класифікації спаму, аналіз методів векторізації, розробка класифікатору спаму, проектування програмного додатку.*
5. Перелік графічного матеріалу із зазначенням креслеників, схем, слайдів, ілюстрацій: *Актуальність роботи, процес розсилки спаму, приклад кольорового спам повідомлення, структура класифікатору спаму, метод оцінки якості, порівняння результатів алгоритмів класифікації, застосування моделей*

машинного навчання, порівняння результатів методів машинного навчання, висновки.

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Спецчастина	доц. Вечур О.В.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1.	Аналіз предметної галузі	18 лютого 2021р.	виконано
2.	Аналіз існуючих методів та алгоритмів	20 березня 2021р.	виконано
3.	Проектування алгоритму класифікації	03 квітня 2021р.	виконано
4.	Реалізація алгоритмів класифікації та векторізації	12 квітня 2021р.	виконано
5.	Розробка демонстраційного додатку	19 квітня 2021р.	виконано
6.	Підготовка пояснювальної записки	06 травня 2021р.	виконано
7.	Підготовка презентації та доповіді	06 травня 2021р.	виконано
8.	Перевірка на академічний плагіат	14 травня 2021р.	виконано
9.	Нормоконтроль	17 травня 2021р.	виконано
10.	Рецензування	18 травня 2021р.	виконано
11.	Попередній захист	20 травня 2021р.	виконано
12.	Занесення диплома в електронний архів	21 травня 2021р.	виконано
13.	Допуск до захисту у зав. кафедри	21 травня 2021р.	виконано

Дата видачі завдання 25 січня 2021р.

Студент _____
(підпис)

Керівник роботи _____ доц. Вечур О.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Кваліфікаційна робота магістра містить: 67 с., 14 рис., 2 табл., 42 джер.

NATURAL LANGUAGE PROCESSING, PYTHON, ROTATION FOREST, NAÏVE BAYES, PSO, PV-DM, TF-IDF, PSO-BO.

Метою роботи є дослідження, аналіз та реалізація алгоритмів класифікації спаму, методів векторізації тексту та створення кращого класифікатору.

Методи розробки базуються на інструментах та бібліотеках для аналізу даних, використовується мова програмування Python, а також бібліотеки обробки тексту NLTK та Spacy.

В результаті роботи був створен класифікатор спаму, який складається з найефективніших компонентів. Він показує найвищі результати в задачі класифікації спаму.

NATURAL LANGUAGE PROCESSING, PYTHON, ROTATION FOREST, NAÏVE BAYES, PSO, PV-DM, TF-IDF, PSO-BO.

The aim of the work is to study, analyze and implement spam classification algorithms, text vectorization methods and create a better classifier.

Development methods are based on tools and libraries for data analysis, the Python programming language is used, as well as NLTK and Spacy word processing libraries.

As a result, a spam classifier, which consists of the most effective components was created. It shows the highest results of spam classification.

Я, Шанін Андрій Олександрович, студент гр. ІПЗм-19-1, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження методів фільтрації спаму на основі вмісту», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElAr KhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлен з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

ВСТУП

Історія спаму не довга, але дуже насичена подіями. Перше комерційне повідомлення електронною поштою було надіслано Кантером і Зігелем у 1994 році, двома адвокатами штату Арізона, які шукали клієнтів, зацікавлених в отриманні “Зеленої карти” США. Надіслане кільком тисячам новинних груп, повідомлення відразу розлютило стільки людей, що Інтернет-провайдер обох адвокатів за кілька днів припинив зв’язок порушників. Кантер і Зігель вирішили заробляти гроші по-іншому - опублікувавши "Як заробити багатство на інформаційній супермагістралі: Партизанський посібник з маркетингу в Інтернеті та інші онлайн послуги", який незабаром став джерелом натхнення для інших Інтернет для можливості. Джефф Слатон, прізвисько якого пізніше стало «Королем спаму», скористався цією можливістю - спочатку надсилаючи власні спам-повідомлення, а згодом пропонуючи свої послуги іншим. Оскільки послуги Інтернет-провайдера Слатона неодноразово припинялися, він почав розробляти методи для обходу автоматичних методів блокування, розроблені Інтернет-провайдерами. Використовуючи викидаючі рахунки та сервери з відкритою ретрансляцією, він розпочав гонку озброєнь з постачальниками послуг.

Щоб протистояти небажаним повідомленням електронної пошти, постачальники послуг почали сканувати вміст, спочатку шукаючи дайджести, пов'язані з попередніми повідомленнями про спам, а пізніше розглядаючи ключові слова. Коли постачальники послуг почали блокувати велику кількість повідомлень, що містять ключові слова зі спамом, спамери швидко почали протистояти шляхом перетворювання ключових слів. Гарний приклад цього прийшов з фармацевтичного спаму, комерційний успіх якого в 2000-х роках спричинив його швидке збільшення обсягу до понад трьох чвертей всього спаму, коли спамери почали використовувати

“V ! @ g.r A” та подібні мутації замість “Viagra”, щоб обійти фільтрування спаму на основі вмісту. Для спамерів, метою яких був просто продаж продуктів не мало значення, що одержувачі їх електронних листів розуміють, що повідомлення є спамом.

Щоб забезпечити більш надійні механізми боротьби зі спамом, постачальники послуг розпочали використовувати машинне навчання для фільтрації спаму на основі вмісту. Інші підходи, включаючи чорний список IP, аутентифікацію джерела електронної пошти (наприклад, SPF, DKIM, DMARC), виявлення поведінки (наприклад, обсяг повідомлень, які надсилає відправник) та метрики залучення (що одержувачі зазвичай роблять з повідомленнями - наприклад, відкривають повідомлення, розміщують його у папці зі спамом, додавання відправника до адресної книги тощо) також використовуються для виявлення спаму.

Щоб обійти ці підходи виявлення, спамери ввели поліморфний, на основі зображень та Unicode. Щоб уникнути зворотного списку IP та поведінкових підходів, спамери використовують мережі ботів (ботнети), надаючи їм великий пул змінних IP-адресів для надсилання спаму, і використовують стратегію «snowshoe spam» (надсилання електронних листів невеликими партіями та ротацію використаних IP-адресів) для уникнення того, щоб їх не помітили. Більше того, вони використовують такі методи, як викрадення IP (також зване викраденням BGP), щоб підробити IP джерела відправників та відтворення повідомлення, щоб обійти механізми аутентифікації.

На сьогодняшній день небажані електронні листи включають не лише рекламний спам, а й електронні листи з Троянськими вірусними програмами та шахрайством. Троянські електронні листи використовують механізми спаму для розповсюдження шкідливих вкладень або посилань, що спричиняє завантаження шкідливих exe-файлів. Електронні листи з шахрайством ошукують одержувачів за допомогою техніки соціальної інженерії. Під час фішингової атаки особисті та

банківські дані викрадаються веб-сайтом, що претендує на законний бізнес (наприклад, PayPal, Bank of America).

Способи фільтрації спаму класифікуються на кілька категорій. Наприклад, чорні та білі списки, блокування IP-адресу, фільтрація на основі заголовків та фільтрація на основі вмісту. Чорні списки, білі списки та блокування IP - це досить швидкий спосіб виявлення спамерів порівняно з іншими підходами до виявлення. Однак чорні списки та білі списки або блокування IP мають потенційні проблеми, через які спамер може змінити поточний акаунт електронної пошти або один IP на інший, щоб уникнути виявлення. У цьому випадку звичайні методи не можуть легко фільтрувати ці електронні листи зі спамом. Низька продуктивність та низька точність є результатом використання цих підходів. Тобто фільтрація на основі вмісту є фінальним етапом перевірки якості електронного повідомлення.

Отже, метою роботи є дослідження найбільш ефективних методів фільтрації спаму на основі вмісту. Об’єктом дослідження електронні повідомлення, які є спамом і навпаки, електронні повідомлення, які не є спамом (“ham”) та моделі їх класифікації. Методами дослідження є проведення експерементів для визначення найбільш ефективного методу для класифікації електроних повідомлень.

1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Актуальність роботи

Оскільки Інтернет почав набирати популярність на початку 1990-х, його швидко визнали чудовим рекламним інструментом. Практично безкоштовно, людина може скористатися Інтернетом, щоб відправити електронне повідомлення тисячам людей. Справді, електронні листи є ефективним, швидким та дешевим засобом спілкування. Це робить його улюбленим як у професійних, так і в особистих листуваннях. Крім того, періодичне читання електронного листа з невідомого джерела і вміст яких не представляє інтересу для користувача - насправді не біда. Однак, коли більше 60% або навіть 90% електронних листів є такого роду, і часто є незаконними; це те, що можна назвати кошмаром. Цей тип повідомлень називається спамом [1].

SpamCon Inc. [2] підрахував вартість, включаючи втрату продуктивності та ресурсів, спричиненою лише одним небажаним електронним листом у розмірі 1-2 долара. Помножений на кількість спаму, що відправляється та отримується щодня, один долар стає тоді мільйонами. International Data Corp. оцінює кількість спамів щоденно надсилається через мережу на 7,3 мільярда. Ці статистичні дані були достатніми, щоб переконати великих користувачів служби електронної пошти прогнозувати додатковий бюджет для боротьби зі спамом. UUNet, один з найважливіших Інтернет-провайдерів, має групу з шести осіб з бюджетом 1 мільйон доларів, лише для боротьби зі спамами. Netcom підрахував, що 10% від рахунок-фактури для кінцевого користувача призначений для фільтрації спаму. Дослідження Міжнародної корпорації даних поставило спам на друге місце серед проблем Інтернет-провайдерів. Тоді виникає одне питання – чому хтось із задоволенням надсилає стільки електронних листів, і як він отримує стільки адрес? Хоча мотивація

інколи буває різною, спам, як правило, має рекламний зміст. Для трансляції реклама на телеканалі коштує в сотні разів більше, ніж розсилка мільйонів спаму. Отримати стільки адрес електронної пошти зовсім не складно, оскільки багато доступні в самому Інтернеті. Деякі спамери використовують адреси, знайдені в групах новин, загальнодоступних. Деякі інші використовують веб-боти, які зазвичай називають спам-ботами – програмним забезпеченням яке автоматично переглядає Інтернет, шукаючи адреси електронної пошти. Як правило, спам-боти використовують методи зіставлення ключових слів для вилучення адрес електронної пошти. Один із очевидних способів - це пошук символу '@'. Деякі інші використовують програмне забезпечення для генерації випадкових адрес, а потім записують усі адреси, з яких вони не отримують відповіді про збій доставки.

Методи створення спаму з кожним днем стають все винахідливішими. Тому і антиспам алгоритми також потрібно вдосконалювати з кожним днем, як і придумувати нові ідеї по фільтрації спаму. З усього вищесказаного можна сказати, що тема класифікації спаму буде залишатися актуальною поки існує безкоштовний інтернет [3].

1.2 Опис загального методу розсилки спаму та область роботи антиспаму

У 2015 році Symantec оцінив кількість електронної пошти, яка розповсюджується в Інтернеті, приблизно до 28 мільярдів, що відповідає 60% всього електронного трафіку. Для побудови мережі розповсюдження спаму, стійкої до виявлення спаму, спамери значною мірою покладаються на мережі компрометованих машин (чорний список) [4]. Повідомляється, що 74% трафіку спаму розподіляється ботнетами. Ботнет - це мережа комп'ютерів, яка заражена

шкідливим програмним забезпеченням. Кіберзлочинці використовують ботнет-мережі, які складаються з великої кількості комп'ютерів для різних шкідливих дій без відома користувачів. Сам по собі одноранговий ботнет Kelihos відповідав за 52% всього спам-трафіку.

