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: *"Кластеризація спам-доменів методами машинного навчання"*

Виконав: студент VI курсу, групи СБм-61
Спеціальності:
125 «Кібербезпека»
(шифр і назва напряму підготовки, спеціальності)
Грицюк В. П.
підпис (прізвище та ініціали)

Керівник *Стадник М. А.*
підпис (прізвище та ініціали)

Нормоконтроль *Лобур Т. Б.*
підпис (прізвище та ініціали)

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

Рецензент
підпис (прізвище та ініціали)

м. Тернопіль – 2022

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)
«__» _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)
за спеціальністю 125 Кібербезпека
(шифр і назва спеціальності)
Студенту Грицюк Владислав Петрович
(прізвище, ім'я, по батькові)
1. Тема роботи Кластеризація спам-доменів методами машинного навчання
Керівник роботи Стадник М. А., к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
Затверджені наказом ректора від « 06 » грудня 2022 року № 4/7-987
2. Термін подання студентом завершеної роботи 14.12.2022
3. Вихідні дані до роботи Наукові публікації кластеризацію та спам
4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ. 1. СПАМ та його основні методи фільтрації. 1.1 Спам статистика 1.2 Методи фільтрації спаму 1.3 Збір спаму 1.4 Принцип роботи електронної пошти. 1.5 Аналіз останній досліджень. 2. Кластерний аналіз. 2.1 Сфери застосування кластерного аналізу 2.2 Алгоритми кластеризації. 2.2.1 Групування. 2.2.2 k-means. 2.2.3 Ієрархічні методи 2.2.4 Дерева k-d. 2.2.5 Local-sensitive хешування (LSH). 2.2.6 DBSCAN. 2.3 Оцінка результатів кластеризації. 3. Кластеризація SPAM-доменів. 3.1 Процес спам-кластеризації. 3.2 Генерування кластерів. 3.2.1 Групування кластерів. 3.2.2 LSH. 3.2.3 k-means. 3.3 Оцінка кластерів. 4. Охорона праці та Безпека в надзвичайних ситуаціях. 4.1 Охорона праці. 4.2 Концепція захисту населення і територій у разі загрози та виникненні надзвичайних ситуацій. Висновки. Список літературних джерел. Додатки.

5. Перелік графічного матеріалу. 1. Титулка. 2. Мета та задачі дослідження.

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 14.11.2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	14.11.2022-15.11.2022	*Виконано*
2.	Підбір наукових джерел про кластеризацію спам-доменів.	16.11.2022-20.11.2022	*Виконано*
3.	Переклад та опрацювання наукових джерел про кластеризацію спам доменів та їх причини виникнення.	21.11.2022-23.11.2022	*Виконано*
4.	Виконання кластеризації спам доменів	24.11.2022-27.11.2022	*Виконано*
5.	Оформлення розділу “Спам та його основіні методи фільтрації”	28.11.2022-30.11.2022	*Виконано*
6.	Оформлення розділу “Кластерний аналіз”	01.12.2022-04.12.2022	*Виконано*
7.	Оформлення розділу “ Кластеризація SPAM - доменів”	05.12.2022-07.12.2022	*Виконано*
8.	Виконання завдання до підрозділу «Охорона праці»	08.12.2022-09.12.2022	*Виконано*
9.	Виконання завдання до підрозділу «Безпека в надзвичайних ситуаціях»	10.12.2022-11.12.2022	*Виконано*
10.	Оформлення кваліфікаційної роботи	12.12.2022-13.12.2022	*Виконано*
11.	Нормоконтроль	14.12.2022-15.12.2022	*Виконано*
12.	Перевірка на плагіат	9.12.2022	*Виконано*
13.	Попередній захист кваліфікаційної роботи	16.12.2022	*Виконано*
14.	Захист кваліфікаційної роботи	21.12.2022	

Студент _____ *Грицюк В. П.*
(підпис) (прізвище та ініціали)

Керівник роботи _____ *Стадник М. А.*
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кластеризація спам-доменів методами машинного навчання // Кваліфікаційна робота освітнього рівня «Магістр» // Грицюк Владислав Петрович// Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБм-61 // Тернопіль, 2022 // С. – 66, рис. – 11, табл. – 3 , кресл. – 14, додат. – 2.

Ключові слова: СПАМ, СПАМ-ДОМЕН, КЛАСТЕРИЗАЦІЯ, ЕЛКТРОННА ПОШТА

В кваліфікаційній роботі вирішується проблема кластеризації спам доменів з використанням k-means, LSH, групування з метою подальшого застосування при процесі фільтрації різноманітних листів електронної пошти.. В роботі наведено основні методи фільтрації від спаму, а також основні методології їх виникнення. Детально розглянуто основні методи кластеризації, такі як: k-means, групування, ієрархічні методи, дерева, LSH, DBSCAN. Наведено методи оцінки кластеризації.

Здійснено кластеризацію спам доменів на основі реального сформованого набору даних з використанням інформації з сайтів Alexa та stopforumspams.com. Здійснено оцінку результату кластеризації з використанням додатково штучно введених функцій при маркуванні набору даних.

ANNOTATION

Clustering of spam domains using machine learning methods // Qualification work of the educational level “Master” // Hrytsyuk Vladyslav Petrovych // Ternopil Ivan Pulyuy National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Cybersecurity, СБм-61 group // Ternopil, 2022 // P. – 66, fig. – 11, table – 3, drawing – 14, apendix – 2.

Key words: SPAM, SPAM DOMAIN, CLUSTERIZATION, EMAIL

The qualification work solves the problem of clustering spam domains using k-means, LSH, grouping with the purpose of further application in the process of filtering various e-mails. The work provides the main methods of spam filtering, as well as the main methodologies of their occurrence. The main methods of clustering, such as: k-means, grouping, hierarchical methods, trees, LSH, DBSCAN, are considered in detail. Methods of clustering assessment are presented.

Clustering of spam domains was carried out on the basis of a real generated data set using information from the Alexa and stopforumspams.com sites. The result of clustering was evaluated using additionally artificially introduced functions when labeling the data set.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

DBSCAN	Density-based spatial clustering of applications with noise
DCC	Distributed Checksum Clearinghouse
DNS	Domain Name System
DNSBL	DNS blocklist
HTML	HyperText Markup Language
IMAP	Internet Message Access Protocol
IP	Internet Protocol
LSH	Locality-sensitive hashing
POP3	Post Office Protocol Version 3
RBL	Realtime Blackhole List
URL	Uniform Resource Locator,
ЗК	Зловмисні кластери
ХК	Хороші кластери

ВСТУП

Актуальність теми. Проблема кластеризації є, мабуть, однією з найбільш досліджуваних у спільнотах інтелектуального аналізу даних і машинного навчання. Ця проблема вивчалася дослідниками з кількох дисциплін протягом п'яти десятиліть. Застосування кластеризації включає широкий спектр проблемних областей, таких як текст, мультимедіа, соціальні мережі та біологічні дані. Крім того, проблема може виникнути в кількох різних сценаріях, таких як потокове передавання або невизначені дані. Кластеризація — це досить різноманітна тема, і базові алгоритми значною мірою залежать від області даних і сценарію проблеми.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи освітнього рівня «Магістр» є виконання кластеризації спам-доменів та успішне виконання перевірки результатів кластеризації.

Для досягнення поставленої мети було потрібно виконати наступні завдання:

- Дослідити джерела виникнення спаму;
- Провести огляд основних технологій захисту від спаму;
- Проаналізувати методи кластеризації;
- Проаналізувати вимоги щодо успішного проходження кластеризації;
- Дослідити можливості методів кластеризації на
- Провести експериментальні дослідження та отримати результати кластеризації;
- Здійснити аналіз результатів кластеризації спам-доменів.

Об’єкт дослідження. Процес кластеризації спам-доменів.

Предмет дослідження. Набір даних із сайтів Alexa та stopforumspams.com.

Наукова новизна одержаних результатів кваліфікаційної роботи полягає у тому, що використано алгоритм кластеризації на основі експериментального виконання дослідження з використанням набору даних із сайтів Alexa та stopforumspams.com, що включає етап кластеризації та класифікації.

Практичне значення одержаних результатів. Отримані результати кластеризації за допомогою використаного алгоритму дозволяють з більшою точністю виконувати фільтрацію спаму в інформаційних системах.

Апробація результатів магістерської роботи. Основні результати проведених досліджень обговорювались на: X науково-технічній конференції «Інформаційні моделі, системи та технології» (м.Тернопіль, 7-8 грудня, 2022).

Публікації. Основні результати кваліфікаційної роботи опубліковано у одній праці конференції (див. Додаток А).

Структура й обсяг кваліфікаційної роботи. Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку літератури із 24 найменувань та 2 додатків. Загальний обсяг кваліфікаційної роботи складає 71 сторінки, з них 66 сторінок основного тексту, який містить 11 рисунки та 3 таблиці.

1 СПАМ ТА ЙОГО ОСНОВНІ МЕТОДИ ФІЛЬТРАЦІЇ

1.1 Спам статистика

Кіберзлочини, пов’язані зі спамом, стали серйозною загрозою для суспільства. Поточні дослідження спаму в основному спрямовані на більш ефективне виявлення спаму. Необхідно зауважити, що виявлення та порушення допоміжної інфраструктури, яку використовують спамери, є більш ефективним способом зупинки спаму, ніж фільтрація. Припинення спам-хостів значно зменшить прибуток, який може отримати спамер, і перешкодить його можливості розсилати більше спаму.

Результати показують, що багато, здавалося б, непов’язаних спам-кампаній насправді пов’язані, якщо перевірити доменні імена в URL-адресах; спамери мають складний механізм для боротьби з чорними списками URL-адрес, реєструючи багато нових доменних імен щодня та видаляючи старі домени; домени розміщені на різних IP-адресах у кількох мережах, переважно в Китаї, де законодавство не таке суворе, як у Сполучених Штатах; старі IP-адреси час від часу замінюються новими, але все ще демонструють сильну кореляцію між ними.

Електронна пошта, також відома як email, є поширеним механізмом спілкування, який використовують багато компаній і людей. Компанії використовують електронну пошту, щоб підтримувати зв’язок зі своїми клієнтами, просувати свої товари та послуги, отримувати зворотній зв’язок від клієнтів. Люди часто використовують електронну пошту для спілкування як стандартний спосіб спілкування з іншими людьми.

Як і будь-яка широко використовувана технологія, електронною поштою також можна зловживати. Небажані повідомлення електронної пошти, також відомі як спам, з’явилися одночасно з появою системи електронної пошти. Перше спам-повідомлення було надіслано 1 травня 1978 року компанією Digital Equipment Corporation для реклами свого продукту кільком сотням користувачів ARPANET [1]. Коли в 90-х роках Інтернет відкрився для комерційного та

особистого використання, спам став широко поширеним явищем. Таким чином, із розвитком системи електронної пошти зріс спам. У першому півріччі 2017 року на спам припадало 54% усього трафіку електронної пошти [2].

Більшість спам-листів призначені для агресивної реклами товарів або веб-сайтів, і вони не завдають великої шкоди, окрім марнування часу та ресурсів людей. Однак, крім цього, спам відіграє важливу роль у кібератаках і розповсюдженні шкідливого програмного забезпечення. Згідно з дослідженнями Symantec, це найбільш часто використовуваний механізм доставки шкідливих програм. Крім того, їхні дослідження стверджують, що користувачі вдвічі частіше заражаються шкідливим програмним забезпеченням через електронну пошту чим через шкідливий веб-сайт [2].

Спам також використовується для різних видів шахрайської діяльності, націленої на бізнес. Згідно з недавнім аналізом ФБР [3], між 2013 і 2016 роками компанії зазнали збитків на суму понад 5 мільярдів доларів через шахрайство з електронною поштою. Ці наслідки спаму роблять його основним вектором атаки для компаній, що надають послуги безпеки, щоб інвестувати свої аналітичні ресурси.

Компанії з інформаційної безпеки зацікавлені в розумінні тенденцій спаму та прогнозуванні того, що може статися, і як пом’якшити загрози. Отримання репрезентативного огляду спаму – це завдання, яке вимагає певної підготовки. Для цього компанії в галузі кібербезпеки підтримують приманки електронної пошти та купують величезну кількість спаму у спеціалізованих постачальників.

Отримання якомога більшої кількості зразків зловмисного програмного забезпечення та фішингу та вчасне оновлення баз даних виявлення може значно зменшити шкоду, завдану клієнтам охоронної компанії від зловмисного програмного забезпечення та шахрайства.

Шкідливі вкладення та URL-адреси – це не єдина інформація, яку можна отримати зі спаму. Часто спам розкриває достатньо інформації, щоб згрупувати різні спам-кампанії за загальним відправником. Ідентифікація відправників не

тільки дозволяє органам влади вживати проти них заходів, але й зрозуміти наміри спамерів і вдосконалити автоматичні механізми виявлення.

Однією з ключових особливостей спаму є те, що він часто є не якісним, а кількісним. Навіть якщо лише 0,1% користувачів відкривають спам-лист і переходять за посиланням, це приносить одного клієнта на кожні 1000 повідомлень. Крім того, спам настільки дешевий, що один клієнт може окупити мільйони повідомлень.