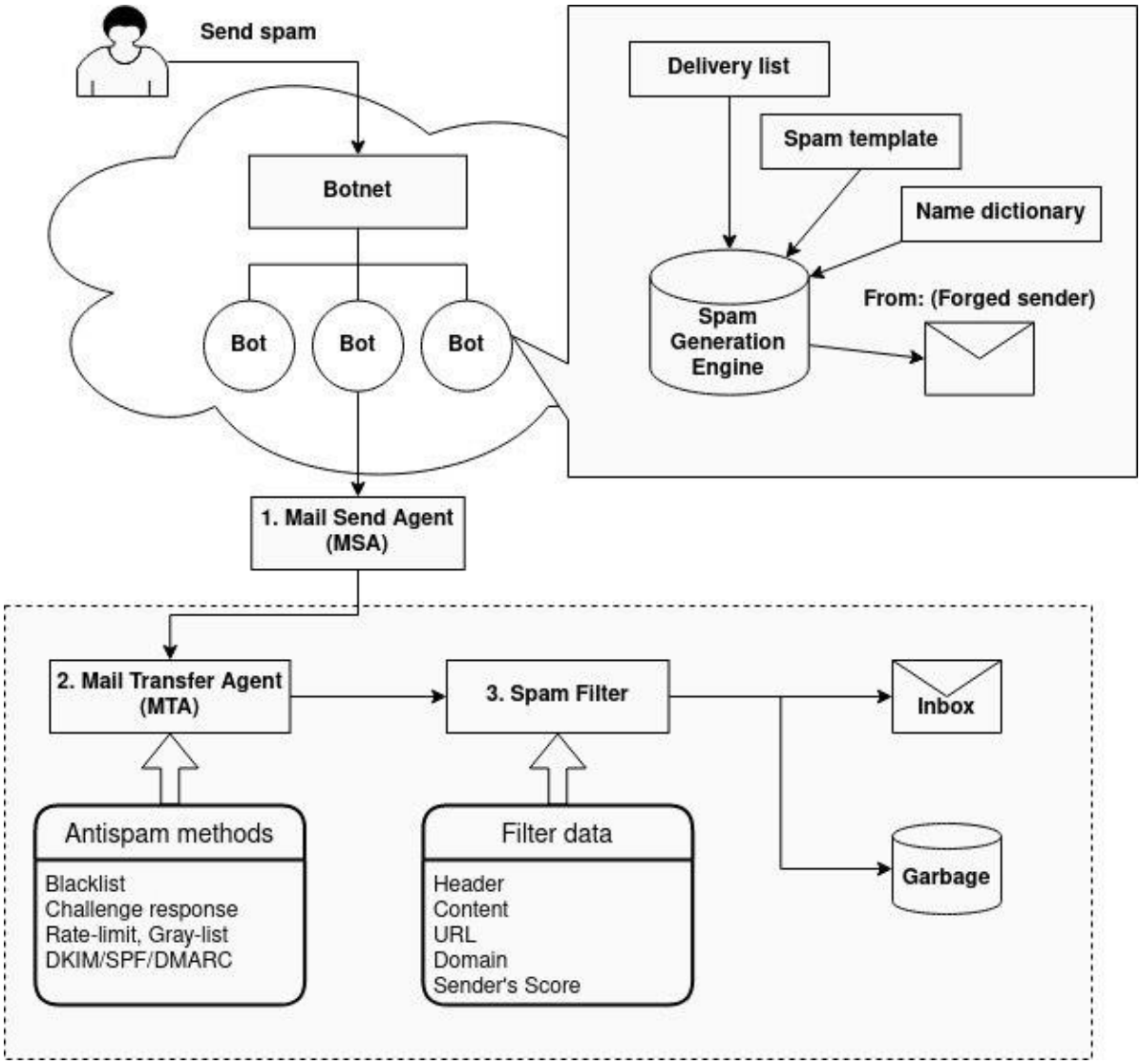

Рисунок 1.1 − Типовий пайплайн спаму

На рисунку 1.1 зображено спрощений потік спам-повідомлення від джерела до пункту призначення. У цій подорожі скомпрометована машина використовує список

доставки, шаблон спаму та словник імен, щоб створити спам-повідомлення та надіслати його агенту подання пошти (MSA). MSA знаходить IP-адресу агента передачі пошти (MTA), шукаючи Сервер доменних імен (DNS) одержувача. За допомогою протоколу SMTP (Simple Mail Transfer Protocol) MSA надсилає електронне повідомлення на MTA. Спам може спрямовуватися через кілька інших MTA, щоб дістатися до поштової скриньки користувача.

При переході від джерела до пункту призначення існує кілька місць, в яких спам можна зупинити. Перше місце - це MSA, який може відмовитись передавати спам-повідомлення. Це робиться шляхом обмеження кількості повідомлень, прийнятих від відправника, включення в чорний список IP-адрес спам-ботів або виявлення скомпрометованих облікових записів електронної пошти. Друге місце - це MTA, який здатний застосувати ряд методів (наприклад, створення чорного списку, SPF, DKIM та DMARC) для перевірки відправників електронних листів.

Крім того, MTA може обмежити обсяг спаму, використовуючи сірий список та механізми реагування на виклики. Нарешті, спам-фільтри можуть класифікувати спам на основі комбінації функцій, включаючи текст, посилання, структуру та аналіз одержувачів електронної пошти [5].

Ці підходи разом можуть виявляти великі та середні спам-кампанії, де є сильні сигнали, такі як повторний вміст, URL-адреси та IP-адреси відправників. Однак деякі типи шахрайських повідомлень залишаються невиявленими з кількох причин. По-перше, ці шахрайські атаки спрямовані на невелику кількість жертв і надсилаються з використанням IP-адрес, яких немає в чорному списку. IP-адреси залишаються не виявленими через низький обсяг повідомлень. По-друге, використовується змінений вміст, який може обходити існуючі механізми фільтрації спаму на основі вмісту через невеликий обсяг таких електронних листів. Нарешті, деякі типи шахрайства використовують вкрадені дані або псевдоніми, надані

постачальниками послуг, для спілкування з користувачами за автентифікованими каналами.

1.3 Опис підходів щодо фільтрації спаму

1.3.1 Метод чорних списків

Велика частина спаму надсилається спам-ботами через відкриті ретранслятори (MSA, які надсилають електронні листи без аутентифікації) або автономні агенти електронної пошти. Список IP-адрес, які, як відомо, надсилають спам, є чорним списком. Деякі компоненти інфраструктури електронної пошти, включаючи MSA, MTA або програмне забезпечення для фільтрації спаму на основі вмісту, проконсультуються з чорними списками для підбору електронних листів. Наприклад, SpamAssassin використовує тридцять п’ять чорних списків та використовує результати підрахунку електронних листів [6]. Великі провайдери використовують найвідоміші чорні списки, такі як SpamCop, Spamhaus та URIBL.

1.3.2 Методи антиспуфінгу

Спуфінг полягає не в тому, щоб змогти надіслати повідомлення, а навпаки, у тому, щоб повідомлення виглядали звичайними. Звичайно, це менш актуально в контексті спаму, ніж у контексті шахрайства.

Простий протокол пересилання пошти (SMTP) не автентифікує відправника електронних листів. Отже, спамер може надіслати електронний лист із удаваною або існуючою електронною адресою. Це називається спуфінгом і в основному

використовується спамерами, щоб маскуватись як авторитетну особу. Для подолання цієї проблеми пропонуються підходи проти спаму для автентифікації відправників.

1.3.2.1 DKIM

DomainKey Identified Mail (DKIM) [7] - це криптографічний метод перевірки цілісності електронної пошти та аутентифікації за допомогою інфраструктури відкритого ключа та цифрового підпису. За допомогою цього методу повідомлення електронної пошти підписуються приватним ключем домену-відправника. Домен одержувача перевіряє підпис електронного листа за допомогою відкритого ключа домену відправника, доступного через DNS.

1.3.2.2 SPF

Використовуючи SMTP, зловмисники можуть легко підробити адресу відправника електронної пошти та зробити так, щоб вона була надіслана із законного домену. Щоб обмежити цю можливість, Sender PolicyFramework (SPF) забезпечив простий механізм перевірки вихідних IP-адресів електронних листів. При цьому підході домен відправника надає список авторизованих IP-адресів з домену, яким дозволено розсилати електронні листи. Цей список поширюється інфраструктурою DNS через записи DNS відправника. Отримавши електронне повідомлення, одержувач запитує DNS відправника основного, щоб знайти список авторизованих IP-адрес. Електронні листи, надіслані з неавторизованих IP-адрес, розглядаються як

спам. Як приклад, запис DNS “example.com. TXT ‘v = spf1 a: mail.example.com –all’” означає, що домен example.com уповноважує лише комп’ютер з іменем домену “mail.example.com” надсилати свої електронні листи. Коли MTA отримує електронне повідомлення від “example.com”, він переглядає записи SPF для DNS відправника та приймає повідомлення електронної пошти, лише якщо IP-адреса відправника відповідає “mail.example.com”. Якщо в DNS відправника немає запису SPF, це потенційно може означати, що домени відправника не підтримують SPF.

1.3.2.3 DMARK

Domain-based Message Authentication, Reporting, and Conformance (DMARC) [8] об'єднує разом існуючі механізми, включаючи DKIM та SPF. Це дозволяє відправнику електронних листів публікувати підтримуваний тип автентифікації електронної пошти (DKIM / SPF), а також пропонувати політики та налаштування для перевірки повідомлень, розподілу та повідомлення про невдалі автентифікації. Більше того, він забезпечує механізм звітування про дії, що виконуються відповідно до цієї політики.

У 2015 році лише 26% вхідного трафіку Gmail надходило з доменів з опублікованою політикою DMARC. Ще більш вражаючим є те, що лише 1,1% із мільйона веб-сайтів Alexa публікують політику DMARC. Хоча 81% доменів підтримують як DKIM, так і SPF, 11% підтримують лише SPF, а близько 2% підтримують лише DKIM. Ця невідповідність обмежує ефективність DKIM та SPF, оскільки одержувачі не можуть визначити, чи є недолік підпису призначеним чи спричиненим підміною.

1.3.3 Класифікація спаму на основі вмісту

1.3.3.1 Класифікація спаму на основі тексту

У своїй простій формі фільтрація спаму може бути перероблена як завдання категоризації, де передбачувані класи це “spam” та “ham”. Різні контрольовані алгоритми машинного навчання були успішно застосовані до завдання фільтрації пошти. Основна ідея таких методів - класифікувати електронний лист на небажаний чи законний, перевіривши деякі параметри тексту повідомлення. Такими параметрами можуть будет, на приклад присутність слова “free”, тобто таке повідомлення повинно бути відмічено як “spam”. Або навпаки, повідомлення містить підтвердження замовлення “order confirmation” та таке повідомлення треба відмітити як “ham”. Але швидше за все такі класифікатори будуть дуже слабкими. Тому люди почали користуватися методами машинного навчання такими як нейронні мережі, вероятносні моделі, дерева ухвалення рішень, регрессийні моделі, їх комбінації та інші.

1.3.3.2 Класифікація спаму на основі зображень

Класифікація текстових електронних листов видаються досі ефективною. Однак спамери не перестають винаходити фокуси, щоб обійти фільтри. Наприклад, включення в тіло повідомлення картинку, або кольоровий текст як на малюнку 1.2.

Рисунок 1.2 – Приклад кольорового спам-повідомлення

В даному прикладі можна помітити що не тільки текст несе в собі смислове навантаження, але й кольори [9]. Тому, крім класифікації самого тексту, присутна можливість додати набір інших ознак, які можна отримати з повідомлення. Коли нам потрібно обробити зображення, нам необхідно розпізнати всі символи картинки і зібрати їх в слова. Це робиться за допомогою Optical Character Recognition (OCR) [10]. Потім кожній букві/слову/реченню присвоюється відповідна характеристика кольору і вже ці дані слід класифікувати для визначення того, чи є повідомлення спамом [11].

1.4 Структура процесу створення класифікатора спаму на основі тексту

На малюнку 1.3 ми можемо побачити візуалізацію процесу створення класифікатора спаму на основі тексту [12].

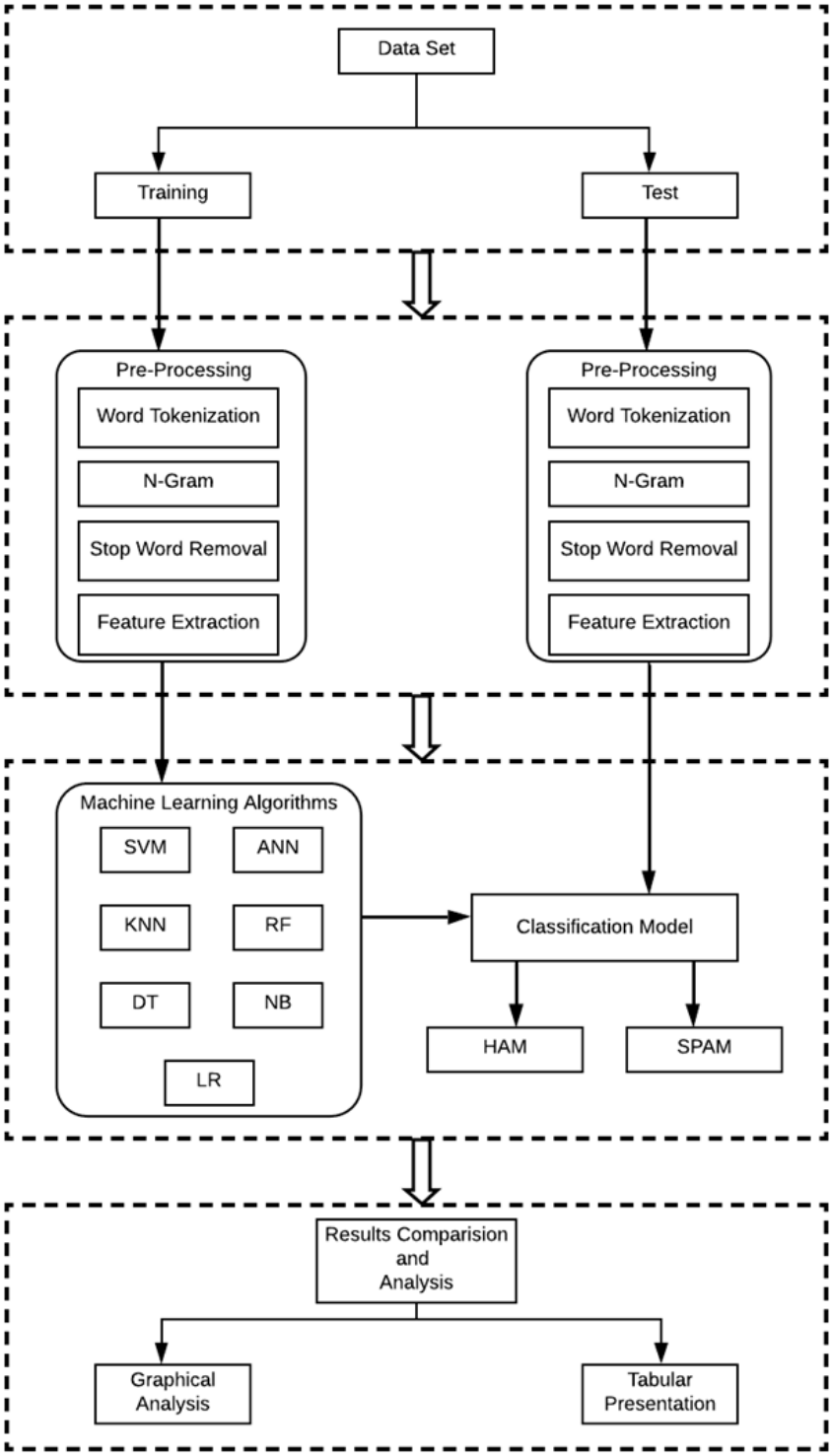

Рисунок 1.3 – Процес створення антиспам класифікатору

1.5 Постановка задачі

Так як в кінцевому рахунку основною інформацією в поштовому повідомленні є текст, то було прийнято рішення порівняти різні підходи для класифікації тексту. На основі результатів аналізу предметної області метою роботи стає дослідження моделей бінарної класифікації, для фільтрації спаму в поштових повідомленнях, методів векторізації тексту та методів первинної обробки тексту. Було вирішено більш детальніше розглянути варіанти первинної обробки тексту, методи векторізації тексту та можливі моделі, якими можна робити бінарну класифікацію. Вибрати для кожного компоненту класифікатору серед розглянутих моделей найкращу та модифікувати її для створення якісного класифікатору.

Було визначено наступні задачі:

- розглянути та вибрати кращі з варіантів препроцессінгу тексту, визначивши їх переваги та недоліки, для класифікатору спаму;
- розглянути та вибрати кращі з варіантів векторізації тексту, визначивши їх переваги та недоліки, для класифікатору спаму;
- розглянути алгоритми класифікації, які використовуються для класифікації спаму, проаналізувати їх, вибрати найкращі;
- модифікувати обрані алгоритми;
- створити класифікатор, натренувати моделі та підібрати гіперпараметри;
- провести набір експериментів, які відображають результати модифікації алгоритму та вибір обраних компонентів класифікатору.

Таким чином буде визначено найкращий спосіб класифікації спаму і подальші кроки для поліпшення даної моделі.

2. ОПИС ПРОВЕДЕНИХ ТЕОРЕТИЧНИХ ДОСЛІДЖЕНЬ

В цьому розділі ми виберемо та обгрунтуємо вибір кожного компоненту класифікатору серед розглянутих моделей, визначемо найкращій алгоритм класифікації та модифікуємо його з обгрунтуванням модифікації для створення якісного класифікатору.

2.1 Вибір методів первісної обробки тексту

Для того що б працювати з текстом, нам потрібно його спочатку привести в належний вигляд, так як первісно він може бути абсолютно різним і навіть не читабельним. Виходячи з [12], [13], будемо використовувати наступні методи первинної обробки тексту:

- видалення чисел і цифр, або заміна їх на якесь слово;
- видалення знаків пунктуації;
- видалення пробільних символів, поділ тексту на набір слів;
- видалення стоп-слів, такі як артиклі, прийменникі, вигуки та ін.;
- стемінг – перетворення слова до початкового виду (кореню), прибираючи при цьому префікс, суфікс і закінчення;
- лематизація – зміна слова в канонічну форму, тобто, інфінітив;
- токенізація – розбиття тексту на менші частини, такі як слово, словосполучення, речення або абзац;
- вилучення 2-граммів – додавання до набору токенів набори з 2 послідовних токенів щоб зберегти більше сенсу з початкового тексту.

2.2 Вибір методу векторізації

Існує кілька часто використовуваних методів векторизації тексту. У текстах різного характеру використовують різні методи та їх комбінації. Далі наведені основні методи векторизації тексту [14]:

- word embedding [15], основною ідеєю якого є дослідження місцевого контексту. Тобто це вміння вгадувати наступне слово, операючись на попередні;
- skip gram model [16], метою якої є знаходження такої репрезентації слів, щоб було зручно знаходити опорні слова у тексті, за якими зазвичай здійснюється пошук документів;
- continuous bag-of-words (CBOW), яка намагається передбачити слово на основі контексту, але порядок слів у контексті немає значення;
- PV-DM використовує вектор абзацу разом із векторами слів, щоб сприяти виконанню завдання передбачення наступного слова;
- DBOW ігнорує контекстні слова у вхідних даних, але змушує модель передбачати слова, випадково вибрані з абзацу у вихідних даних;
- feature selection перетворюють існуючі об’єкти в новий простір об’єктів нижчої розмірності, тобто зменшуються об’єми ознак. Під час цього процесу створюються нові функції на основі лінійних або нелінійних комбінацій ознак з вихідного набору;
- TF-IDF показує наскільки важливе те або інше слово у контексті набора документів.

На основі дослідження [14], було вирішено використовувати методи PV-DM та TF-IDF, так як їх комбінація показувала кращій результат. Ці методи взаємодоповнюють один одного – PV-DM була навчена генерувати вектор для

кожного слова та кожного електронного листа та показує лише семантичне значення слів, у той час як метод TF-IDF вловлює ознаки які показують високу важливість. Тобто сукупність цих методів буде аналізувати і семантику слів, і їх важливість у тексті.

2.3 Вибір основного методу классификації

Існує багато моделей класифікації як багатокласових, так і бінарних, як у нашому випадку. Для того щоб визначити який з них краще підходить до задачі класифікації спаму, були розглянуті ряд останніх публікацій [17] – [26], в яких порівнювалися ефективності деяких алгоритмів класифікації електронних листів та їх модифікацій. Різні алгоритми розроблялися або порівнювалися різними авторами та на різних наборах даних. У роботі [17] були розібрані найбільш об’ємно попередні дослідження інших авторів, алгоритми класифікації які вони використовували для порівняння або модифікації, та деякі нові моделі. Були розглянуті наступні стандартні алгоритми: Naïve Bayes, Logistic Regression, K-neighbors [18], ANN [18], SVC, Random Forest, Random Tree, J48, багатошаровий перцептрон, SVM [22] та інші менш відомі, або їх модифікації: C-PLS, C-RT, CS-CRT, CS-MC4, CS-SVC, SCS-SCM [19], Continouns PLS-DA, PLS-LDA, LDA[3], Bayesnet, Rotation Forest [17], Bayesian Logistic Regression, Hidden Naïve Bayes, Voted Perceptron, REP Tree [17], [23], [24] – [26].

В роботі [22] були поєднаны штучна нейронна мережа, оптимізація рою частинок (PSO) та метод опорних векторів для класифікації та відокремлення спаму. Їх метод порівнювали з іншими методами, такими як класифікація даних Self

Organizing Map та K-середніх з методом оцінки площа під кривою (AUC). Результати показали що цей метод кращій за інші.

У роботі [23] провели експеримент з багатьох методів класифікації спаму, намагаючись знайти найбільш підходящий класифікатор для класифікації електронної пошти як спаму та не-спаму. вони перевірили ефективність багатьох класифікаторів і виявили, що в частині аналізу результатів наївський класифікатор Байєса (NB) забезпечує точність 76%, що показує результат, який кращє за інших двох класифікаторів, таких як SVM та J48. А використання часу для класифікації та тренування для класифікатора NB менше, ніж для інших двох, що означає, що NB класифікатор є найкращим класифікатором серед інших двох для класифікації спаму.