У 2021 році було підраховано, що щодня було надіслано та отримано майже 319,6 мільярда електронних листів. А в грудні 2021 року 45,37% від загальної кількості електронних листів були визнані спамом. З 2020 по 2021 рік світовий обсяг спаму був найвищим у липні 2021 року, коли 283 мільярди з 336,41 мільярда електронних листів були спамом.

Надсилається кілька спам-листів, як-от оголошення, листи ланцюга, фішинг, містифікації, шахрайство з грошима, попередження про зловмисне програмне забезпечення, вміст для дорослих тощо. Розподіл спаму за типом представлений на рис. 1. 1.

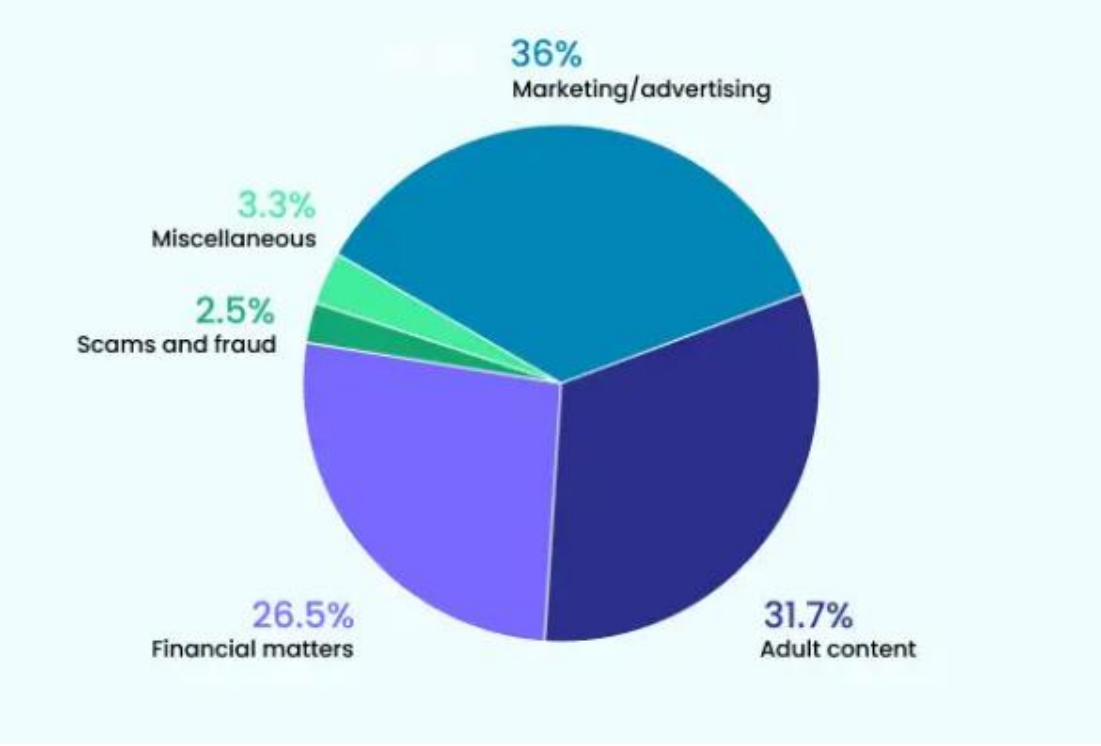

Рисунок 1.1 – Розподіл спаму за типами [4]

Найпоширенішим типом спаму є маркетингові чи рекламні листи, на які припадає майже 36% усіх спам-повідомлень.

Другим за поширеністю типом спаму є листи з вмістом для дорослих, які становлять близько 31,7% усього спаму.

Електронні листи, пов’язані з фінансовими питаннями, є третім за поширеністю типом спаму, що становить приблизно 26,5% усіх спам-повідомлень.

2,5% усіх спам-повідомлень є шахрайством. З них 73% – фішингові листи.

1.2 Методи фільтрації спаму

На початку зародження спам був надзвичайно простим. Будь-яка людина могла надіслати мільйони повідомлень будь-кому, знаючи лише свою електронну адресу. Саме тоді почали з’являтися перші методи фільтрації спаму. Спамери у відповідь також повинні були вдосконалити свої методи. Надзвичайно важливо розуміти ситуацію на ринку спаму та з якими труднощами можна зіткнутися під час кластеризації спаму.

Основними методами фільтрації від спаму є:

- SPF та DKIM.
- RBL та DNSBL.
- RFC.
- “Сірі” списки.
- Razor.
- Spamassassin.
- DCC.
- Байєсовська фільтрація.

SPF та DKIM. Надсилаючи електронний лист, спамер може вказати будь-яку електронну адресу в полі “Від (From)”, а будь-яке доменне ім’я – у команду HELO. SPF і DKIM є двома поширеними методами, які дозволяють прив’язати IP-

адресу до доменного імені. Техніка Sender Policy Framework (SPF) дозволяє перевірити, чи дозволено хосту-відправнику надсилати повідомлення за допомогою певного домену [5]. Щоб увімкнути перевірку SPF, власник домену повинен додати запис DNS, який визначатиме, які IP-адреси мають право використовувати доменне ім'я як ідентифікатор [6].

Іншим способом такої перевірки є Domain Keys Identified Mail (DKIM), але замість того, щоб авторизовані хости зберігатися в DNS, відкритий ключ зберігається як запис DNS [7]. Потім до кожного листа додається поле "DKIM-Signature". У цьому полі зберігається підписана інформація про домен, яка пізніше порівнюється з доменом поля "Від (From)". Щоб перевірити дійсність інформації, одержувач використовує відкритий ключ, який зберігається в записі DNS.

Чорні списки в реальному часі (RBL) або чорні списки DNS (DNSBL) – це техніка захисту від спаму, описана в RFC 5782. Вона використовується для видалення повідомлень, які надійшли з джерел, підозрюваних у надсиланні величезної кількості спаму [8]. Ці джерела не є обов'язковими серверами, створеними спамерами, але також можуть бути неправильно налаштованими легальними SMTP-серверами, якими зловживають спамери.

RBL використовує інфраструктуру DNS для зберігання заблокованих IP-адрес, і будь-який SMTP-сервер може отримати доступ до цих записів зі стандартних DNS-серверів або спеціалізованих серверів, таких як RBLDNS. Цей метод також блокує SMTP-сервери, які дозволяють надсилати через них спам, наприклад, відкриті SMTP-ретранслятори, які передають весь трафік, що надходить, без перевірки правильності IP-адреси. Інтернет сканується, щоб виявити такі вузли SMTP і додати їх до RBL, перш ніж спамери почнуть їх використовувати.

Стандарти електронної пошти визначаються запитом на коментарі або RFC. RFC є основою для взаємодії всього Інтернету на низькому рівні. Спеціальні RFC для електронної пошти визначають, як сервери електронної пошти, які надсилають, підключаються до серверів одержувачів, щоб передати їхні повідомлення. Старіші версії Sendmail слабко тлумачать RFC. Наприклад,

Sendmail версії 8.8 і раніших дуже поблажливі щодо прийняття параметрів за замовчуванням для команд MAIL FROM: і RCPT TO:. Інші суворі методи перевірки заголовків включають вимогу HELO/EHLO, точні параметри для HELO/EHLO тощо. Багато MTA можуть контролювати, наскільки суворим повинен бути сервер, коли приймає вхідні повідомлення. Внесення таких змін може зменшити спам, але також може спричинити проблеми з легітимною доставкою електронної пошти з неправильно налаштованих систем.

“Сірі” списки. Для кожного отриманого електронного листа SMTP-сервер створює три частини даних, які називаються триплетом: IP-адреса відправника, адреса електронної пошти відправника та адреса електронної пошти одержувача [9]. Коли одержувач отримує повідомлення з невидимим триплетом, він відхиляє повідомлення. Це ефективна стратегія, оскільки спамери часто не мають належної інфраструктури для обробки відхилених повідомлень і їх повторного надсилання. Крім того, навіть якщо вони спробують повторно надіслати повідомлення через деякий час, вони вже будуть відфільтровані RBL, оскільки протягом часу затримки спамер надсилав повідомлення іншим клієнтам.

Razor – це розподілена мережа для перевірки вмісту спаму, яка дозволяє виявляти тексти спаму, не розкриваючи справжнє повідомлення [10]. Razor використовує нечіткий алгоритм зіставлення сигнатур під назвою Nilsimsa. Nilsimsa створює статистичну модель повідомлення або частини повідомлення таким чином, що невеликі автоматичні зміни, внесені до повідомлення, не змінюють суттєво результат [11].

Razor залежить від відгуків користувачів про спам-повідомлення. Якщо користувачі Razor не позначають спам-повідомлення як спам, Razor видасть багато хибних негативів.

Distributed Checksum Clearinghouse (DCC) – це подібний спосіб додавання вмісту в чорний список, але він не покладається на відгуки користувачів. Натомість він підраховує схожі повідомлення та вважає спамом усі масово надіслані повідомлення. Таким чином, без білого списку він матиме високий рівень помилкових спрацьовувань [12].

Spamassassin — потужний інструмент для виявлення спаму. Він налаштовується шляхом додавання правил, де кожне правило має оцінку, і всі правила виконуються для кожного отриманого електронного листа [11]. Бали за цими правилами підсумовуються та порівнюються з пороговим значенням. Якщо сума перевищує порогове значення, то повідомлення вважається спамом. Користувачі можуть додавати власні правила у формі регулярного виразу. Правило можна застосувати до всіх частин повідомлення, а не лише до тіла чи заголовка.

Spamassassin також дозволяє додавати складніші правила та плагіни для всіх видів перевірок спаму, таких як тести статистичного аналізу, RBL-тести, байєсовська фільтрація та перевірка доменних імен.

Байєсовська фільтрація спаму базується на статистичній теоремі, яка оцінює ймовірність події. Його можна налаштувати по-різному, але два найпоширеніші підходи – виявлення слів і послідовностей букв [14]. Розбір слів досить простий. Фільтр Байєса потрібно навчити на зразках спаму та легальних повідомлень. Використовуючи інформацію з тренінгу, можна побачити, які слова часто використовуються в спам-листах, а які ні.

Байєсівський фільтр дуже швидко адаптується до нових методів спамерів, наприклад, коли спамери почали використовувати слово “f-r-e-e” замість “free”, щоб уникнути фільтрів слів Spamassassin, байєсівському фільтру вдалося дізнатися, що “f-r-e-e” є хорошим показником спаму, тому що в основному використовується в спамі.

Байєсівський спам-фільтр також здатний виявляти випадково згенеровані тексти, які спамери часто додають до своїх повідомлень, щоб уникнути абсолютно схожих повідомлень. Це можливо тому, що ймовірність наявності пари літер разом не однакова для всіх пар, і згенерований текст не матиме такого розподілу ймовірностей. Одним із прикладів програмного забезпечення, що використовує фільтр Байєса, є DSPAM [15].

1.3 Збір спаму

Спам легко отримати, не докладаючи жодних зусиль, але отримати повне уявлення про те, що відбувається у світі спаму, є складним завданням. Зазвичай використовується п’ять методів збору спаму.

Botnet malware observation. Спостереження за зловмисним програмним забезпеченням ботнету. Зловмисне програмне забезпечення ботнету розміщується в пісочниці та виконує збір усіх електронних листів, надісланих із нього. Цей метод забезпечує найчистіший спам, оскільки він надходить безпосередньо з джерела.

MX honeypot. Налаштування SMTP-сервера для прийняття всіх повідомлень, надісланих на нього. Мета – отримати всі електронні листи, навіть ті, які були створені спамером. Щоб повідомити спамерам, що в домені є сервер електронної пошти, адреси електронної пошти, розміщені на цьому домені, повинні бути спочатку зібрані спамером. Одним із яскравих прикладів розповсюдження адрес є служби тимчасових поштових скриньок, якими користуються реальні люди, щоб уникнути використання їхніх справжніх електронних адрес під час реєстрації. Люди використовують ці тимчасові служби електронної пошти для реєстрації в інших службах, де їм не потрібно отримувати електронні листи від цієї служби. Прийом усіх вхідних повідомлень дозволяє отримувати значні обсяги спаму, призначеного для власників тимчасових адрес. Ніхто не читатиме спам у цих облікових записах, тому спамери можуть відфільтрувати їх за неактивність.

Seeded honey акаунти. Електронні адреси створюються на сторонніх ресурсах електронної пошти та заповнюються власником каналу. Цей метод забезпечує кращу різноманітність доменних імен і, у багатьох випадках, не може бути відфільтрований за доменними іменами з баз даних спамерів. Такі облікові записи також неактивні, тому їх швидко фільтрують просунуті спамери.

Ідентифікація людиною. Зазвичай це справжня служба електронної пошти з реальними користувачами, які можуть вручну позначити вхідний електронний

лист як спам. Повідомлення, позначені реальними користувачами, часто є високоякісними кампаніями, оскільки їм вдається уникнути автоматичних фільтрів спаму та дістатися до користувачів. Недоліком є те, що такі канали не можна масштабувати, оскільки їм потрібні більші канали, щоб залучити більше людей до служби електронної пошти. Іншим недоліком є те, що різні люди мають різне визначення спаму, тому відсоток хибних негативів буде значним.

Чорні списки доменів – це канали, що ведуться вручну, які можуть керуватися різними комбінаціями даних джерел спаму на основі організації, яка їх підтримує.

1.4 Принцип роботи електронної пошти

Електронна пошта – це спосіб обміну текстовими повідомленнями між користувачами комп’ютерів. Основна концепція електронної пошти полягає в тому, що кожен може надсилати повідомлення кожному. Протокол електронної пошти розроблено таким чином, що вся система є децентралізованою. У цьому розділі пояснюється, як працює електронна пошта.

Мережа електронної пошти складається з серверів, які зберігають, передають і обробляють повідомлення електронної пошти. Повідомлення електронної пошти передаються за допомогою серверів Simple Mail Transfer Protocol. SMTP дозволяє встановлювати зв’язок і маршрутизацію між різними користувачами електронної пошти лише за допомогою адреси електронної пошти та інформації з серверів DNS.

Цей протокол працює в комп'ютерних мережах таким чином, що співрозмовникам не потрібно бути онлайн одночасно. Це також дозволяє ретранслювати повідомлення між різними мережами [16].

Хоча можна надсилати й отримувати електронні листи лише за допомогою SMTP-серверів, типовий варіант використання включає службу отримання пошти, наприклад IMAP або POP3. Служби пошуку пошти забезпечують простіший спосіб роботи з вхідними повідомленнями, зберігаючи та

обслуговуючи поштову скриньку. Служби зазвичай працюють на тому ж сервері, що й сервер SMTP, і дозволяють користувачам отримувати повідомлення із системи. Різниця між IMAP і POP3 полягає в тому, що IMAP зберігає повідомлення на сервері, дозволяючи отримати доступ до поштової скриньки з різних пристроїв, а POP3 читає повідомлення, видаляючи їх із сервера.

Хоча більшість клієнтів електронної пошти підтримують POP3 та IMAP, деякі клієнти електронної пошти запроваджують власні протоколи для отримання даних із поштових серверів. На рис. 1.2 зображено спрощений приклад передачі повідомлення електронної пошти.

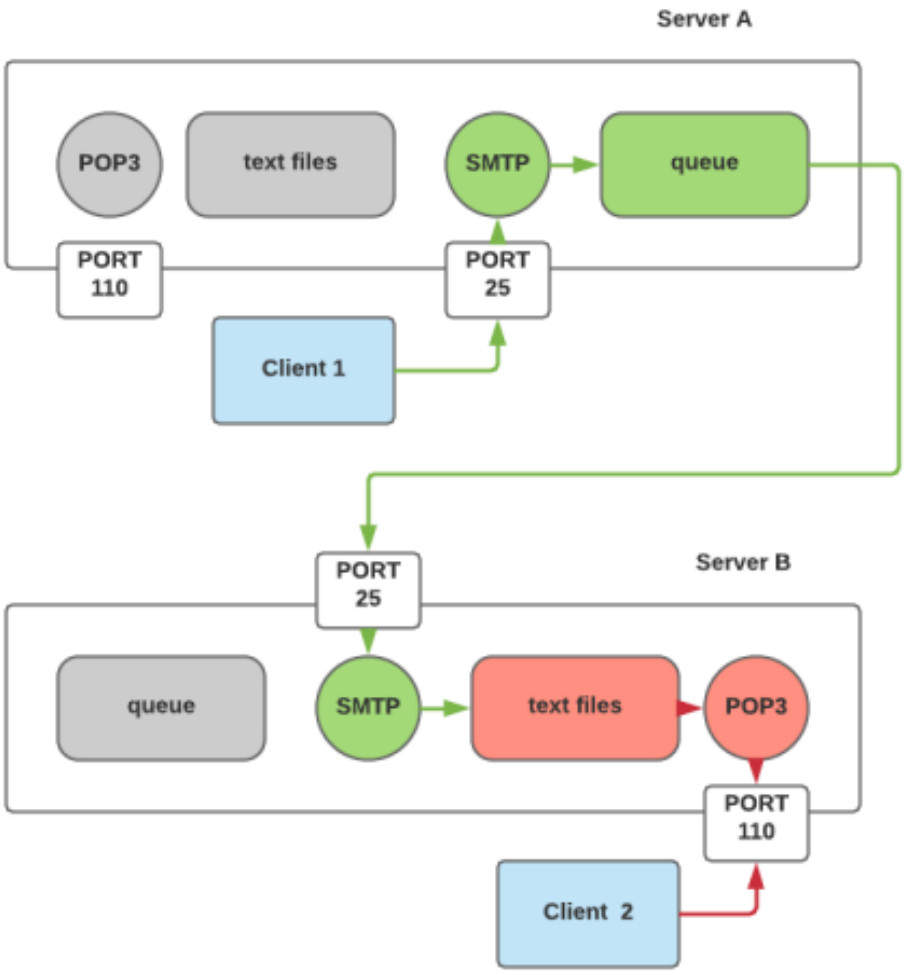

Рисунок 1.2 – Загальний процес передачі електронною поштою

Загальний процес передачі електронною поштою складається з наступних пунктів:

- Поштовий клієнт, наприклад, MS Outlook, підключається до відомого SMTP
- сервер, що працює на хості A. Потім клієнт електронної пошти повідомляє адресу одержувача та вміст повідомлення електронної пошти підключеному серверу SMTP.
- SMTP-сервер A запитує DNS-сервер, щоб отримати IP-адресу SMTP-сервера одержувача.
- SMTP-сервер A підключається до SMTP-сервера B для надсилання інформації, отриманої від поштового клієнта Outlook.
- Сервер SMTP B отримує повідомлення та передає його службі IMAP.
- Поштовий клієнт 2 підключається до служби IMAP, перевіряє автентичність і отримує нові повідомлення. Тепер користувач може бачити ці повідомлення у своєму поштовому клієнті.

Корисне навантаження електронної пошти не змінюється під час передачі. Однак є один виняток: кожен SMTP-сервер має додати до електронного листа заголовок «Отримано», у якому зберігається інформація про поточний SMTP-сервер. Це робиться для відстеження маршруту повідомлення та пошуку повільних серверів.

По суті, повідомлення електронної пошти – це послідовність символів, розділених на рядки фіксованої довжини. Рядки розділені розривом рядка “CRLF”. Кожне повідомлення електронної пошти складається із заголовків і тіла. Тіло є необов’язковим і його можна опустити.

Заголовок є набором рядків, що складаються з друкованих символів US-ASCII. Він відокремлений від основної частини порожнім рядком. Кожен рядок електронної пошти не має бути довшим за 998 символів і не має бути довшим за 78 символів. Кожне поле заголовка починається з назви поля, після якого ставиться двокрапка. Рядок закінчується “CRLF”.

Тіло повідомлення є рядки символів US-ASCII з кількома обмеженнями: рядок не може бути довшим за 998 символів, а “CRLF” можна використовувати

лише для розривів рядків. RFC 5322 посилається на кілька інших RFC, які значно розширюють специфікацію тіла, дозволяючи кодувати нетекстові тіла повідомлень, тіла повідомлень із кількох частин і тіло/заголовки різними наборами символів.

Поля заголовка логічно розділені на групи за призначенням. RFC 5322 визначає вісім груп полів заголовків: поля дати походження (Origination Date Fields), поля джерела (Originator Fields), поля адреси призначення (Destination Address Fields), поля ідентифікації (Identification Fields), інформаційні поля (Informational Fields), поля повторного відправлення та додаткові поля (Resent Fields та Optional Fields).

1.5 Аналіз останній досліджень

Останніми роками було проведено дослідження щодо визначення та класифікації спаму, яке дає змогу глибше зрозуміти стратегії розсилання спаму та інфраструктуру шахрайства.

Автор статті [17] кластеризував електронні листи зі спамом, якщо будь-який із наступних трьох атрибутів ідентичний: IP-адреса надсилання, тіло повідомлення та тема електронного листа. Найбільший кластер, про який повідомлялося, містить 85% усіх електронних листів за 9-денний період у грудні 2007 року. Електронні листи стосувалися копій годинників, азартних ігор, порнографії. Однак розгляд лише ідентичних тем і тіл повідомлень не зможе знайти повідомлення електронної пошти з подібними темами чи тілами повідомлень, які створюються за допомогою шаблонів, що дуже часто можна побачити в сучасних спам-повідомленнях. Крім того, електронні листи зі загальними темами, як-от «Re:» і «Fwd», не обов’язково можуть мати будь-який зв’язок між собою.

Автори статті [18] використовували чотири атрибути (мова, тип повідомлення, макет повідомлення та URL-адреси) для згрупування спам-кампаній. Електронні листи, які мають спільні часті функції, будуть згруповані. Деякі великі спам-групи, як повідомляється, складаються з понад 100 000 спам-

листів, які були зібрані приманками в кількох бразильських мережах. У статті також досліджено мережеві шаблони машин-відправників (зловживання HTTP, проксі-сервери SOCKS і відкриті ретранслятори). Документ не досліджував URL-адреси, як-от вибірку веб-сторінок, пошук IP-адрес хостингу або інформацію WHOIS.

Проект Spamscatter [19] також отримував веб-сторінки за допомогою посилань у спам-листах і кластеризував веб-сторінки на основі схожості знімків екрана. Вони класифікували спам-кампанії на основі вмісту веб-сайтів. Проте десять найбільших категорій шахрайства на віртуальному хостингу, які вони перерахували, містили три категорії «годинники», дві категорії «аптеки» та дві категорії «програмне забезпечення», і не було вказівок на те, пов’язані вони чи ні. Вони відстежували домени протягом приблизно двох тижнів і виявили, що кілька віртуальних хостів (різні домени, які обслуговуються одним сервером) і кілька фізичних хостів (різні IP-адреси) зустрічаються рідко. Це може бути неправдою, оскільки найбільший кластер, який ми знайшли, містить багато доменів, кожен з яких розміщено на одному наборі IP-адрес.

Вони також дослідили тривалість життя шахрайських хостів і виявили, що більшість із них недовговічні. Однак спамер може направити веб-сайт на іншу IP-адресу, змінивши записи DNS і створивши нові доменні імена для видалення замість старих, які внесено до чорного списку. Таким чином, припинення дії імені хоста чи домену не обов’язково означає завершення спам-кампанії. У нашому дослідженні найбільший кластер триває весь період експерименту, тоді як нові доменні імена вводяться щодня, а IP-адреси хостингу час від часу змінюються.

Автори статті [20] спостерігали тенденції створення спам-повідомлень, особливо методи обфускації в спам-повідомленнях на основі HTML. Потім вони створили спам-корпус Webb, який складається з майже 350 000 веб-сторінок, отриманих за URL-адресами спам-листів на основі HTML. Вони також виявили, що веб-хости в їх Корпусі були тісно пов’язані один з одним веб-посиланнями. Але графік був надто сильно згрупований, щоб побачити будь-яку детальну інформацію про те, як хости фактично були підключені. Використання Web Spam

Corpus, вони розділили веб-сторінки на п’ять категорій: рекламні ферми, парковані домени, реклама, порнографія та переадресація [20]. Вони виявили, що спам-сторінки зазвичай мають більше дублікатів і перенаправлень, ніж звичайні веб-сторінки. Вони також визначили 10 найкращих IP-адрес хостингу з найбільшою кількістю веб-сторінок, а два діапазони IP-адрес становлять 84% IP-адрес хостингу. Однак вони не вказали, чи пов’язані ці IP-адреси чи ні.

2 КЛАСТЕРНИЙ АНАЛІЗ

2.1 Сфери застосування кластерного аналізу

Проблема кластеризації є, мабуть, однією з найбільш досліджуваних у спільнотах інтелектуального аналізу даних і машинного навчання. Ця проблема вивчалася дослідниками з кількох дисциплін протягом п'яти десятиліть. Застосування кластеризації включає широкий спектр проблемних областей, таких як текст, мультимедіа, соціальні мережі та біологічні дані. Крім того, проблема може виникнути в кількох різних сценаріях, таких як потокове передавання або невизначені дані. За відсутності конкретного промаркованого набору даних чи інформації, кластеризацію можна вважати лаконічною моделлю даних, яку можна інтерпретувати в сенсі підсумкової або генеративної моделі. Основну проблему кластеризації можна сформулювати так: “Маючи набір даних (точок в n-вимірному просторі) необхідно їх розділити на набір груп, які є максимально подібними”.

Необхідно зауважити, що це дуже приблизне визначення, і варіації у визначенні проблеми можуть бути значними залежно від конкретної моделі, що використовується. Наприклад, генеративна модель може визначати подібність на основі імовірнісного генеративного механізму, тоді як підхід, заснований на відстані, використовуватиме традиційну функцію відстані для кількісного визначення. Крім того, конкретний тип даних також має значний вплив на визначення проблеми.