У роботі, яка показала найбільш ємне дослідження [17], порівнявши більш перспективні алгоритми, які були виявлені раніше, такі як Bayessian Logistic Regression, Hidden Naïve Bayes, RBF Network, Voted Perceptron, Lazy Bayessian Rules, Logit Boost, Rotation Forest, NNge, Logistic Model Tree, REP Tree, Multilayes perceptron, Naïve Bayes, J48, Random Tree. Порівняння цих алгоритмів показало, що найбільш ефективно разпізнає спам алгоритм Rotation Forest [27].

Так як Rotation Forest [27] показав найбільш ефективній результат в рамках цієї задачі, то метою даного дослідження було визначено поліпшення цього алгоритму. Для цього ми можемо спробувати використовувати Rotation Forest в якості бази для іншого ансамблієвого підходу або спробувати інші слабкі класифікатори в якості бази для класифікатору Rotation Forest замість стандартного Decision Tree.

2.4 Концепція модифікованого алгорітму Rotation Forest

У цьому розділі ми обговоримо основні концепції алгоритму класифікації, створеного на базі Rotation Forest.

Для його поліпшення, розглянемо роботу з описом корнцепціі алгоритму [27]. В цієх роботі розповідається про напрямки поліпшення роботи алгоритму, серед яких присутні концепція використання Rotation Forest в якості бази для інших ансамбліевой моделі, та концепція використання іншого алгоритму в якості базового для Rotation Forest, наприклад, алгоритму наївного Байєсу.

Ансамблієві методи зазвичай використовують в якості базових алгоритмів слабкі моделі навчання, тобто ті, які не сильно відрізняються від звичайного вгадування. А модель Rotation Forest є крім того що ансамбліевой метамоделлю, за своєю суттю, так ще і дуже сильною моделлю. Тобто, скоріше за все, використання цього алгорітму в якості бази для іншого ансамблієвого алгорітму дасть незначний приріст в якості алгоритму в рамках класифікації спаму.

Розглядаючи концепцію використання іншого алгоритму в якості базового для Rotation Forest, можна побачити що це може бути гарною ідеєю. Наприклад, безліч досліджень [3], [12], [17] показало, що алгоритм наївного Байєса є найбільш ефективним простим класифікатором для використання в задачі класифікації спаму, значно ефективнішим за Decision Tree, яке є стандартною базою для Rotation Forest.

Розглянувши різні роботи з використанням наївного Байесовского класифікатору в якості одного з базових алгоритмів для Rotation Forest [28], [29], ми можемо зробити висновок, що ця модифікація швидше за все буде дуже ефективною.

В роботі [30] ми можемо побачити, що метод роя частинок (PSO) [31] дуже сильно підвищує ефективність пошуку локального мінімуму Байєсовським класифікатором в розпізнаванні спаму.

Виходячи з даних висновків, в якості основного алгоритму класифікації в даному дослідженні буде використано Rotation Forest, з базовим алгоритмом Multinomial Naïve Bayes, який буде навчатися за допомогою методу PSO. Але, щоб порівняти яка модифікація як впливає на базовий алгоритм, будемо використовувати різні комбінації цих модифікацій.

Оптимізація рою частинок [31] – це обчислювальний метод, який оптимізує проблему шляхом ітеративних спроб поліпшити рішення кандидата з урахуванням заданого показника якості. Він вирішує проблему, маючи сукупність рішень, і переміщуючи ці частинки в просторі пошуку відповідно до простої математичної формули над положенням і швидкістю частинки. На рух кожної частинки впливає її найвідоміше місцеве положення, але воно також спрямовується до найвідоміших позицій у просторі пошуку, які оновлюються, оскільки кращі позиції знаходять інші частинки. Очікується, що це рухатиме рій до найкращих рішень.

У PSO, відповідно до взаємодії між різними частинками, швидкість кожної частинки оновлюється. Положення векторних частинок і швидкість частинок - дві основні динаміки алгоритму PSO. Кожна частинка ділиться своїм досвідом з іншими частинками, а також змінює свою траєкторію руху відповідно до досвіду інших частинок, щоб досягти кращого рішення. PSO в основному використовується з метою глобальної оптимізації рішення. У цьому дослідженні PSO використовується для оптимізації параметрів алгоритму Naive Bayes для класифікації спаму електронною поштою.

2.5 Оптимізований метод пошуку гіперпараметрів PSO-BO

Методи оптимізації рою частинок були успішно використані при оцінці параметрів моделі. Однак, коли мова заходить про проблему оцінки гіперпараметрів, методи рою частинок, як і багато інших методів оптимізації, не можуть бути безпосередньо використані для вирішення цієї проблеми. Складність полягає в тому, що відображення від гіперпараметрів моделі до функції втрат або помилки узагальнення полягає у відсутності явних математичних виразів, а складність обчислень дуже велика.

Тому наївні методи, такі як пошук у сітці та випадковий пошук, завжди використовуються у галузі оцінки гіперпараметрів у традиційних інженерних практиках. Ці методи проводять велику кількість незалежних експериментів під різними припущеннями про гіперпараметри, а потім вибирають найкращі гіперпараметри.

У роботі [32] було створено підхід PSO-BO. У рамках PSO-BO метод PSO використовується для оптимізації функції придбання щоб отримати нові локальні мінімуми, що значно зменшує обчислювальне навантаження. Емпірична оцінка моделі показала, що PSO-BO покращує стандартні методи пошуку гиперпараметрів. Отриманий метод можна використовувати з більшістю функцій отримання. Однак алгоритм працює повільно у багатовимірному просторі. Тобто рекомендується його використовувати з невеликою кількістю різних гіперпараметрів, але серед одного гіперпараметру можна перевіряти досить багато значень. У подальшій роботі будемо використовувати метод PSO-BO для пошуку гіперпараметрів.

2.6 Метрики оцінки алгоритмів

Ефективність запропонованого алгоритму оцінюється з точки зору точності (accuracy), f-міри (f-measure), влучності (precision) та повноти (recall). Ці параметри обчислюються за допомогою Істинно Позитивний (TP), Хибно Позитивний (FP), Істинно Негативний (TN), Хибно Негативний (FN). Ці заходи визначені нижче:

- TP – кількість електронних листів зі спамом правильно визначено як спам;
- TN – кількість електронних листів, які не є спамом, правильно визначено як не спам;
- FP – кількість звичайних електронних листів неправильно визначено як спам;
- FN – кількість спам-листів неправильно визначено як звичайні повідомлення.

Експерименти, які будуть проведені будуть порівняні за наступними метриками:

- Recall: можна визначити як ймовірність правильної класифікації спаму. Вище відкликання вказує на те, що фільтр має тенденцію не створювати FN, але він може створювати FP, яка визначається за формулою 1:

$$Recall = \frac{TP}{TP + FN} \tag{1}$$

- Precision: вимірює точність методу фільтрації для правильної класифікації спаму, який визначається за формулою 2:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

– Accuracy: це здатність методу фільтрації правильно класифікувати законні електронні листи та електронну пошту зі спамом, яка визначається за формулою 3:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{3}$$

– F-міра: популярний показник, що поєднує в собі влучність та повноту, обчислюючи їх середнє гармонічне значення. Цей показник відображає значність класифікації спаму лише тоді, коли насправді повідомлення є спамом, ніж фільтрування всього спаму, яка визначається за формулою 4:

$$\text{F}_1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

3. ОПИС ПРОЦЕСУ ТА РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

3.1 Вибір середовища та програмного забезпечення

Для дослідження було прийнято рішення реалізовувати методи мовою Python (версії 3.7.3) з використанням опенсорсних бібліотек та фреймворків. Це рішення було прийняте тому що більшість методів машинного навчання реалізовано на цій мові, а середовище (фреймворки та бібліотеки) дуже зручно використовувати. Всі алгоритми були взяті або реалізовані за допомогою nltk[33], sklearn[34], tensorflow[35], keras[36]. Графіки та діаграми були построєні за допомогою бібліотеки matplotlib [37].

3.2 Процес дослідження

Для подальших експериментів необхідно було визначити базу поштових повідомлень, на основі яких будуть визначені результати досліджень. Було знайдено декілька датасетів, інформація про які наведено у таблиці 1.

Таблиця 1 – Публічні поштові спам датасети

Назва	Кількість “spam”	Кількість “ham”
Spam archive [38]	15090	0
Spamassasin [39]	1897	4150
Enron-spam [40]	16545	17171
Trec 2007 [41]	50199	25220
Ling-spam [42]	481	2412

На основі робот [12], [14], було вирішено взяти набір датасетів, які буде скомпоновано в один, для того щоб класифікатор працював більш стабільно на різих даних. Для проведення дослідження були об’єднані наступні датасети: Enron-spam, SpamAssasin, Ling-spam. Спочатку був вибран датасет “Trec 2007”, але потім його було відкинуто, бо не вистачило обчислювальної потужності, для його тренування та тестування треба виділити великий кластер. Датасет Spam archive не був обран з того що він зберігає тільки спам, що не відповідає нормам датасетів в рамках нашого дослідження.

Скомпонований датасет в 18923 спам повідомлень та 25533 звичайних був розділений на 8891 повідомлень для тестування алгоритму та 35565 повідомлень для тренування. На базі цього датасета треба провести експерименти класифікації повідомлень, використовуючи різні методи машинного навчання.

Процес дослідження було вирішено скласти наступним чином:

- завантажити та проаналізувати датасет;
- зробити первинну обробку даних та декілька варіантів векторізації;
- проаналізувати алгоритми за допомогою методу кросс-валідації;
- проаналізувати алгоритми за допомогою метрик, описаних раніше;
- проаналізувати алгоритм за допомогою ROC-кривої;
- провести фінальне коригування алгоритму та отримати результати.

3.2.1 Аналіз датасету

Перш за все спочатку треба проаналізувати датасет, визначити кількість спам повідомлень, та їх характеристики. На малюнку 3.1 зображене розподілення звичайних та спам повідомлень в вхідному датасеті.

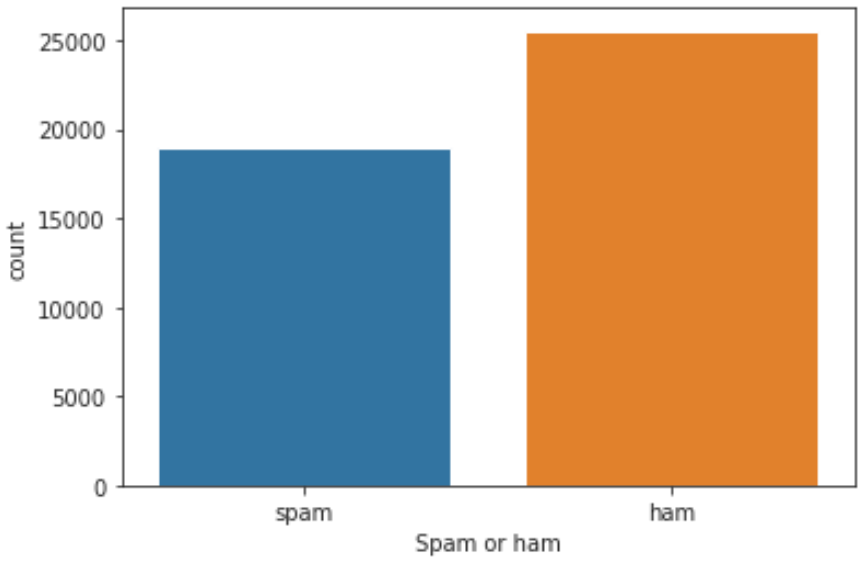

Рисунок 3.1 – Візуалізація розподілення поштових повідомлень в датасеті

На данному малюнку ми бачимо що кількість звичайних повідомлень перевищує кількість спаму. Це показник того, що клас не зовсім збалансований. Але перед тим, як його балансувати, потрібно тричі подумати, чи потрібно це робити. Коли ми балансуємо класи, ми або щбільшуємо, або зменщуємо деякі класи. Але в той же час, розподілення букв, слов та інших ознак може змінитися, та й дуже сильно. Тому в нашому дослідженні ми не будемо нічого робити з датасетом. Але це не є постулатом – робота з датасетом також є предметом для дослідження.