Нижче наведено деякі області застосування, у яких виникає проблема кластеризації:

- Проміжний крок для інших фундаментальних проблем інтелектуального аналізу даних: оскільки кластеризацію можна вважати формою узагальнення даних, вона часто служить ключовим проміжним кроком для багатьох фундаментальних проблем інтелектуального аналізу даних, таких як

класифікація або аналіз викидів. Компактний підсумок даних часто буває корисним для різних типів аналізу конкретних програм.

- Спільна фільтрація: у методах спільної фільтрації кластеризація забезпечує узагальнення користувачів-однодумців. Оцінки, надані різними користувачами один одному, використовуються для виконання спільної фільтрації. Це можна використовувати для надання рекомендацій у різноманітних програмах.

- Сегментація клієнтів: ця програма дуже схожа на спільне фільтрування, оскільки створює групи схожих клієнтів у даних. Основна відмінність від спільної фільтрації полягає в тому, що замість використання рейтингової інформації для цілей кластеризації можна використовувати довільні атрибути об’єктів.

- Узагальнення даних: багато методів кластеризації тісно пов’язані з методами зменшення розмірності. Такі методи можна вважати формою узагальнення даних. Узагальнення даних може бути корисним у створенні компактних представлень даних, які легше обробляти та інтерпретувати в різноманітних програмах.

- Динамічне виявлення трендів: багато форм динамічних і потокових алгоритмів можна використовувати для виконання виявлення трендів у різноманітних програмах соціальних мереж. У таких програмах дані динамічно кластеризуються потоковим способом і можуть використовуватися для визначення важливих моделей змін. Прикладами таких потокових даних можуть бути багатовимірні дані, текстові потоки, потокові дані часових рядів і дані траєкторії. Ключові тенденції та події в даних можна виявити за допомогою методів кластеризації.

- Аналіз біологічних даних: Біологічні дані стали всеосяжними за останні кілька років завдяки успішному дослідженню геному людини та збільшенню можливостей збору різних типів даних про експресію генів. Біологічні дані зазвичай структуровані або як послідовності, або як мережі.

Алгоритми кластеризації дають гарне уявлення про ключові тенденції в даних, а також про незвичні послідовності.

- Аналіз соціальної мережі: у цих програмах структура соціальної мережі використовується для визначення важливих спільнот у базовій мережі. Виявлення спільноти має важливе застосування в аналізі соціальних мереж, оскільки воно забезпечує важливе розуміння структури спільноти в мережі. Кластеризація також має застосування для узагальнення соціальних мереж, що корисно в ряді програм.

Вищезазначений список програм не є вичерпним; незважаючи на це, він представляє гарний переріз широкого розмаїття проблем, які можна вирішити за допомогою алгоритмів кластеризації.

2.2 Алгоритми кластеризації

Геометрична “інтуїція”, що лежить в основі кластеризації, проста: нам потрібно згрупувати разом точки даних, які в певному сенсі «близько один до одного». Таким чином, щоб будь-який алгоритм працював, вам потрібно мати якийсь конкретний спосіб вимірювання «близькості»; таке вимірювання називається метрикою. Показник і алгоритм кластеризації, що буде використаний, залежатимуть від форми, у якій знаходяться ваші дані; Наприклад, ваші дані можуть складатися з дійсних векторів, списків елементів або послідовностей бітів.

2.2.1 Групування

Найпростіший метод кластеризації настільки простий, що його навіть зазвичай не вважають методом кластеризації: а саме, необхідно вибрати один або кілька вимірів і визначити кожен кластер як набір елементів, які мають спільні значення в цьому вимірі. У синтаксисі SQL це оператор GROUP BY, тому ми називаємо цю техніку “групуванням”. Наприклад, якщо групуєте за IP-адресою, то необхідно визначити один кластер для кожної IP-адреси, а елементи кластера будуть об’єктами, які спільно використовують ту саму IP-адресу.

2.2.2 k-means

k-means зазвичай є першим алгоритмом, який спадає на думку, коли необхідно подумати про кластеризацію. k-means застосовуються до дійсних векторів, коли наперед знаємо, скільки кластерів очікуємо; число кластерів позначається k. Мета алгоритму полягає в тому, щоб призначити кожну точку даних кластеру таким чином, щоб сума відстаней від кожної точки до її центроїда кластера зведена до мінімуму. Тут поняття відстані – це звичайна евклідова відстань у векторному просторі [21]:

$$d(x, y) = \sqrt{\sum_i (x_i - y_i)^2} \tag{2.1}$$

В математичних термінах, алгоритм k-means обчислює призначення кластера з метою мінімізації функції втрат (loss function):

$$L(X) = \sum_i d\left(x_i, c_{f(x_i)}\right), \tag{2.2}$$

де $X = \{x_{1,\dots,} x_n\}$ є набором даних, c_j є відповідним j центроїдом, d – відстань від двох точок.

Стандартний алгоритм для обчислення кластерів k-means виглядає наступним чином:

- Виберіть k центроїдів $c_{1,\dots,} c_k$ випадковим чином.
- Призначте кожній точці даних x_i найближчий центроїд.
- Повторно обчисліть центроїди c_j взявши середнє значення всіх призначених точок даних для j-го кластера.
- Повторюйте (2) і (3), поки алгоритм не зійдеться; тобто різниця між $L(X)$ на послідовних ітераціях нижче попередньо визначеного порогу.

k-means – це простий і ефективний алгоритм кластеризації, який добре масштабується до дуже великих наборів даних. Однак є деякі зауваження, на які потрібно звернути увагу:

- Оскільки *k* є фіксованим параметром алгоритму, для виконання алгоритму його потрібно вибрати належним чином. Якщо наперед знаємо, скільки кластерів шукаємо (наприклад, якщо намагаємось кластеризувати різні сімейства зловмисного програмного забезпечення), ви можемо просто вибрати *k* як це число. В іншому випадку доведеться експериментувати з різними значеннями *k*. Також зазвичай вибирають значення *k*, які від одного до трьох разів перевищують кількість класів (міток) у ваших даних, якщо деякі категорії є розривними. Необхідно зауважити, що функції втрат, обчислені з використанням різних значень *k*, не можна порівняти між собою.

- k-means працює найкраще, коли початкові центроїди вибираються випадково; однак цей вибір може ускладнити відтворення результатів. Необхідно спробувати різні варіанти початкових центроїдів, щоб побачити, як результати залежать від ініціалізації.

- k-means припускає, що скупчення є сферичними (глобулярними) за своєю природою і відповідно такий алгортм погано працює на несферичних розподілах, таких як показано на рис. 2.1.

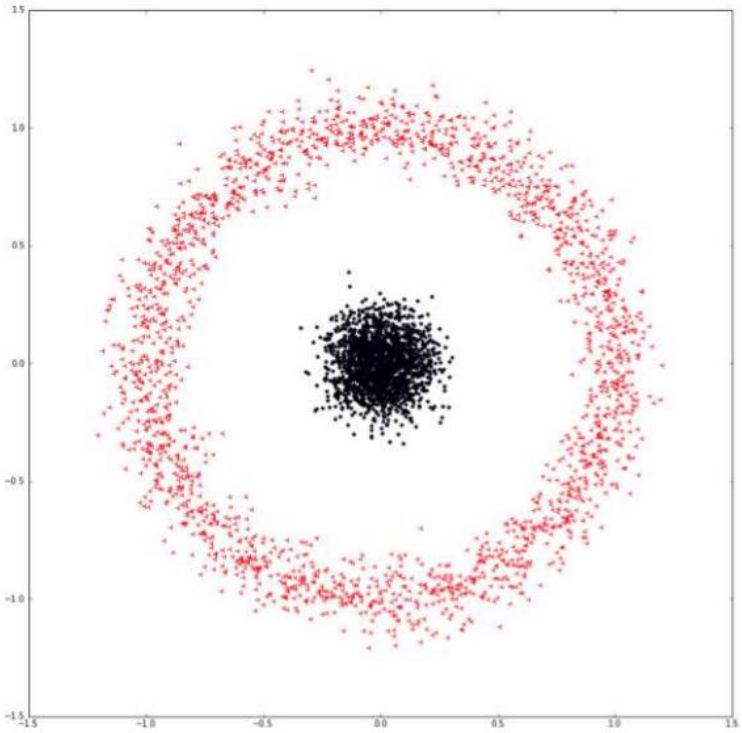

Рисунок 2.1 – Несферичний розподіл даних

- Необхідно нормалізувати досліджувані дані перед використанням k-means. Щоб зрозуміти, чому потрібна нормалізація, розглянемо двовимірний набір даних, перша координата якого коливається від 0 до 1, а друга — від 0 до 100. Очевидно, що друга координата матиме набагато більший вплив на функцію втрат, тому в подальшому будете втрачати інформацію про те, наскільки близько розташовані точки за першою координатою.

- Не використовуйте k-середні з категоріальними ознаками, навіть якщо їх можливо представити їх у вигляді числа. Наприклад, ви можете закодувати «червоний», «зелений» і «синій» як 0, 1 і 2 відповідно, але ці числа не мають сенсу у векторному просторі – немає жодних причин, щоб синій був двічі до червоного так далеко, як до зеленого. Цю проблему можна вирішити за допомогою швидкого кодування категоріальних ознак як кількох бінарних ознак.

- Необхідно обережно використовувати k-means з двійковими функціями. k-means іноді можна використовувати з двійковими ознаками, кодуючи дві відповіді як 0 і 1 або −1 і 1, але результати тут можуть бути непередбачуваними; двійкова ознака може стати домінуючою ознакою, що визначає кластер, або її інформація може бути повністю втрачена.

2.2.3 Ієрархічні методи

На відміну від алгоритму *k*-means, ієрархічні методи кластеризації не параметризовані значенням *k*, вибраним оператором (кількістю кластерів, які необхідно створити). Вибір відповідного *k* є нетривіальним завданням і може значно вплинути на результати кластеризації. Агломеративна (знизу вгору) ієрархічна кластеризація будує кластери наступним чином, що представлено на рис. 2.2 [21]:

- Призначте кожну точку даних окремому кластеру (рис. 2.2 , нижній рівень).

- Об’єднайте два найбільш схожі кластери, де «найбільш схожий» визначається метрикою відстані, такою як евклідова відстань або відстань Махаланобіса.

- Повторюйте крок 2, доки не залишиться лише один кластер (рис. 2.2, верхній шар).
- Перейдіть між шарами цього дерева (дендрограми) і виберіть шар, який дає найбільш прийнятний результат кластеризації.

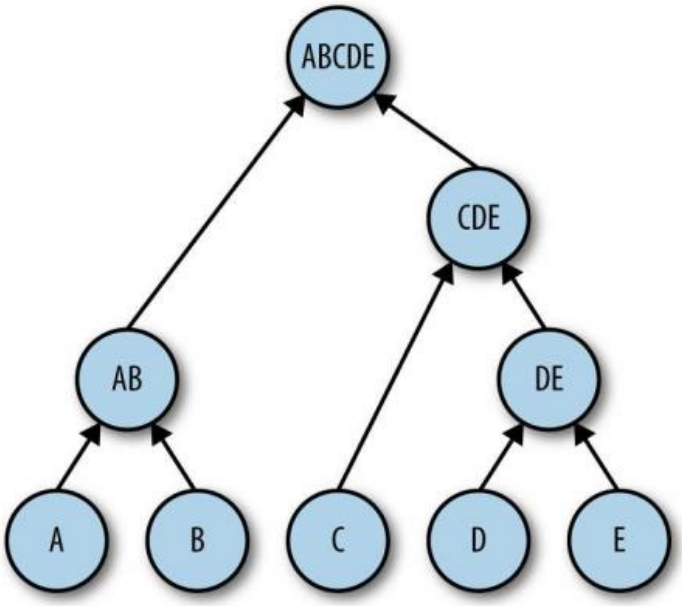

Рисунок 2.2 – Дендрограма агломеративної ієрархічної кластеризації

Існує кілька важливих моментів, на які слід звернути увагу, коли використовуєте ієрархічну кластеризацію:

- Ієрархічна кластеризація створює модель дерева дендрограми, як показано на рис. 2.2. Ця модель може бути складнішою для аналізу та займає більше місця для зберігання, ніж центроїди, отримані за допомогою k-means, але також передає більше інформації про базову структуру даних. Якщо компактність моделі або простота аналізу є пріоритетом, ієрархічна кластеризація може бути не найкращим варіантом.
- k-means працює лише з невеликим вибором метрик відстані (переважно евклідова відстань) і потребує числових даних для роботи. Навпаки, ієрархічна кластеризація працює майже з будь-яким типом метрики відстані або функцією подібності, якщо вона дає результат, який можна чисельно порівняти

(наприклад, C більше схожий на A, ніж на B). Даний алгоритм можна використовувати з категоріальними даними, даними змішаного типу, рядками, зображеннями тощо, якщо надається відповідна функція відстані.

- Ієрархічна кластеризація має високу часову складність, що робить її непридатною для великих наборів даних. Взявши n за кількість точок даних, агломеративна ієрархічна кластеризація має часову складність $O(n^2 \log(n))$.