Далі розглянемо розподілення даних по кількості букв, слов та речень в повідомленні, яке ми зможемо побачити на рисунку 3.2.

	count	mean	std	min	25%	50%	75%	max
num_characters	33716.0	1661.038676	4592.037257	2.0	362.0	779.0	1746.25	242650.0
num_words	33716.0	364.407818	959.567539	2.0	83.0	178.0	393.00	50536.0
num_sentence	33716.0	13.512368	45.013157	1.0	3.0	6.0	13.00	2991.0

Рисунок 3.2 – Токенізірована інформація о датасеті

Можемо побачити, що повідомлення бувають дуже різні за кількістю інформації. Бувають повідомлення як двосимвольні, так і навіть з трьох тисяч речень. На малюнку 3.3 ми можемо побачити розподілення цих повідомлень за кількістю символів, слов та речень на графіках

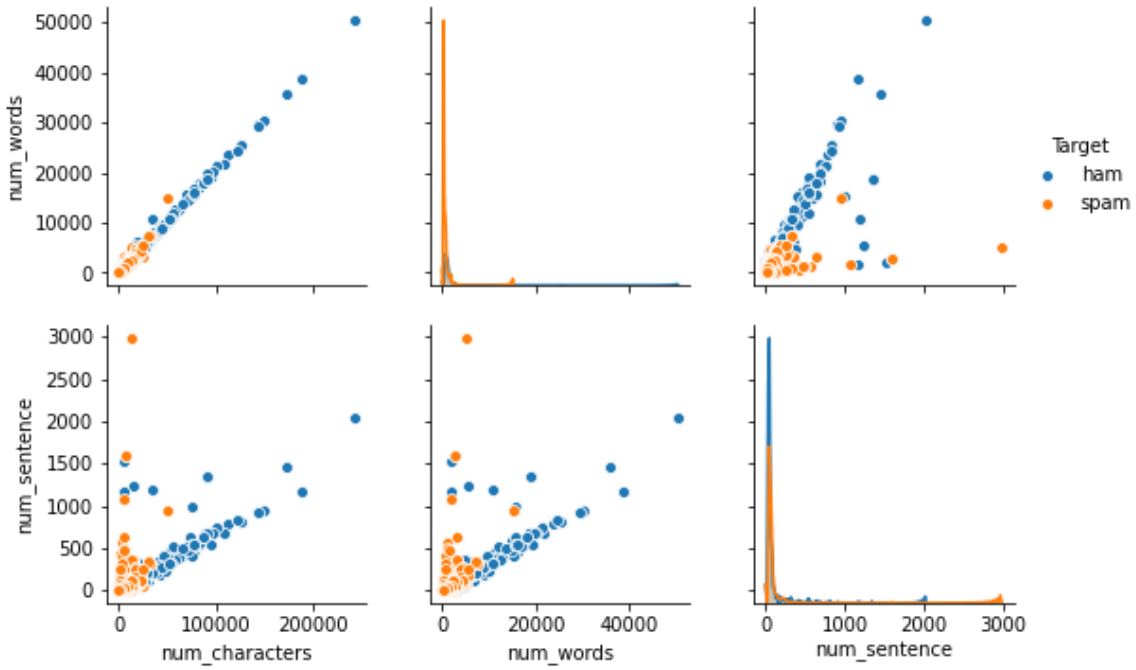

Рисунок 3.3 – Розподілення повідомлень за кількістю символів, слов та речень

На цьому малюнку ми бачимо що чим більше у повідомленні символів та слов, тим більше шанс що це повідомлення не є спамом. Але ця гіпотеза не завжди стосується кількості речень, бо кількість слов прямо пропорційна кількості речень лише у повідомленнях, де кількість речень не перевищує 1000, в інших випадках кількість слов може бути довільной. Повідомлення де кількість рнечень перевищує за 1000 в даному випадку можна назвати вибросами. Але, по перше треба визначити, чи можуть алгоритми класифікації добре розпизнавати ці повідомлення.

3.2.2 Первинна обробка даних

Спочатку треба текстові документи перевести в набор токенів (слов). Але перед цим треба видалити з тексту зайву інформацію: посилання, числа, цифри, спеціальні знаки, знаки пунктуації. Також треба видалити стоп-слова – слова які найчастіше зустрічаються в тексті такі як “can” “so” “an” в англійскої мові. Потім треба перетворити слова, які залишилися в єдину форму за допомогою стемінгу та лемматизації – це перетворення слова у інфінітив, та зміна даяких його схожих букв-цифр, наприклад ‘1’ перетворюється у ‘i’. Після цього треба токенізувати повідомлення. Це можна зробити двома способами, за допомогою модулей sklearn та keras – ми скористуємося обидвома способами та виберемо кращій. При процесі токенізації є багато варіантів перетворити текст у набір токенів. Токеном може бути одне або декілька слів, які називаються n-грамами. Як було сказано раніше, в якості ознак документа ми будемо використовувати токени з одного слова та 2-грами. Тільки після цього процесу можна починати перетворювати текст в вектора та матриці, тобто виконувати процес векторізації.

3.2.3 Детальний опис процесу векторізації

Для векторізації будемо використовувати комбонований підхід PV-DM та TF-IDF. Мета векторізації цим методом полягає в тому, що перше представлення описує глобальний контекст електронної пошти, тоді як друге представлення описує локальний контекст відповідних функцій кожного електронного листа. Роблячи це, ми намагаємось забезпечити кращу репрезентацію, яка фіксує семантичний аспект

слів, комбінуючи вбудовану інформацію, витягнуту як із локального, так і з глобального контексту кожного електронного листа.

В даному підході обчислюються два векторних подання кожного електронного листа за допомогою моделі глибокого навчання PV-DM та методу TF-IDF. Модель PV-DM була навчена генерувати вектор для кожного слова та кожного електронного листа в навчальному корпусі. Сформовані вектори були згруповані в дві матриці:

– матриця D, де кожен стовпець представляє векторне представлення електронної пошти;
– матриця W, де кожен стовпець є векторним поданням слова.

Перший вектор даного електронного листа отримується шляхом вилучення відповідного вектора (стовпця) з матриці D. Витягнутий вектор ми називаємо V_{PVDM}.

Щоб отримати другий векторн даного електронного повідомлення, метод TF-IDF був застосований до навчального корпусу. Значення $tf - idf$ збільшується із збільшенням кількості випадків входження слова в електронну пошту, але зменшується із збільшенням кількості випадків, коли слово входить у весь корпус електронних листів.

Тому метод TF-IDF вловлює лише відповідні особливості з високим значенням. Однак на наступному кроці ми витягнемо векторне подання вибраних ознак з матриці W, щоб обчислити їх середнє значення. Отриманий вектор використовується як другий вектор представлення електронної пошти, який ми називаємо V_{avg_F}.

Для векторізації будемо використовувати комбонований підхід PV-DM та TF-IDF. В даному підході обчислюються два векторних подання кожного електронного листа за допомогою моделі глибокого навчання PV-DM та методу TF-IDF. Модель PV-DM була навчена генерувати вектор для кожного слова та кожного електронного листа в навчальному корпусі. Сформовані вектори були згруповані в дві матриці, які зображені на рисунку 3.4:

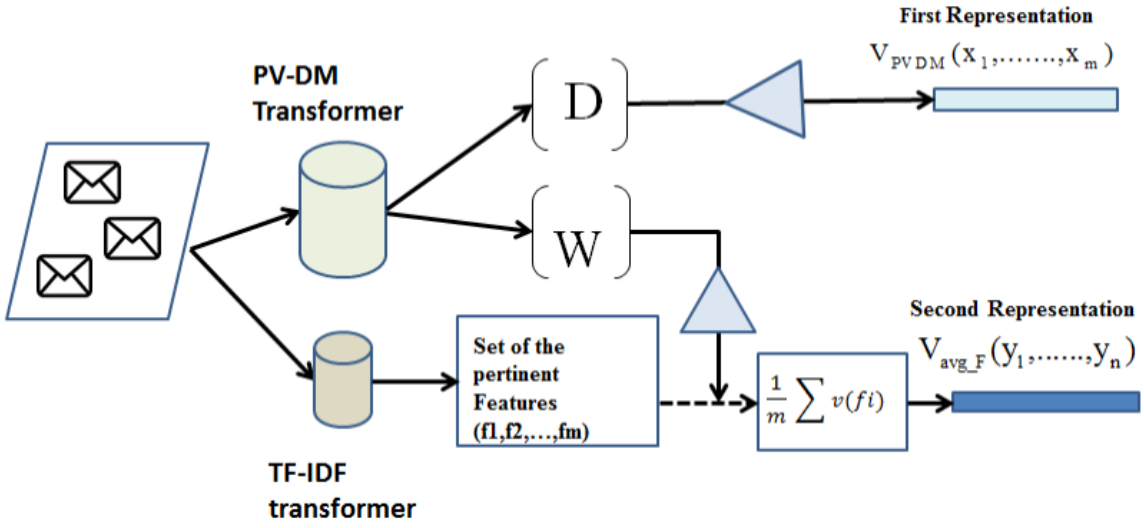

Fig 6. Structure of the representation phase

Рисунок 3.4 – Структура фази векторізації

3.2.4 Оцінка алгоритмів методом кросс-валідації

Вивчення параметрів функції класифікації та тестування її на одних і тих же даних є методологічною помилкою: модель, яка б просто повторювала мітки зразків, які вона щойно бачила, мала б ідеальну оцінку, але не змогла б передбачити щось корисне даних, яких вона не бачe. Така ситуація називається перенавчанням. Щоб уникнути цього, загальноприйнятою практикою при проведенні експерименту машинного навчання є надання частини доступних даних як тестового набору.

Крос-валідація або ковзний контроль – це процедура емпіричного оцінювання узагальнюючої здатності алгоритмів. За допомогою крос-валідації емулюється наявність тестової вибірки, яка не бере участі в навчанні, але для якої відомі правильні відповіді. Процесс кросс-валідації складається з декількох етапів на кожному з которих усі дані розподіляються на тестову та тренувальну вибірку, де

алгоритм тренується на тренувальній вибірці та тестується на тестовій. Особливістю цих етапів є факт того що тестові виборки не перетинаються між собою. Тобто ми зможемо визначити здатність алгоритму до навчання на цьому датасеті. На малюнку 3.5 наведена схема типового крос-валідаційного робочого процесу під час навчання моделей. Найкращі параметри алгоритмів зазвичай можна визначити жадібними методами пошуку по сітці.

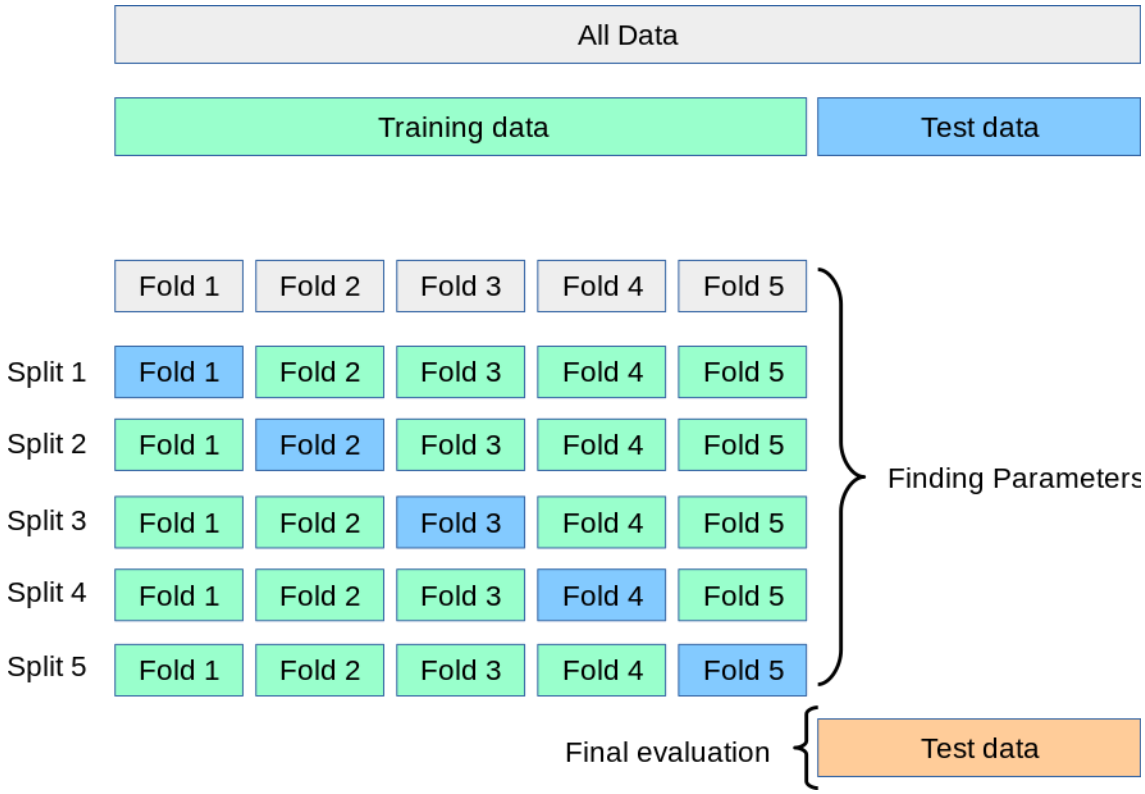

Рисунок 3.5 – Тренування за методом крос-валідації

Показником продуктивності, на базі k-кратної перехресної валідації є середнє значення, обчислене у циклі. Цей підхід може бути обчислювально дорогим, але не витрачає занадто багато даних, що є основною перевагою в таких задачах, як зворотний висновок, коли кількість вибірок дуже мала.

На рисунку 3.6 показані результати роботи кросс-валідації алгоритмів, де відображене мінімальне та максимальне значення, нижній та верхній квартилі,

середнє значення та викроси – це найзручніший графік, який надає багато інформації для аналізу.

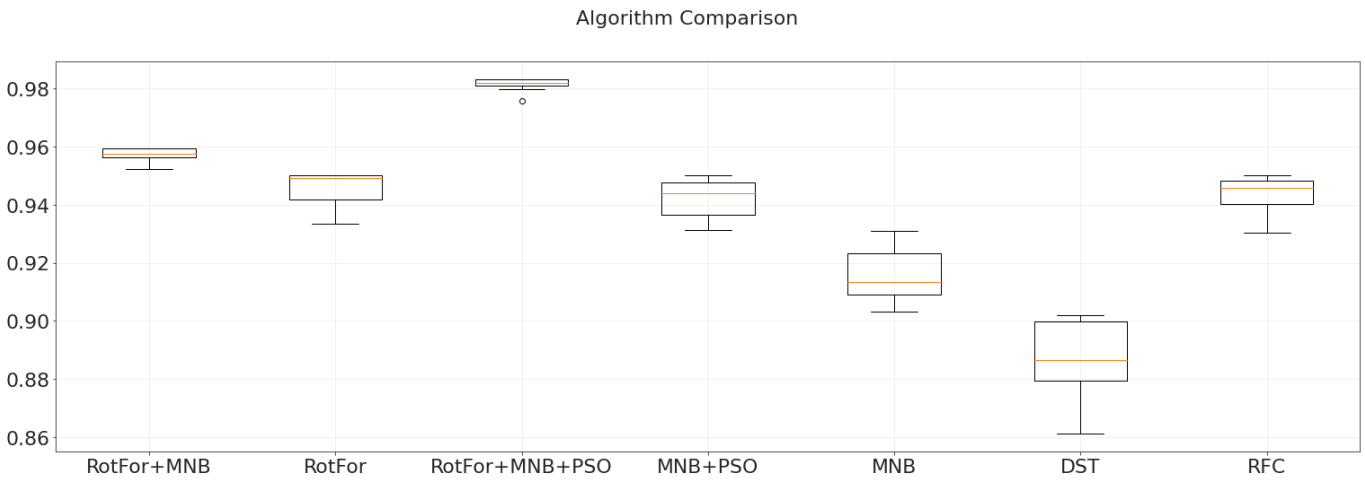

Рисунок 3.6 – Результати роботи кросс-валідації

Виходячи з результатів роботи процесу кросс-валідації на тестових даних, можно побачити, що алгоритм Rotation Forest, модифікований на основі приведеної концепції, працює кращє ніж будь які інші модифікації цього алгоритму. Приведений алгоритм спрацював з середнєю точністю в 98.16%, що на 3.5% кращє ніж стандартний алгоритм Rotation Forest. А алгоритм дерева ухвалення рішень працює гірше за все. Нижче, в таблиці 2 можно побачити середнє значення результатів роботи кросс-валідації.

Таблиця 2 – Середнє значення результатів кросс-валідації

Алгоритм	Середня точність класифікації
Rotation Forest + Naïve Bayes	0.957303
Rotation Forest	0.945032
Rotation Forest + Naïve Bayes + PSO	0.981626

Кінець таблиці 2

Алгоритм	Середня точність класифікації
Multinomial Naïve Bayes + PSO	0.942462
Multinomial Naïve Bayes	0.915984
Decision Tree Classifier	0.88674
Random Forest Classifier	0.943732

На базі цих обчислень висновки, що були наведені у розділі 2 були експерементально доведені. Алгоритм наївного Байєсу дійсно кращє підходить в якості базового алгоритму для ансамблієвої моделі Rotation Forest в рамках задачі класифікації спаму, ніж стандартний базовий алгоритм Decision Tree. Та оптимізація рою частинок дійсно покращує алгоритм наївного Байєсу. Алгоритм Rotation Forest з будьякими модифікаціями працює кращє ніж Random Forest.

3.2.5 Порівняння результатів алгоритмів

Для порівняння алгоритмів, які були обучені за допомогою методу кросс-валідації, були використані метрики Precision, Recall, Accuracy та F-Measure. Поперше, треба перевірити як алгоритми працюють на даних, векторізованих за допомогою комбінованого методу векторізації PV-DM та TF-IDF. Результати порівняння наведено на рисунку 3.7.

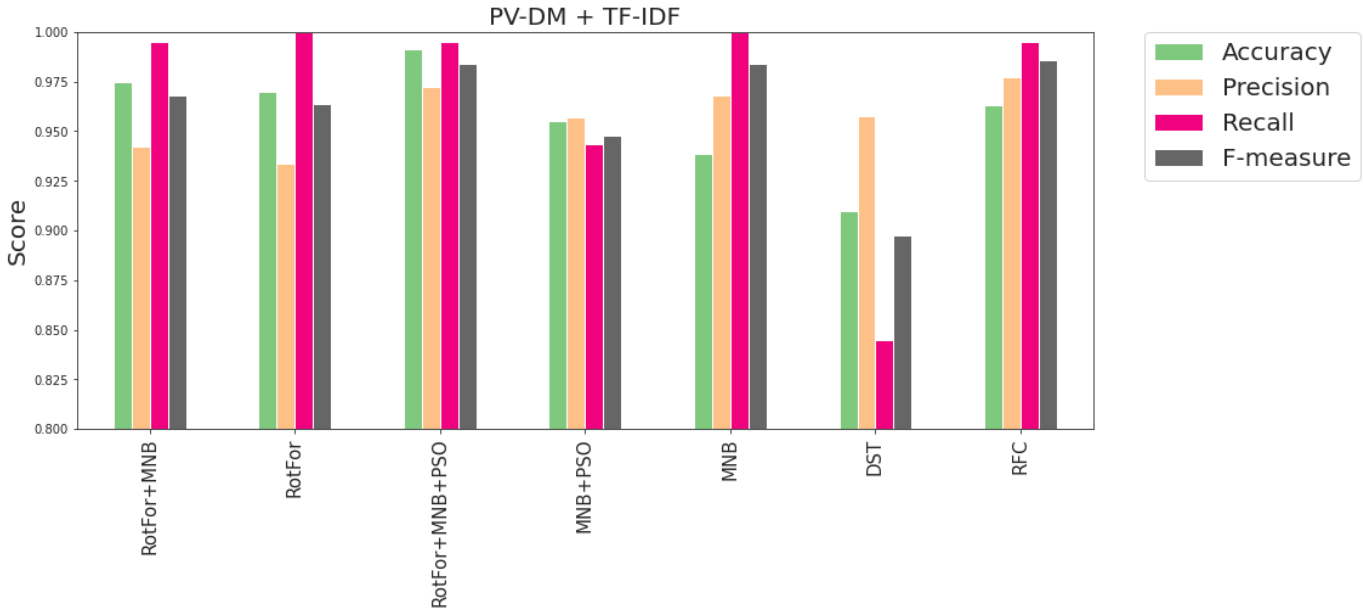

Рисунок 3.7 – Результати роботи алгоритмів за наведеними метриками з використанням PV-DM та TF-IDF

Для можливості порівняти алгоритми в стандартних умовах, були проведени експерименти з використанням TF-IDF без 2-грамів, наведені у рисунку 3.8.

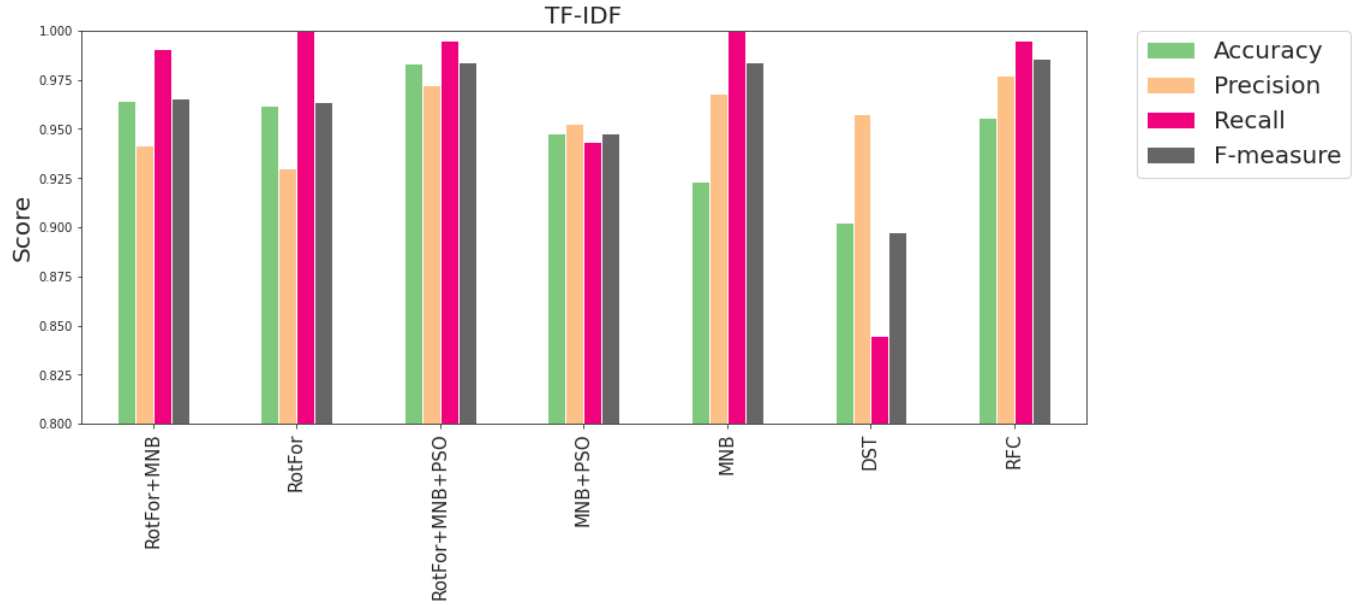

Рисунок 3.8 – Результати роботи алгоритмів з використанням TF-IDF

Також, на рисунку 3.9 наведено результати алгоритмів без використовування 2-грамів та з використанням стандартного методу векторизації Bag of Words (BOW).