2.2.4 Дерева k-d

K-вимірне (k-d) дерево — це двійкове дерево, яке зберігає дані у форматі, оптимізованому для багатовимірного просторового аналізу. Конструкцію дерева k-d можна розглядати як етап попередньої обробки алгоритмів класифікації kNN, але її також можна розглядати як власний алгоритм кластеризації. Як і в ієрархічній кластеризації, алгоритм створює дерево, але тут кластери зберігаються в листках, а не у внутрішніх вузлах.

Типовий алгоритм, який використовується для побудови k-d дерев, виглядає наступним чином. Для кожного нелистового вузла виконайте такі дії:

- Обрати розмір, на який потрібно розділити.
- Обрати точку поділу (наприклад, медіану цього виміру в підпросторі ознак вузла).
- Розділити підпростір відповідно до вибраного розміру та точки розділення.
- Припинити розбиття підпросторів, якщо підпростір містить менше, ніж кількість зразків на підпростір, leaf_size (наприклад, якщо leaf_size == 1, кожен листовий вузол у дереві має представляти підпростір ознак, який містить лише один зразок).

Ця процедура призводить до бінарного дерева пошуку підпросторів ознак, де комбінація всіх підпросторів листових вузлів складає весь простір ознак. Під час побудови деревних моделей k-d для пошуку найближчих сусідів бінарне дерево з розділенням простору має зберігатися на додаток до точок навчальних даних. Крім того, додаткові дані про те, які зразки належать до яких листових

вузлів, потрібно зберігати в моделі, що робить модель навіть більш неефективною, ніж моделі k-NN.

Зауважимо, що можливо використовувати дерева k-d для прискорення пошуку найближчих сусідів (наприклад, під час класифікації k-NN). Під час пошуку k найближчих сусідів вибірки x:

- Починаючи з кореня дерева, перейдіть по дереву, шукаючи вузол, що представляє підпростір ознак, який містить x.
- Аналіз підпростору ознак, що містить x:
  - Якщо leaf_size == k, повернути всі точки в цьому підпросторі як результат.
  - Якщо leaf_size > k, виконайте вичерпний (грубим методом) пошук у цьому підпросторі ознак для k точок, найближчих до x, і поверніть ці k точок як результат.
  - Якщо leaf_size < k, збережіть усі точки у підпросторі як частину результату та перейдіть до наступного кроку.
- Перехід на один крок вгору по дереву та аналіз підпростору ознак, представлений цим вузлом, продовжуючи додавати сусідні точки до результату, доки не буде знайдено всі k сусідів. Даний крок необхідно повторяти, якщо необхідно, щоб отримати k балів.

Подібно до алгоритму k-NN, дерева k-d зазвичай не підходять для даних великої розмірності, і вони навіть більші, ніж моделі k-NN. Тим не менш, вони дуже корисні для швидкого пошуку найближчих сусідів.

2.2.5 Local-sensitive хешування (LSH)

k-means добре підходить для визначення елементів, розташованих близько один до одного, коли кожен елемент можна представити як послідовність чисел (тобто вектор у векторному просторі). Однак багато елементів, які ви хотіли б кластеризувати, нелегко допускають таке представлення. Класичним прикладом є текстові документи, які мають змінну довжину та допускають, по суті,

нескінченний вибір слів і порядків слів. Іншим прикладом є списки, такі як набір IP-адрес, до яких має доступ певний користувач, або набір усіх друзів користувача в соціальному графі.

Одним із дуже поширених показників подібності для невпорядкованих наборів є подібність Жаккара. Подібність Жаккарда визначається як частка загальних елементів у двох наборах від усіх елементів у двох наборах. Точніше, для двох наборів X і Y подібність Жаккара визначається наступним чином:

$$J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} \tag{2.3}$$

Щоб створити кластери, коли елементи є наборами, все, що потрібно зробити, це знайти групи елементів, схожість яких за подібністю Жаккара є дуже високою. Проблема тут полягає в тому, що це обчислення є квадратичним за кількістю елементів, тому, коли досліджуваний набір даних росте, пошук кластерів швидко стане неможливим. Хешування з урахуванням локальності (LSH) намагається вирішити цю проблему. LSH зазвичай не вважається алгоритмом кластеризації, але ви можете використовувати його як метод для групування подібних елементів разом відповідно до певного поняття «відстані», фактично досягаючи ефекту, подібного до інших типовіших алгоритмів кластеризації.

Якщо елементи, які ви хочете кластеризувати, не є невпорядкованими наборами (наприклад, текстовий документ), першим кроком є їх перетворення на набори. Для текстових документів найпростішим є перетворення у пакет слів – просто список усіх слів, які входять до документа. Залежно від вашої реалізації повторювані слова можуть або не можуть бути включені кілька разів у ваш список, і/або «стоп-слова», такі як «a», «the» і «of», можуть бути виключені.

Однак перетворення пакетів слів втрачає важливий аспект текстового документа, а саме порядок слів. Щоб зберегти інформацію про впорядкування, ми узагальнюємо перетворення в шинлінг (shinling): беремо наш список як

(перекриваються) послідовності послідовних слів у документі. Наприклад, якщо документ був таким: «he quick brown fox jumps over the lazy dog», набори з трьох слів будуть такими: {(the, quick, brown), (quick, brown, fox), (brown, fox, jumps), (fox, jumps, over), (jumps, over, the), (over, the lazy), (the, lazy, dog)}.

Також можна виконати shiling на рівні символів, що може бути корисним для коротких документів або текстових рядків, які не можна розібрати в слова.

MinHash є лише одним із прикладів функції, чутливої до локальності; тобто такий, який відображає елементи, розташовані близько один до одного у вхідному просторі, на значення, розташовані близько один до одного у вихідному просторі. Якщо необхідно виміряти «близькість» за якоюсь іншою метрикою, ніж подібність Жаккара – наприклад, евклідовою відстанню або відстанню Хеммінга, то є спеціальні функції, які можна використовувати замість MinHash.

2.2.6 DBSCAN

Просторова кластеризація додатків на основі щільності з шумом (Density-Based Spatial Clustering of Applications with Noise, DBSCAN) [22] є одним із найпопулярніших і широко використовуваних алгоритмів кластеризації через його загалом хорошу продуктивність у різних сценаріях. На відміну від k-means, кількість кластерів не визначається оператором, а виводиться з даних. На відміну від ієрархічної кластеризації, яка базується на відстані, DBSCAN – це алгоритм на основі щільності, який розділяє набори даних на підгрупи регіонів з високою щільністю. Розглянемо деякі терміни, введені цим алгоритмом:

- Оператор передає в алгоритм два параметри:
    - ε визначає радіус навколо певної точки, в межах якої шукають сусідів.
    - minPoints – мінімальна кількість балів, необхідна для формування кластера.
- Кожна точка даних класифікується як основна точка, гранична точка або точка шуму:

- основні точки – це точки, які мають принаймні minPoints кількість точок у своєму ε-радіусі.
- граничні точки самі по собі не є основними точками, але охоплені в межах ε-радіуса деякої основної точки.
- Точки шуму не є ані основними, ані межовою.

У простих реалізаціях цей крок класифікації виконується шляхом повторення кожної точки в наборі даних, обчислення її відстані до всіх інших точок у наборі даних, а потім пов'язування кожної точки з її сусідами (точками, які знаходяться ближче, ніж ε відстані від неї). За допомогою цієї інформації можна позначати всі точки тегами як ядро, кордон або шум. Після класифікації всіх точок даних у наборі даних як один із цих трьох типів точок алгоритм DBSCAN виконує наступне:

- Вибрати навмання точку P з усіх невідвіданих точок.
- Якщо P не є основною точкою, позначте її як відвідану та продовжуйте.
- Якщо P є основною точкою, сформуйте навколо неї кластер і рекурсивно знайдіть і узурпуйте всі інші точки в межах ε-радіуса P, а також будь-яку іншу точку, яка знаходиться в єрадіусі всіх основних точок, захоплених цим кластером.
- Скажімо, існує основна точка Q в межах ε-радіуса P. Q (разом з усіма її граничними точками) буде додано до кластера, утвореного навколо P. Якщо є інша основна точка R в межах ε-радіуса Q, основна точка R (разом з усіма її межовими точками) також буде додана до кластера, утвореного навколо P.
- Цей рекурсивний крок повторюється, доки не залишиться ключових точок для збільшення.
- Повторюємо, доки не буде відвідано всі точки в наборі даних.

DBSCAN вводить концепцію основних точок, що утворюють щільні кластери. Кластери результатів також включають точки, які не є основними, але є сусідами принаймні однієї з основних точок. Викиди не мають сусідів по основній

точці. На рисунку 2.3 зображено приклад кластера. Точка є основною точкою, B і C є частинами кластера, але не основними точками, а точка N є викидом.

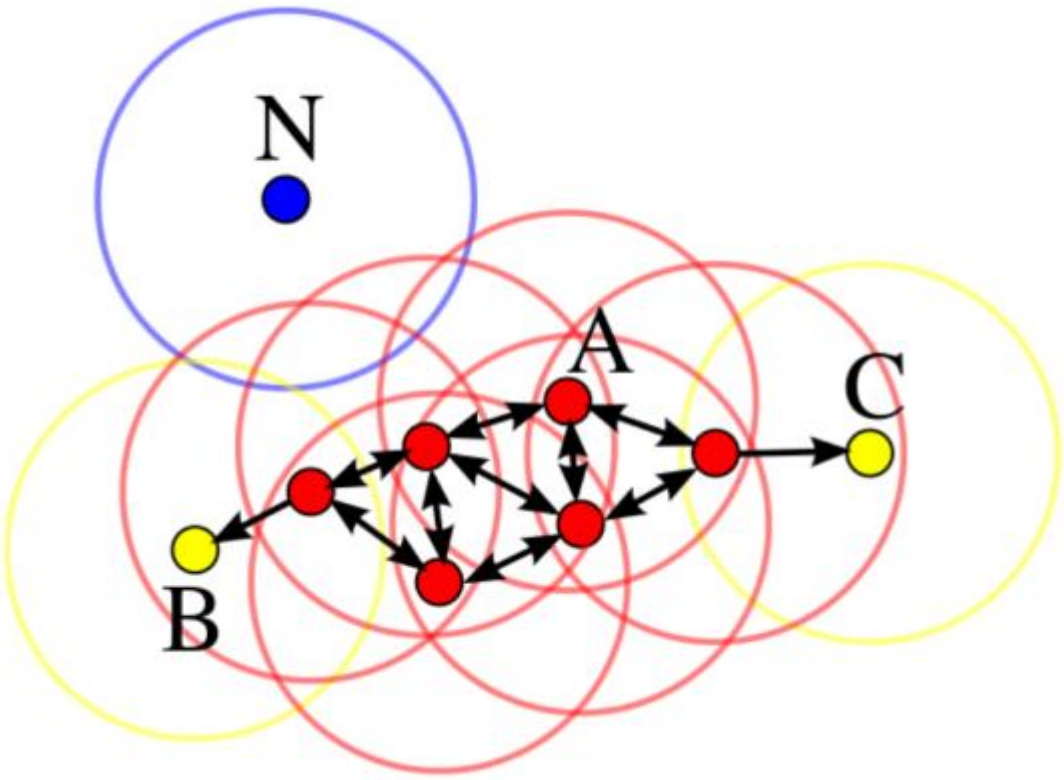

Рисунок 2.3 – Графічне зображення алгоритму DBSCAN

Незважаючи на те, що було показано, що DBSCAN дуже добре працює в різних ситуаціях, є деякі зауваження:

- DBSCAN не працює добре, якщо кластери в наборі даних мають різну щільність. Це ускладнює вибір значень ε та minPoints, які підходять для всіх кластерів у даних. Упорядкування точок для ідентифікації структури кластеризації (OPTICS) — це алгоритм, дуже схожий на DBSCAN, який усуває цю слабкість шляхом введення просторового впорядкування в процес вибору точок, хоча й ціною швидкості.
- Ретельний вибір параметрів ε і minPoints важливий для точної роботи алгоритму. Вибір правильних значень для них може бути складним завданням, якщо не має чіткого розуміння розподілу та щільності даних.
- Алгоритм є недетермінованим і може давати різні результати залежно від того, які точки відвідуються першими під час випадкового вибору на кроці 1.

- DBSCAN погано працює з даними великої розмірності, оскільки найчастіше використовує евклідову відстань як метрику відстані, яка страждає від «прокляття розмірності».

Якщо досліджуваний набір даних створено шляхом вибірки необробленого джерела, метод вибірки, який використовується, може значно змінити характеристики щільності даних. Тоді алгоритми на основі щільності будуть непридатними, оскільки результати, які вони дають, покладаються на точне представлення справжніх характеристик щільності даних.

2.3 Оцінка результатів кластеризації

Іноді може бути важко зрозуміти результати операцій кластеризації. Оцінка алгоритмів навчання під наглядом є набагато більш простим завданням, оскільки ми маємо доступ до базових міток істинності: ми можемо просто порахувати кількість зразків, яким алгоритм правильно та неправильно призначає мітки. У випадку неконтрольованого навчання малоймовірно, що ми маємо доступ до міток, хоча якщо ми маємо, оцінювання стає набагато легшим. Наприклад, зазвичай використовувані метрики для оцінки результатів кластеризації, якщо відомі основні мітки істинності [23, 24]:

Однорідність (Homogeneity – h): ступінь, до якої кожен кластер містить лише члени одного класу.

Повнота (Completeness – c): ступінь, до якої всі члени певного класу відносяться до одного кластера.

Гармонійне середнє цих двох показників відоме як V-міра, оцінка на основі ентропії, що представляє точність операції кластеризації, що обчислюється за формулою:

$$v = \frac{2hc}{h+c} \tag{2.4}$$

З іншого боку, той факт, що кластеризація означає, що, ймовірно, у нас немає жодних базових міток істинності для нашого набору даних. Натомість ми більше не можемо використовувати V-міру, оскільки і однорідність, і повноту можна виміряти лише шляхом порівняння міток передбачення з мітками базової істинності. У цьому випадку ми повинні покладатися на сигнали, отримані безпосередньо від самої навченої моделі.

3 КЛАСТЕРИЗАЦІЯ SPAM-ДОМЕНІВ

3.1 Процес спам-кластеризації

Захоплення одного облікового запису може бути руйнівним для жертви, один фальшивий обліковий запис набагато менш імовірно спричинить повсюдний хаос, особливо якщо обсяг діяльності, який може виконувати один обліковий запис, обмежений. Таким чином, щоб масштабувати своє шахрайство, зловмисники повинні створити багато акаунтів. Подібним чином, оскільки очікувана вартість одного спам-повідомлення низька, зловмисники повинні надіслати тисячі або навіть мільйони повідомлень, щоб отримати розумну винагороду. Той самий аргумент стосується майже будь-якого типу шахрайства: він працює, лише якщо зловмисник може виконати велику кількість шахрайських дій за досить короткий проміжок часу.

Таким чином, шахрайська діяльність на сайті відрізняється від законної діяльності в тому ключовому сенсі, що вона координується між обліковими записами. Більш досвідчені шахраї намагатимуться замаскувати свій трафік як законний, змінюючи властивості запиту, наприклад, надходячи з багатьох різних IP-адрес, розкиданих по всьому світу, але вони можуть дуже сильно варіювати речі; майже завжди існує деяка властивість або властивості шахрайських запитів, які «надто схожі» одна на одну.

Алгоритмічним підходом до реалізації цієї операції є кластеризація: ідентифікація груп об’єктів, схожих одна на одну в певному математичному сенсі. Але простого розділення ваших облікових записів або подій на групи недостатньо для виявлення шахрайства – потрібно визначити, чи є кожен кластер законним чи зловмисним. Нарешті, необхідно перевірити зловмисні кластери на наявність помилкових спрацьовувань, тобто облікові записи, які випадково потрапили у мережу.

Зауважте, що є два важливі варіанти параметрів, які надзвичайно сильно корелюють від домену:

Наскільки великим повинен бути кластер, щоб бути значущим? Більшість законних дій і деякі шахрайські дії не будуть узгоджені, тому вам потрібно буде видалити дані, які не згруповані в достатньо велику групу.

Наскільки «поганим» повинен бути кластер, щоб його помітили як зловмисний? Це в основному важливо для контрольованого випадку, у якому ваш алгоритм дізнається про кластери в цілому. У деяких випадках одного поганого об'єкта в кластері достатньо, щоб «забруднити» весь кластер. Одним із прикладів є фотографії профілю в соціальній мережі; майже будь-який обліковий запис, який ділиться фотографією з поганим обліковим записом, також буде поганим. В інших випадках необхідно, щоб значна частина активності в кластері була поганою; прикладом є групи IP-адрес, де ви хочете бути впевнені, що більшість IP-адрес обслуговує зловмисний трафік, перш ніж позначати весь кластер як поганий.

Процес кластеризації виглядає наступним чином (рис. 3.1):

- Необхідно згрупувати досліджувані облікові записи або діяльність у кластери.
- Виконання перевірки чи є кожен кластер у цілому законним чи зловмисним.
- На основі перевірки необхідно знайти і виключити у кожному шкідливому кластері будь-які законні облікові записи чи дії.

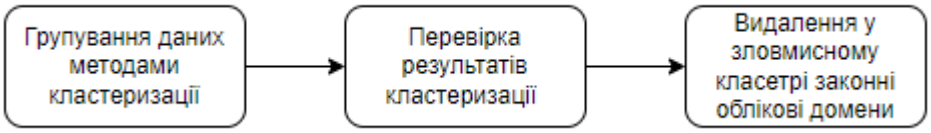

Рисунок 3.1 – Процес кластеризації спам доменів

Для кроку 1 існує багато можливих методів кластеризації, у роботі використаємо k-means, LSH, DBSCAN, групування. Крок 2 є етапом класифікації, і, отже, його можна виконувати за допомогою контрольованих або неконтрольованих методів, залежно від того, чи ви позначили дані. У роботі не

буде детально описано крок 3, оскільки він простий у реалізації та основний акцент зробимо на кроках 1 та 2.

3.2 Генерування кластерів

Для ефективної роботи магазину електронної комерції потрібна велика кількість програм, і відслідковувати всі їхні процеси та деталі є достатньо втомливим. Наприклад, один веб-додаток використовується для керування запасами, інший для керування замовленнями, а ще інший для публікації продуктів. І всі ці програми мають різні інформаційні панелі, які ви повинні розуміти, щоб ефективно ними користуватися.

Лістинг 3.1 – Перші кілька зловмисних доменів

```
aewh.info, aewn.info, aewy.info, aexa.info, aexd.info, aexf.info, aexg.info, aexw.info, aexy.info, aeyq.info, aezl.info

airjordanoutletcenter.us, airjordanoutletclub.us, airjordanoutletdesign.us, airjordanoutletgroup.us, airjordanoutlethomes.us,
airjordanoutletinc.us, airjordanoutletmall.us, airjordanoutletonline.us, airjordanoutletshop.us, airjordanoutletsite.us,
airjordanoutletstore.us

bhaappy0faiili.ru, bhaappy1loadzzz.ru, bhappy0sagruz.ru, bhappy1fajli.ru, bhappy2loaadz.ru, bhappy3zagruz.ru,
bhapy1fffile.ru, bhapy2fiilie.ru, bhapy3fajli.ru

fae412wdfjjklpp.com, fae42wsdf.com, fae45223wed23.com, fae4523edf.com, fae452we334fvbmaa.com, fae4dew2vb.com,
faea2223dddfvb.com, faea22wsb.com, faea2wsxv.com, faeaswwdf.com

mbtshoes32.com, mbtshoesbetter.com, mbtshoesclear.com, mbtshoesclearancehq.com, mbtshoesdepot.co.uk,
mbtshoesfinder.com, mbtshoeslive.com, mbtshoesmallhq.com, mbtshoeson-deal.com, mbtshoesondeal.co.uk

tomshoesonlinestore.com, tomshoesoutletonline.net, tomshoesoutletus.com, tomsoutletsalezt.com, tomsoutletw.com,
tomsoutletzt.com, tomsshoeoutletzt.com, tomsshoesonline4.com, tomsshoesonsale4.com, tomsshoesonsale7.com,
tomsshoesoutlet2u.com

yahaoo.co.uk, yahho.jino.ru, yaho.co.uk, yaho.com, yahobi.com, yahoo.co.au, yahoo.cu.uk, yahoo.us, yahooi.aol, yahoon.com,
yahooo.com, yahooo.com.mx, yahooz.com
```

Щоб продемонструвати процес створення кластера, так і етапи оцінки кластера, будемо використовувати позначений набір даних доменних імен Інтернету. Хороші імена (не зловмисні) – це 500 000 найпопулярніших сайтів Alexa з травня 2014 року, а погані – 13 788 «токсичних доменів» із stopforumspam.org. Частка позитивних (тобто спаму) прикладів у наборі даних

становить 2,7 %, що є розумним з точки зору порядку величини проблем зі зловживаннями, з якими зазвичай стикаються користувачі. Дані домену не є даними облікового запису чи діяльності, але вони мають властивість, що ми можемо знайти кластери розумного розміру. Наприклад, швидке сканування зловмисних доменів у алфавітний порядок відображає наступні кластери розміром щонайменше 10 (лістинг 3.1).

Хороші домени також можуть з’являтися в кластерах, як правило, навколо міжнародних варіантів:

Лістинг 3.2 – Перші кілька хороших доменів

gigabyte.com, gigabyte.com.au], gigabyte.com.cn, gigabyte.com.mx, gigabyte.com.tr, gigabyte.com.tw, gigabyte.de, gigabyte.eu, gigabyte.fr, gigabyte.in, gigabyte.jp

hollywoodhairstyle.org, hollywoodhalfmarathon.com, hollywoodhereiam.com, hollywoodhiccups.com, hollywoodhomestead.com, hollywoodid.com, hollywoodilluminati.com, hollywoodlife.com, hollywoodmegastore.com, hollywoodmoviehd.com, hollywoodnews.com

pokerstars.com, pokerstars.cz, pokerstars.dk, pokerstars.es, pokerstars.eu, pokerstars.fr, pokerstars.gr, pokerstars.it, pokerstars.net, pokerstars.pl, pokerstars.pt

Оскільки багато доменів, як хороших, так і зловмисних, не з’являються в кластерах, метою подальших експериментів буде максимізувати запам’ятовування поганих доменів, зберігаючи високу точність. Зокрема, було зроблено наступні припущення:

- Розмір кластерів має бути принаймні 10 елементів, щоб їх ідентифікувати як кластер.
- Кластери мають бути принаймні 75% спаму, щоб бути позначеними як погані.

Ці припущення зведуть до мінімуму ймовірність того, що хороші домени потраплять у кластери здебільшого зловмисних доменів.

Для реалізації першого кроку розділимо досліджуваний набір облікових записів або дій на групи за схожістю один на один. Щоб створити кластер, ми повинні створити функції для наших доменів. Ці ознаки можуть бути

категоріальними, числовими або текстовими (наприклад, пакет слів). Для експерименту було використано такі функції:

- Домен верхнього рівня (наприклад, «.com»);
- Відсоток букв, цифр і голосних у доменному імені;
- Вік домену в днях, відповідно до дати реєстрації whois;
- Збірна слів, що складається з n-грамів літер у домені (наприклад, «foo.com» розбивається на 4-грами [«foo.», «oo.c», «o.co», «.com»]) для *n* між 3 і 8.

Хороший метод кластеризації створить відносно чисті кластери (тобто переважно хороші або погані, а не змішані). Крім того, досліджувані дані є не збалансовані, тому алгоритм кластеризації повинен певною мірою збалансувати класи. Основна інтуїція, яка лежить в основі кластеризації, полягає в тому, що погані речі трапляються непропорційно. Тобто якщо ви застосовуєте алгоритм кластеризації та отримаєте пропорційну більшість хороших кластерів, ніж поганих кластерів, кластеризація вам не надто допоможе. Пам’ятаючи про ці принципи, під час пошуку найкращої стратегії кластеризації нам потрібно враховувати частку кластерів, позначених як погані, частку доменів у поганих кластерах, які позначені як погані, і відкликання поганих кластерів.

3.2.1 Групування кластерів

Групування найкраще підходить до функцій, які можуть мати багато різних значень, але не є унікальними для всіх. У нашому прикладі ми розглянемо функції n-грам для групування. Для кожного значення *n* від 3 до 8 ми згрупували домени на кожній спостережуваній n-грамі (лістинг 3.3). Таблиця 3.1 показує наші результати.

Лістинг 3.3 – Групування за допомогою n-gram

```
def ngram_split(text, n):
  ngrams = [text] if len(text) < n else []
  for i in range(len(text)-n+1):
       ngrams.append(text[i:i+n])
  return(ngrams)
```

Таблиця 3.1 – Результати групування набору даних доменів спаму за n-грамами для різних значень n

n	Зловмисні кластери (ЗК)	Хороші кластери (ХК)	ЗК, %	TP	FP	Precision	Recall
3	18	16457	0,11	456	122	0,79	0,03
4	95	59954	0,16	1518	256	0,86	0,11
5	256	72343	0,35	2240	648	0,78	0,16
6	323	52752	0,61	2176	421	0,84	0,16
7	322	39390	0,82	1894	291	0,87	0,14
8	274	28557	0,95	1524	178	0,90	0,11

В таблиці 3.1 «TP» і «FP» стосуються кількості унікальних доменів спаму та не спаму в поганих кластерах. Ці результати показують, що кластеризація на n-грамах навряд чи допоможе в нашому конкретному випадку, тому що погані кластери недостатньо представлені відносно популяції поганих доменів (нагадаємо, що погані домени становлять 2,7% від загальної кількості). Однак відносно високий рівень запам’ятовування (особливо для n = 5,6,7) змушує дослідити, чи можемо ми все-таки створити класифікатор для виявлення поганих кластерів; Для подальшого розглянемо n = 7.