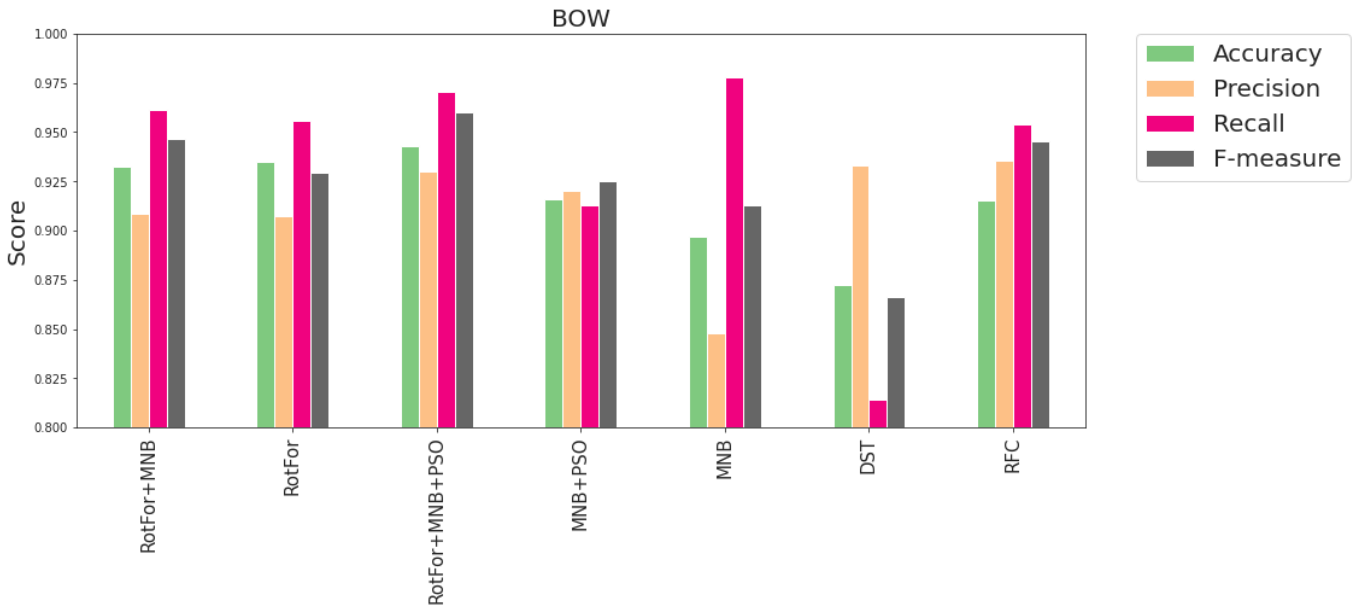

Рисунок 3.9 − Результати роботи алгоритмів за наведеними метриками з використанням Bag of Words

Усі дані, за якими були построєні дані графіки наведені на малюнку 3.10, де вони розподілені трьома методами векторізації.

Vectorisation type	BOW				TF-IDF				PV-DM + TF-IDF			
Algorithm	Accuracy	Precision	Recall	F-measure	Accuracy	Precision	Recall	F-measure	Accuracy	Precision	Recall	F-measure
RotFor+MNB	0.932473	0.908437	0.961363	0.94672	0.964286	0.941964	0.99061	0.965675	0.974558	0.942222	0.995305	0.968037
RotFor	0.934912	0.907151	0.95552	0.929517	0.961905	0.930131	1	0.963801	0.969757	0.933628	1	0.963801
RotFor+MNB+PSO	0.94273	0.930154	0.970564	0.96023	0.983333	0.972477	0.995305	0.983759	0.991361	0.972477	0.995305	0.983759
MNB+PSO	0.9161	0.920212	0.912971	0.924836	0.947619	0.952607	0.943662	0.948113	0.955355	0.956938	0.943662	0.948113
MNB	0.896778	0.8481	0.977777	0.912933	0.923129	0.968182	1	0.983834	0.938552	0.968182	1	0.983834
DST	0.87226	0.932976	0.814121	0.866007	0.902381	0.957447	0.84507	0.897756	0.909747	0.957447	0.84507	0.897756
RFC	0.915084	0.935362	0.954151	0.945239	0.955539	0.976959	0.995305	0.986047	0.96334	0.976959	0.995305	0.986047

Рисунок 3.10 – Результати роботи алгоритмів на базі різних методів векторізації

Як ми можемо побачити на цих результатах, Rotation Forest на базі класифікатору наївного Байєсу та PSO оптимізацією працює набагато кращє ніж інші алгоритми. Це видно по всім метрикам оцінки з будь якими методами векторізації тексту.

3.2.6 Підбір гіперпараметрів та ROC аналіз

Як було сказано раніше методу пошуку гиперпараметрів PSO-BO працює набагато кращє та швидше, ніж стандартні жадібні методи пошуку гіперпараметрів по сітці. Тому для покращення результатів, підібремо кращі гіперпараметри для нашії класифікаторів.

Так як результатом моделі є відносне значення, яке відображає імовірність того що повідомлення є спамом, то треба корректеризувати поріг входження спаму. Жоден користувач не бажає щоб алгоритм пропустив хоча б одне звичайне повідомлення. Тому треба налаштувати поріг так, щоб усі звичайні повідомлення не потрапляли у папку спаму. Це робиться за допомогою ROC кривих.

ROC-крива: графік, який допомогає оцінювати якість бінарної класифікації, оскільки результат класифікації зазвичай відображає імовірність, а поріг результату класифікації може змінюватися. Аналіз цієї кривої надає можливість для ранжування моделей незалежно від контексту витрат або розподілу класів. На рисунку 3.11 показани ROC криві класифікаторів, використаних у цьому дослідженні.

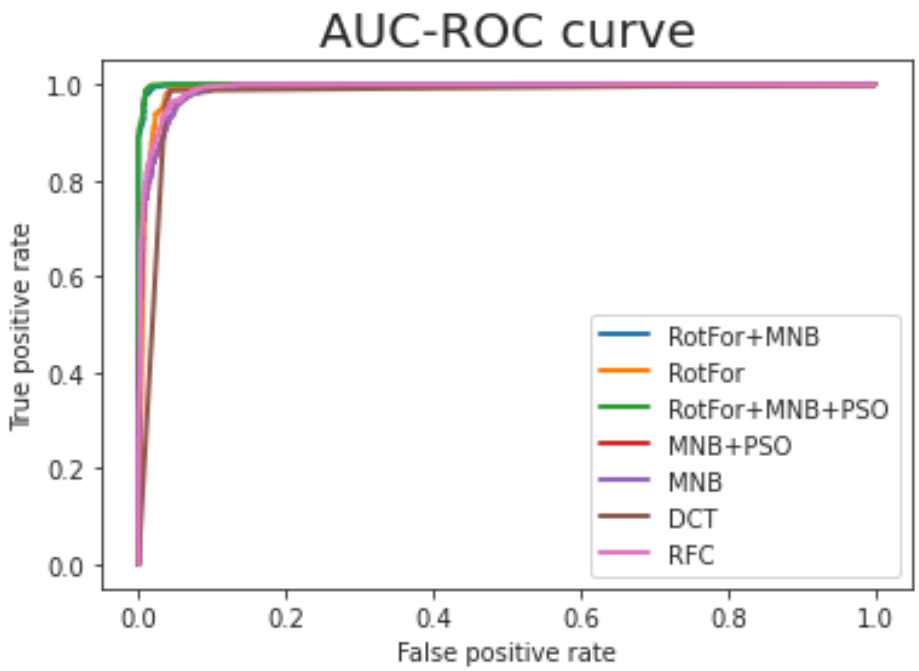

Рисунок 3.11 – Графік ROC (AUC) кривої, заснованої на даних, векторизованих за допомогою PV-DM та TF-IDF

Як ми можемо побачити на цьому графіку, запропонований алгоритм працює набагато кращє, ніж інші алгоритми, обрані для порівняння. Опираючись на попередні обчислення, можно зробити висновок, що запропонований алгоритм працює кращє ніж стандартний Rotation Forest та інші варіації його модифікацій. Точність класифікації запропонованого алгоритму привищує як мінімум на 1.7% усі інші алгоритми. Головною перевагою цього алгоритму є використання методики оптимізації PSO, яка має можливість оптимізувати рішення за допомогою глобального простору пошукових рішень. Іншою перевагою є використання алгоритма наївного Байесу в якості базового алгоритма для Rotation Forest, яке дало трохи менший приріст в ефективності, ніж оптимізація PSO.

3.3 Використання ресурсів

Для проведення дослідження було вікористано електронно-обчислювальний пристрій з восьмиядерним процесором та 16 гігабайтами оперативної пам’яті (RAM), але цього було недостатньо для проведення деяких обчислень. Тому був орендований сервер на 64 гб RAM для обчислень з великою кількістю даних. Цього вистачило для усіх обчислень на базі використаного датасету, але деяким алгоритмам не вистачило цього обсягу оперативної пам’яті для роботи з датасетом “Trec 2007”, тому було вирішено оставити цей датасет для подальших досліджень.

ВИСНОВКИ

Це дослідження було зумовлено збільшенням кількості спаму по електронній пошті по всьому світу. З огляду літератури щодо наявності алгоритмів класифікації був зроблен висновок, що в кожному з етапів класифікатору спаму проводиться багато експериментів, і існує необхідність в створенні кращого класифікатору, який буде містити в собі кращі з компонентів. Для покращення був обран метод Rotation Forest, експерименти з яким показали, що використання іншого базового алгоритму може значно покращіти ефективність алгоритму.

Експерименти були проведені на базі датасету, який є об'єднанням датасетів Enron, Ling та SpamAssasin, та показали, що модифікований алгоритм Rotation Forest працює з точністю 99.14%. Це на 2.17% кращє ніж базовий алгоритм, який працює з точністю в 96.97%, в якому використовується дерева ухвалення рішень. Та на 2.81% кращє ніж класифікатор Random Forest, який працює з точністю 96.33%.