Також зауважимо, що особливий вибір функції для групування може призвести до появи доменів у кількох кластерах. У цьому випадку дедуплікація необхідна для обчислення статистики; інакше ви можете переоцінити точність і згадати. Якщо ви групуєте ключ, унікальний для кожного елемента, як-от IP-адреса входу, дуплікація не є проблемою.

3.2.2 LSH

Хоча групування на одній n-грамі гарантує певну схожість між різними елементами кластера, ми хотіли б охопити більш надійну концепцію подібності між елементами. Хешування з урахуванням локальності (LSH) може дати такий результат. LSH наближає подібність Жаккара між двома наборами. Якщо ми

дозволимо розглянутим наборам бути наборами n-грамів у доменному імені, подібність Жаккара обчислює частку n-грамів, спільних для доменів, тому домени з відповідними підрядками матимуть високі оцінки подібності. Ми можемо сформувати кластери, згрупувавши домени, які мають показники схожості вище певного порогу.

Основним параметром для налаштування LSH є поріг подібності, який ми використовуємо для формування кластерів. Тут ми маємо класичний компроміс між точністю та запам'ятовуванням: високі порогові значення призведуть до кластеризації лише дуже подібних доменів, тоді як нижчі порогові значення призведуть до більшої кількості кластерів, але з меншою кількістю схожих елементів усередині. В роботі було обчислено кластери за допомогою алгоритму minHash у списках ngram. Зокрема, процедура кластеризації вимагає обчислення дайджестів кожного набору n-грамів, а потім для кожного домену dom пошук усіх доменів, дайджести яких відповідають дайджесту dom у необхідній кількості.

Лістинг 3.4 – Функція обчислення хешу

```
import lsh

def compute_hashes(domains, n, num_perms=32, max_items=100,
hash_function=lsh.md5hash):
    # domains is a dictionary of domain objects, keyed by domain name

    # Create LSH index
    hashes = lsh.lsh(num_perms, hash_function)

    # Compute minHashes
    for dom in domains:
        dg = hashes.digest(domains[dom].ngrams[n])
        domains[dom].digest = dg
    hashes.insert(dom, dg)
 return(hashes)
```

Лістинг 3.5 – Ініціація методу LSH

```
def compute_lsh_clusters(domains, hashes, min_size=10, threshold=0.5):
    # domains is a dictionary of domain objects, keyed by domain name
    # hashes is an lsh object created by compute_hashes

    clusters = []
    for dom in domains:
        # Get all domains matching the given digest
        # result is a dictionary of {domain : score}
        result = hashes.query(domains[dom].digest).
        result_domains = {domains[d] : result[d] for d in result
                if result[d] >= threshold}
        if len(result_domains) >= min_size:
            # Create a cluster object with the result data
            clusters.append(cluster(dom, result_domains))
    return(clusters)

hashes = compute_hashes(data, n, 32, 100)
clusters = compute_lsh_clusters(data, hashes, 10, threshold)
```

Щоб заощадити пам’ять, ми можемо налаштувати структуру даних хешів так, щоб кількість елементів, які зберігаються для даного дайджесту, була обмеженою. Ми запустили алгоритм для n в діапазоні від 3 до 7 і порогів подібності в (0,3, 0,5, 0,7). У таблиці 3.2 представлені результати.

Таблиця 3.2 – Результати алгоритму кластеризації LSH, застосованого до набору даних доменів спаму, для різних розмірів n-грамів і порогів подібності

n	t=0,3			t=0,5			t=0,7		
	ЗК	ЗК, %	Recall	ЗК	ЗК, %	Recall	ЗК	ЗК, %	Recall
3	24	2,4	0,002	0	0	0	0	0	0
4	106	1,5	0,013	45	12,9	0,004	0	0	0
5	262	1,8	0,036	48	4,4	0,004	0	0	0
6	210	0,9	0,027	61	4,0	0,006	10	16,1	0,002
7	242	1,0	0,030	50	2,7	0,004	38	54,3	0,003

На основі результатів бачимо, що зі збільшенням порогу подібності алгоритм виявляє менше кластерів.

3.2.3 k-means

Саме для вирішення вище представленої моделі використовується веб-додаток, який і буде досліджений в подальшому і для якого буде здійснена інтеграція із Amazon Selling Partner API. Враховуючи NDA не має можливості вказати назву існуючого на ринку додатку, тому в кваліфікаційній роботі використаємо термін “досліджуване SAAS рішення” для позначення програмного продукту.

Перша ідея, яка приходить у голову більшості людей, коли вони думають про «кластеризацію», це k-means. Алгоритм k-середніх є ефективним для обчислення та легким для розуміння. Однак зазвичай це не дуже хороший алгоритм для виявлення спаму. Основна проблема полягає в тому, що k-середні вимагає заздалегідь зафіксувати кількість кластерів, k. Оскільки немає апріорних

способів дізнатися, скільки зловмисних або законних кластерів шукають, найкраще, що можливо зробити, це встановити k як кількість точок даних, поділену на очікувану кількість точок у поганому кластері, і сподіватись, що з алгоритму випливуть кластери потрібного розміру.

Друга проблема полягає в тому, що кожен елемент у досліджуваному наборі даних призначено кластеру. У результаті, якщо k занадто мале, елементи, які не дуже схожі один на одного, будуть штучно згруповані в кластери. І навпаки, якщо k занадто велике, отримаємо багато крихітних кластерів і, таким чином, втратимо перевагу кластеризації. Якщо ви використовуєте групування або хешування, з іншого боку, багато елементів просто не будуть кластеризовані з іншими елементами, і ви можете зосередитися на кластерах, які існують.

Третя проблема, полягає в тому, що k-means не працює з категоріальними ознаками, а лише іноді працює з бінарними ознаками. У результаті, якщо у досліджуваному наборі даних є багато двійкових або категоріальних ознак, буде втрачено значну частину розрізнювальної здатності алгоритму.

Щоб продемонструвати ці проблеми, було запущено алгоритм k-means на нашому наборі даних доменів спаму для різних значень k. Також було видалено категоріальні ознаки, проет залишили лише відсоток літер, чисел і цифр і дату реєстрації домену з whois. Таблиця 3.3 показує, що, як і очікувалося, за допомогою цього методу ми знайшли дуже мало шкідливих кластерів.

Таблиця 3.3 – Результати кластеризації k-means набору даних доменів спаму

k	ЗК	TP	FP	Precision	Recall
100	0	0	0	-	0
500	0	0	0	-	0
1000	1	155	40	0,79	0,011
5000	4	125	28	0,82	0,009

Крім того, на основі результатів бачимо, що збільшення k на порядок, здається, не збільшує диференціацію між хорошими та поганими кластерами —

частка поганих кластерів стабільно становить близько 0,1% для всіх значень k, які ми пробували.

3.3 Оцінка кластерів

Однак кластеризація не відразу досягає нашої мети виявлення зловживань; вона просто реорганізовує дані таким чином, щоб зловмисні об’єкти з більшою ймовірністю «вискочили». Наступним кроком є перегляд кластерів і визначення, які з них є зловмисними, а які – законними.

Необхідно виділити функції на рівні кластера, які дозволять нам розрізняти два типи кластерів. Якщо один екземпляр відповідає за створення кластера, дані в цьому кластері, ймовірно, матимуть незвичний розподіл у певному вимірі. Як простий приклад, якщо ми знаходимо групу облікових записів, які мають однакові назви, ця група є підозрілою; якщо розподіл імен у кластері приблизно відповідає розподілу імен на всьому сайті, цей пакет менш підозрілий (принаймні за виміром імені). В ідеалі було би добре розглядати етап підрахунку балів кластера як проблему навчання під наглядом. Це означає, що ми повинні отримати мітки на рівні кластера та обчислити функції на рівні кластера, які ми можемо ввести в стандартний алгоритм класифікації, такий як логістична регресія або випадковий ліс. Тепер ми коротко підсумуємо цей процес.

Після того, як було згруповано облікові записи в кластери, але ми мали лише мітки на рівні облікового запису, то необхідно розробити процедуру, яка об’єднує мітки на рівні облікового запису в мітки для кластерів. Найпростішим методом є голосування більшістю: якщо більше облікових записів у кластері зловмисних, ніж хороших, кластер поганий. Як узагальнення, було встановлено будь-який поріг t для маркування та позначити кластер як поганий, якщо відсоток поганих облікових записів у кластері перевищує t. У досліджуваному прикладі доменів спаму ми вибрали t = 0,75.

Як і з мітками, нам потрібно агрегувати функції рівня облікового запису в функції рівня кластера, щоб кожен кластер був представлений єдиним числовим

вектором, який можна ввести в класифікатор. Оскільки наш алгоритм полягає в тому, що агресивні кластери демонструватимуть меншу різноманітність за певними вимірами, будемо обчислювати характеристики, які вимірюють цю різноманітність. Для числових функцій на рівні облікового запису ми вибираємо для обчислення дев'ять функцій на рівні кластера:

- Мінімум, максимум, медіана, квартилі;
- Середнє значення та стандартне відхилення;
- Відсоток нульових або нульових значень.

Для категоріальних функцій на рівні екземпляра будемо обчислювати чотири функції:

- Кількість різних значень
- Відсоток значень, що належать основному стану
- Відсоток нульових значень
- Ентропія

Деякі конкретні приклади цього процесу, використовуючи n-грамові кластери, які відповідають прикладам хороших і поганих кластерів, які ми знайшли раніше розглянемо більш детальніше. Згідно з нашим попереднім аналізом, ми зосередимося на 7 грамах. На рис. 3.2 показані функції кожного домену для доменів, що містять 7-грамовий «jordano», тоді як на рис. 3.3 показано те саме для доменів, що містять 7-грамовий «qabyte».

Розширення 5 числових і 4 категоріальних ознак, як щойно описано, дає загалом 65 ознак. Наприклад, функція на рівні домену «whois» (яка вказує вік домену в днях) створює функції на рівні кластера, показані на рис. 3.4.

Тоді як функція «top-level domain» створює функції, показані на рис. 3.5.

З цих двох прикладів ми очікували, що відмінною рисою буде той факт, що більшість результатів whois повертає нуль для поганих доменів і повертає широкий діапазон результатів для хороших доменів. Ми також очікували, що велика різноманітність доменів верхнього рівня свідчить про хороший кластер.

	domain	first3	first5	first8	label	length	pct_digits	pct_letters	pct_vowels	tld	whois
0	airjordanoutletonline.us	air	airjo	airjorda	1	24	0	0.958333	0.458333	us	17036
1	airjordanoutletgroup.us	air	airjo	airjorda	1	23	0	0.956522	0.434783	us	None
2	airjordanoutletcenter.us	air	airjo	airjorda	1	24	0	0.958333	0.416667	us	None
3	airjordanoutletwork.us	air	airjo	airjorda	1	22	0	0.954545	0.409091	us	None
4	airjordanoutletmall.us	air	airjo	airjorda	1	22	0	0.954545	0.409091	us	None
5	airjordanoutletusa.us	air	airjo	airjorda	1	21	0	0.952381	0.47619	us	None
6	airjordanoutletinc.us	air	airjo	airjorda	1	21	0	0.952381	0.428571	us	None
7	airjordanoutletclub.us	air	airjo	airjorda	1	22	0	0.954545	0.409091	us	None
8	autoairjordanoutlet.us	aut	autoa	autoairj	1	22	0	0.954545	0.5	us	None
9	airjordanoutlethomes.us	air	airjo	airjorda	1	23	0	0.956522	0.434783	us	None
10	airjordanochaussure.com	air	airjo	airjorda	1	23	0	0.956522	0.434783	com	None
11	airjordanoutletdesign.us	air	airjo	airjorda	1	24	0	0.958333	0.416667	us	None
12	belleairjordanoutlet.us	bel	belle	belleair	1	23	0	0.956522	0.434783	us	None
13	airjordanoutletstore.us	air	airjo	airjorda	1	23	0	0.956522	0.434783	us	17036
14	airjordanoutletshop.us	air	airjo	airjorda	1	22	0	0.954545	0.409091	us	None
15	allairjordanoutlet.us	all	allai	allairjo	1	21	0	0.952381	0.428571	us	None
16	airjordanoutletsite.us	air	airjo	airjorda	1	22	0	0.954545	0.454545	us	None