Роблячи висновок з результатів класифікаторів, які працювали на даних, векторизованих різними способами, можна сказати, що оптимізація рою частинок дає більший приріст в ефективності, ніж заміна базового методу наївним Байесом.

В результаті роботи був розроблен класифікатор спам пофідомлень, який працює за допомогою комбінованого методу векторізації PV-DM з TF-IDF, модифікованого алгоритму Rotation Forest з базовим класифікатором наївного Байєсу, який тренується за допомогою методу PSO та методу підбора гіперпараметрів PSO-BO.

Для поліпшення стабильності класифікатору рекомендується використовувати додатково інші датасети, такі як Trec 2007, Trec 2005, Spam archive та ін. Тому що чим різноманітніші дані, тим стабільніше працює алгоритм.

Наступні роботи в напрямку дослідження алгоритмів класифікації спаму доцільно спрямувати на аналіз сучасних алгоритмів класифікації у рамках задачі класифікації спаму, та спробувати їх використовувати за допомогою різних інструментів.

За результатами досліджень опублікована стаття на першій міжнародній науковій інтернет-конференції «МІЖДИСЦИПЛІНАРНІ НАУКОВІ ДОСЛІДЖЕННЯ ТА ПЕРСПЕКТИВИ ЇХ РОЗВИТКУ» та в збірнику “Системи обробки інформації”, категорії “Б”.

ПЕРЕЛІК ПОСИЛАНЬ

1. 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.
2. The Radicati Group. Email Statistics Report, 2017-2021. URL: https://www.radicati.com/wp/wp-content/uploads/2017/01/Email-Statistics-Report-2017-2021-Executive-Summary.pdf.
3. Thiago S. Guzella, W.M. Caminhas. A review of machine learning approaches to Spam filtering. *Expert Systems with Applications.* 2009. Vol. 36, No. 7, P. 10206–10222. DOI: http://dx.doi.org/10.1016/j.eswa.2009.02.037.
4. М.В. Шопинский, Н. В. Голян, І.В. Афанасьєва. Principles of searching and sorting optimization in social networks using a multi-factor assessment system. *Науково-технічний журнал "Біоніка інтелекту"*. 2019. Vol. 1, No. 92, P. 47-53. DOI: https://doi.org/10.30837/bi.2019.1(92).08.
5. Володин Д. А., Афанасьева И. В. Сравнительный анализ моделей для распознавания данных. *Актуальные научные исследования в современном мире*. 2019. Vol. 4, No. 48, P. 144-147.
6. Jaeyeon Jung, Emil Sit. An empirical study of spam traffic and the use of dns black lists. *4th ACM SIGCOMM conference on Internet measurement*. 2004. P. 370–375. DOI: http://dx.doi.org/10.1145/1028788.1028838.
7. Domainkeys identified mail (dkim) signatures. URL: https://tools.ietf.org/html/rfc6376 (дата звернення: 26.01.2021).
8. Zakir Durumeric, David Adrian, Ariana Mirian, James Kasten, Elie Bursztein, Nicolas Lidzborski, Kurt Thomas, Vijay Eranti, Michael Bailey, and J Alex Halderman. Neither snow nor rain nor mitm...: An empirical analysis of email delivery security. *2015 ACM Conference on Internet Measurement Conference*. 2015. P. 27–39.
9. Andriy Yerokhin, Oleg Zolotukhin. Fuzzy probabilistic neural network in document classification tasks. *Information Extraction and Processing*. 2018. No. 46, P. 68-71. DOI: http://dx.doi.org/10.15407/vidbir2018.46.068.
10. Meghali Das, Vijay Prasad. Analysis of an Image Spam in Email Based on Content Analysis. *International Conference on Natural Language Processing and Cognitive Computing*. 2014. P. 129-140. DOI: http://dx.doi.org/10.5121/ijnlc.2014.3313.
11. Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, Alexander J. Smola. Stacked attention networks for image question answering. *2016 IEEE Conference on Computer Vision and Pattern Recognition*. 2016. P. 21-29. DOI: https://doi.org/10.1109/CVPR.2016.10.

12. N. Sutta, Z. Liu, X. Zhang. A Study of Machine Learning Algorithms on Email Spam Classification. *35th International Conference on Computers and Their Applications*. 2020. Vol. 69, P. 170–179. DOI: https://doi.org/10.29007/qshd.

13. A. Selman Bozkir, Esra Sahin, Murat Aydos, Ebru Akcapinar Sezer. Spam E-Mail Classification by Utilizing N-Gram Features of Hyperlink Texts. IEEE 11th International Conference on Application of Information and Communication Technologies. 2017. P. 1–5. DOI: doi.org/10.1109/ICAICT.2017.8687020.

14. Samira. Douzi, Feda A. AlShahwan, Mouad. Lemoudden, Bouabid. El Ouahidi. Hybrid Email Spam Detection Model Using Artificial Intelligence. *International Journal of Machine Learning and Computing*. 2020. Vol. 10, No. 2, P. 316–322. DOI: http://dx.doi.org/10.18178/ijmlc.2020.10.2.937.

15. Jean Charbonnier, Christian Wartena. Using Word Embeddings for Unsupervised Acronym Disambiguation. *The 27th International Conference on Computational Linguistics*. 2018. P. 2610–2619.

16. Tomas Mikolov, Ilya Sutskever, Kai Chen, G.s. Corrado, Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. *Neural Information Processing Systems*. 2013. P. 3111–3119. URL: https://arxiv.org/abs/1310.4546.

17. Shafi'i Muhammad Abdulhamid, Maryam Shuaib, Oluwafemi Osho. Comparative Analysis of Classification Algorithms for Email Spam Detection. *International Journal of Computer Network and Information Security*. 2018. Vol. 1, P. 60–67. DOI: doi.org/10.5815/ijcnis.2018.01.07.

18. Simranjit Kaur Tuteja, Bogiri Nagaraju. Email Spam filtering using BPNN classification algorithm. *International Conference on Automatic Control and Dynamic Optimization Techniques*. 2016. P. 915–919. DOI: http://dx.doi.org/10.1109/ICACDOT.2016.7877720.

19. T. Kumaresan; C. Palanisamy. E-mail spam classification using S-cuckoo search and support vector machine. *International Journal of Bio-Inspired Computation*. 2017. Vol. 9 No. 3, P. 142–156. DOI: https://dx.doi.org/10.1504/IJBIC.2017.083677.

20. Aakanksha Sharaff, Naresh Nagwani, Abhishek Dhadse. Comparative Study of Classification Algorithms for Spam Email Detection. *Emerging Research in Computing, Information, Communication and Applications*. 2016. P. 237–244. DOI: http://dx.doi.org/10.1007/978-81-322-2553-9_23.

21. Muhammad Iqbal, Malik Muneeb Abid, Mushtaq Ahmad, Faisal Khurshid. Study on the Effectiveness of Spam Detection Technologies. *International Journal of Information Technology and Computer Science*. 2016. Vol. 8, No. 1, P. 11–21. DOI: http://dx.doi.org/10.5815/ijitcs.2016.01.02.

22. Mohammad Zavvar, Meysam Rezaei, Shole Garavand. Email Spam Detection Using Combination of Particle Swarm Optimization and Artificial Neural Network and Support Vector Machine. *International Journal of Modern Education and Computer Science*. 2016. Vol. 8, No. 7, P. 68–74. DOI: http://dx.doi.org/10.5815/ijmecs.2016.07.08.

23. Vishal Kumar Singh, Shweta Bhardwaj. Spam Mail Detection Using Classification Techniques and Global Training Set, *Intelligent Computing and Information and Communication*. 2018. P. 623–632. DOI: http:/dx.doi.org/10.1007/978-981-10-7245-1_61.

24. Reena Sharma, Gurjot Kaur. E-Mail Spam Detection Using SVM and RBF. *International Journal of Modern Education and Computer Science*. 2016. Vol. 8, No. 4, P. 57–63. DOI: http://dx.doi.org/10.5815/ijmecs.2016.04.07.

25. D.S. Nazarenko, I.V. Afanasieva, N.V. Golian. Нейросетевой подход для эмоционального распознавания в тексте. *Бионика интеллекта*. 2019. No. 1, P. 9-14.

26. Andriy Yerokhin, Alina Nechyporenko, Andrii Babii, Oleksii Turuta. A new intelligence-based approach for rhinomanometric data processing. *IEEE 36th International Conference on Electronics and Nanotechnology*. 2016. P. 198-201. DOI: https://doi.org/10.1109/ELNANO.2016.7493047.

27. Juan J. Rodríguez, Ludmila Kuncheva, Carlos J. Alonso. Rotation Forest: A New Classifier Ensemble Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. October 2006. Vol. 28, No. 10, P. 1619–1630. DOI: http://dx.doi.org/10.1109/TPAMI.2006.211.

28. Binh Thai Pham, Dieu Tien Bui, M.B. Dholakia, Indra Prakash, Ha Viet Pham, Khalid Mehmood, Hung Quoc Le. A Novel Ensemble Classifier of Rotation Forest and Naïve Bayer for Landslide Susceptibility Assessment at the Luc Yen District, Yen Bai Province (Viet Nam) Using GIS. *Geomatics, Natural Hazards and Risk*. P. 649–671. DOI: https://doi.org/10.1080/19475705.2016.1255667.

29. Borja Ayerdi, Manuel Graña. Anticipative Hybrid Extreme Rotation Forest. *Procedia Computer Science*. 2016. Vol. 80, P. 1671–1681. DOI: https://doi.org/10.1016/j.procs.2016.05.507.

30. Nandan Parmar, Ankita Sharma, Harshita Jain, Dr. Amol K. Kadam. Email Spam Detection using Naïve Bayes and Particle Swarm Optimization. *Second International Conference on Intelligent Computing and Control Systems*. 2018. P. 685–690. DOI: https://doi.org/10.1109/ICCONS.2018.8662957.

31. Mihaela Breaban, Madalina Ionita, Cornelius Croitoru. A new PSO approach to constraint satisfaction. *IEEE Congress on Evolutionary Computation*. 2007. P. 1948–1954. DOI: https://doi.org/10.1109/CEC.2007.4424712.

32. Yaru Li, Yulai Zhang, Xiaohan Wei. Hyper-parameter estimation method with particle swarm optimization. 2020. URL: https://arxiv.org/pdf/2011.11944.pdf.

33. Steven Bird, Ewan Klein, and Edward Loper. Natural Language Processing with Python. URL: https://www.nltk.org/book.

34. Aurélien Géron. Hands-On Machine Learning with Scikit-Learn and TensorFlow. O'Reilly Media. 2017. URL: http://index-of.es/Varios-2/Hands%20on%20Machine%20Learning%20with%20Scikit%20Learn%20and%20Tensorflow.pdf.

35. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isardet al. Tensorflow: A system for large-scale machine learning. *Symposium on Operating Systems Design and Implementation*. 2016. P. 265–283.
36. A. Gulli and S. Pal. Deep Learning with Keras. Publish Pro Ltd. 2017.
37. Duncan M. McGregger. Mastering Matplotlib. Packt Publishing. 2015.
38. Spam archive. URL: http://untroubled.org/spam/.
39. SpamAssasin dataset. URL: https://www.kaggle.com/beatoa/spamassassin-public-corpus.
40. Enron spam dataset. URL: http://nlp.cs.aueb.gr/software_and_datasets/Enron-Spam/index.html.
41. 2007 TREC Public Spam Corpus. URL: https://plg.uwaterloo.ca/~gvcormac/treccorpus07/.
42. Ling-Spam Dataset. URL: https://metatext.io/datasets/ling-spam-dataset.