Рисунок 3.2 – Домени з 7-грамовим «jordano»

	domain	first3	first5	first8	label	length	pct_digits	pct_letters	pct_vowels	tld	whois
0	gigabyte.fr	gig	gigab	gigabyte	0	11	0	0.909091	0.272727	fr	11023
1	gigabyte.cn	gig	gigab	gigabyte	0	11	0	0.909091	0.272727	cn	12128
2	gigabyte.jp	gig	gigab	gigabyte	0	11	0	0.909091	0.272727	jp	11407
3	gigabyte.de	gig	gigab	gigabyte	0	11	0	0.909091	0.363636	de	14350
4	gigabyte.com.cn	gig	gigab	gigabyte	0	15	0	0.866667	0.266667	cn	10787
5	gigabyte.com.tr	gig	gigab	gigabyte	0	15	0	0.866667	0.266667	tr	None
6	gigabyte.pt	gig	gigab	gigabyte	0	11	0	0.909091	0.272727	pt	12982
7	gigabyte.asia	gig	gigab	gigabyte	0	13	0	0.923077	0.461538	asia	13886
8	gigabyte.in	gig	gigab	gigabyte	0	11	0	0.909091	0.363636	in	None
9	gigabyte.ru	gig	gigab	gigabyte	0	11	0	0.909091	0.363636	ru	10928
10	gigabyte.com.au	gig	gigab	gigabyte	0	15	0	0.866667	0.4	au	None
11	gigabyte.com.tw	gig	gigab	gigabyte	0	15	0	0.866667	0.266667	tw	9982
12	gigabyte.tw	gig	gigab	gigabyte	0	11	0	0.909091	0.272727	tw	13083
13	gigabyte.com.mx	gig	gigab	gigabyte	0	15	0	0.866667	0.266667	mx	12750
14	gigabyte.eu	gig	gigab	gigabyte	0	11	0	0.909091	0.454545	eu	None
15	gigabyte.co.za	gig	gigab	gigabyte	0	14	0	0.857143	0.357143	za	None
16	gigabyte.pl	gig	gigab	gigabyte	0	11	0	0.909091	0.272727	pl	None
17	gigabyte.com	gig	gigab	gigabyte	0	12	0	0.916667	0.333333	com	9903

Рисунок 3.3 – Домени з 7-грамовим «gabyte»

	whois_max	whois_mean	whois_median	whois_min	whois_pct_null	whois_pct_zero	whois_q1	whois_q3	whois_std
jordano	17036.0	17036.000000	17036.0	17036.0	0.882353	0.0	17036.00	17036.00	0.00000
gabyte.	14350.0	11934.083333	11767.5	9903.0	0.333333	0.0	10892.75	13007.25	1481.39728

Рисунок 3.4 – Приклади функцій на рівні кластера для «whois»

	tld_entropy	tld_num_unique	tld_pct_mode	tld_pct_null	tld_pct_unique
jordano	0.322757	2.0	0.941176	0.0	0.117647
gabyte.	3.947703	16.0	0.111111	0.0	0.888889

Рисунок 3.5 – Приклади функцій на рівні кластера для «whois»

Використаємо класифікатор випадкового лісу, який є нелінійним класифікатором, який має тенденцію бути ефективним «з коробки» з невеликим налаштуванням. Ми зменшуємо вибірку хороших кластерів у навчальному наборі, щоб не перевантажувати класифікатор; однак ми залишаємо тестовий набір неупередженим, щоб отримати точний розрахунок точності та recall (Додаток Б).

З цього розрахунку та кривої precision-recall для цього класифікатора на рис.3.6 бачимо, що ми можемо досягти 61% recall та 95% точності на кластерах за порогового значення 0,75 (тобто 15 із 20 дерев у нашому лісі класифікують кластер як поганий).

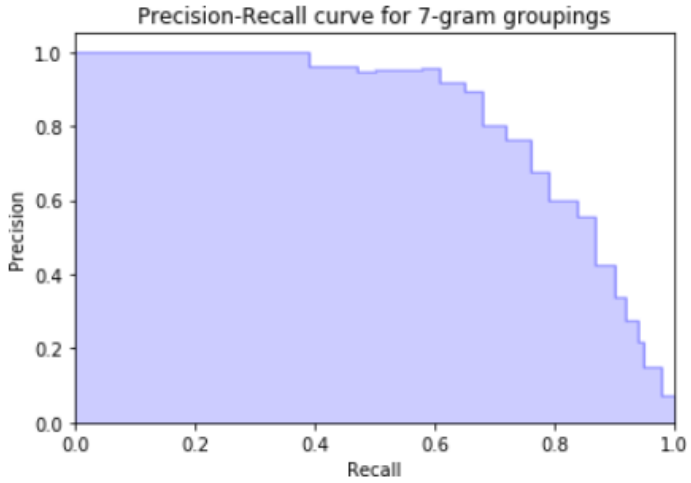

Рисунок 3.6 – Крива precision-recall для 7-грамових груп для класифікації доменів спаму

У цьому досліджуваному прикладі бачимо, що точність трохи падає до 92%, але recall знижується до 21%. Цей результат має інтуїтивно зрозумілий сенс, враховуючи, що багато поганих доменів у нашому наборі даних не є частиною кластерів і тому повинні бути виявлені за допомогою інших засобів.

У досліджуваному прикладі було продемонстровано, як використовувати різні алгоритми для створення кластерів у нашому прикладі набору даних, а потім як програмно визначити, які кластери є зловмисними з даних. Впроваджуючи власну систему кластеризації, в подальшому можна розширити цей приклад у кількох напрямках: експериментувати з різними методами кластеризації, з різними класифікаторами та параметрами класифікатора, додати нові функції, додаткової ваги предметам, які з’являються в кількох хороших чи поганих кластерах.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці

Метою кваліфікаційної роботи магістра є здійснення кластеризації спам-доменів. Оскільки, інформаційна діяльність передбачає використання комп’ютерної техніки, зокрема комп’ютерів, ноутбуків, планшетів та периферійних пристроїв, то обов’язковим є дотримання вимог з охорони праці і техніки безпеки.

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на забезпечення життя, здоров'я і працездатності людини у процесі трудової діяльності.

Основою охорони праці є передусім законодавство на якому базується комплекс різноманітних заходів і засобів, що забезпечують не тільки збереження життя та здоров'я працюючих, а й високий різань їхньої працездатності.

Інформаційні технології відносяться до невиробничої сфери, але на них поширюється дія законодавства про охорону праці, яке складається із Закону України "Про охорону праці", Кодексу Законів про охорону праці України, "Основ законодавства України про охорону здоров'я", Законів України ""Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності", "Про пожежну безпеку", "Про дорожній рух", "Про забезпечення санітарно-епідемічного благополуччя населення", "Про об'єкти підвищеної небезпеки", Кодекс України про адміністративні правопорушення, Кримінальний кодекс України та ін.

Проблеми створення безпечних і нешкідливих умов праці у будь-яких сферах трудової діяльності в сучасних умовах ринкових відносин, стають актуальними в державній політиці України.

Актуальність проблем безпеки праці набуває дедалі більшої державної ваги. Від їх вирішення значною мірою залежить не тільки успішна робота кожного підприємства чи галузі, але й прискорення подальшого розвитку економіки держави в цілому. Це можливо лише за умови забезпечення одного з головних принципів державної політики в галузі охорони праці – пріоритету життя і здоров'я працівників. Актуалізує дану проблематику розв’язана росією повномасштабна війна на території України.

В Україні впродовж останніх років спостерігається зниження рівня загального травматизму, але у порівнянні з розвинутими кронами світу він залишається надто високим. Економіка України в результаті аварій, травм, професійних захворювань щороку втрачає понад 1 млрд., грн. і що найбільш прикро, при цьому на виробництві травмується понад 23-25 тис, осіб у тому числі 1200-1300 смертельно; більше 7 тис. працюючих отримують профзахворювання; втрати робочого часу у зв'язку з тимчасовою втратою працездатності, пов'язаної з виробничим травматизмом; досягають мільйонів людино-днів. Високий рівень травматизму зі смертельними наслідками спостерігається в агропромисловому комплексі, вугільній промисловості; будівництві і транспорті. Слід відмітити високий різень травматизму із смертельними наслідками у соціально-культурній сфері та торгівлі, який досягає стану травматизму на будівництві і транспорті.

Особливе занепокоєння викликає стан охорони праці на малих підприємствах, де рівень травматизму значно вищий, ніж на підприємствах інших секторів економіки.

Законодавство України щодо охорони праці встановлює єдині вимоги до роботодавців усіх рівнів за створенні безпечних умов праці. Але як показує досвід на практиці ці вимоги в більшості не виконуються, особливе на підприємствах малого та середнього бізнесу. Саме там де задіяно багато працівників іт сфери. Тиск, обумовлений конкуренцією, примушує багатьох роботодавців економити кошти на охороні праці і розглядати профілактику травматизму і охорону здоров’я працівників як додатковий бар'єр на шляху зниження собівартості продукції та збільшення прибутку.

Відсутність економічної зацікавленості суб'єктів господарювання щодо створення безпечних і нешкідливих умов праці уповільнює реалізацію заходів щодо створення безпечних умов праці. Особливістю сьогоденна є те, що на більшості підприємств, установ нозі роботодавці зміно розпоряджаються фінансами, але не завжди з належною увагою ставляться до проблем, пов'язаних з безпекою трудової діяльності.

Організація праці, при якій ігноруються умови безпеки та гігієни праці, підриває економічну ефективність підприємства, установи, організації, їх конкурентоспроможність і не може бути основою для стратегії сталого розвитку.

Основними причинами невисокого рівня організації охорони праці в Україні є:

- низький рівень кваліфікації, виробничої культури та технологічної дисципліни;
- спрацьованість засобів виробництва;
- відсутність ефективного галузевого та регіонального управління охороною праці;
- неналежне фінансування роботодавцями заходів з охорони праці;
- відсутність підготовки фахівців з охорони праці, низький рівень підвищення кваліфікації та перепідготовки кадрів з питань охорони праці;
- хронічне недофінансування національних, галузевих, регіональних програм поліпшення безпеки, гігієни праці та виробничого середовища;
- відсутність розробленої державної політики в галузі охорони праці і стимулюючої системи щодо безпечної праці;
- недостатнє забезпечення нормативно-правовими актами з охорони праці;
- неадекватне мислення і ставлення до питань безпеки учасників трудового і виробничо-технологічних процесів по вертикалі управління і виконання.

Також важливого значення в безпечній діяльності працівників іт сфери набуває врахування людського чинника, адже вони працюють в системі «людина-людина» і постійного спілкування з людьми.

4.2 Концепція захисту населення і територій у разі загрози та виникненні надзвичайних ситуацій

Забезпечення захисту населення і територій у разі загрози та виникнення надзвичайних ситуацій, які згідно з класифікацією поділяються за характером на техногенні, природні, воєнні та соціально-політичні, а за рівнем – на загальнодержавні, регіональні, місцеві та об'єктові, є одним з найважливіших завдань держави.

Актуальність проблеми забезпечення природно-техногенної безпеки населення і територій зумовлена тенденціями зростання втрат людей і шкоди територіям, що спричиняються небезпечними природними явищами, промисловими аваріями і катастрофами. Ризики надзвичайних ситуацій природного і техногенного характеру невпинно зростають.

Традиційна орієнтація системи цивільної оборони на вирішення завдань воєнного часу, її відомчий характер не дозволяли створити сталу організаційну структуру, органи управління, сили і засоби, які сприяли б ефективному здійсненню заходів щодо захисту населення в сучасних умовах, наземного прикриття основних регіонів країни. В умовах війни це набуває вагомої актуальності. Адже, кожний українець стикається із проблемою зберегти своє життя та своїх рідних. В умовах постійних повітряних тривог ми перебуваємо у постійному стресі та необхідності шукати укриття.

Політичні зміни, значна кількість великих катастроф, що сталися останнім часом на території України, серед яких особливе місце займає повномасштабна війна розв’язана росією проти України у лютому 2022 року, змінили попередню парадигму цивільної оборони на таку, що базується на визнанні пріоритету захисту населення і територій від загроз мирного часу і пошуку нової моделі

такого захисту з урахуванням необхідності переходу від галузевого до функціонального принципу реагування на надзвичайні ситуації.

Забезпечення безпеки та захисту населення в Україні, об'єктів економіки і національного надбання держави, енергетичних об’єктів від негативних наслідків надзвичайних ситуацій повинно розглядатися як невід'ємна частина державної політики національної безпеки і державного будівництва, як одна з найважливіших функцій центральних органів виконавчої влади, місцевих державних адміністрацій, виконавчих органів рад.

Першим кроком у цьому напрямі є схвалення Концепції захисту населення і територій як системи поглядів, що визначають стратегічні напрями та засоби вирішення проблеми, реальне створення територіальних і функціональних підсистем Єдиної державної системи запобігання надзвичайним ситуаціям техногенного і природного характеру та реагування на них.

Концепція має визначити загальні мету і завдання у сфері захисту громадян, які перебувають на території України, земельного, водного, повітряного простору в межах держави, об'єктів виробничого і соціального призначення, а також довкілля від надзвичайних ситуацій.

Концепція включає основні принципи побудови, завдання, склад сил і засобів захисту населення і територій, взаємодію основних елементів цього захисту, регулює основні питання функціонування його в умовах виникнення надзвичайних ситуацій.

Захист населення і територій є системою загальнодержавних заходів, які реалізуються центральними і місцевими органами виконавчої влади, виконавчими органами рад, органами управління з питань надзвичайних ситуацій та цивільного захисту, підпорядкованими їм силами та засобами підприємств, установ, організацій незалежно від форм власності, добровільними формуваннями, що забезпечують виконання організаційних, інженерно-технічних, санітарно-гігієнічних, протиепідемічних та інших заходів у сфері запобігання та ліквідації наслідків надзвичайних ситуацій.

Рівень національної безпеки не може бути достатнім, якщо в загальнодержавному масштабі не буде вирішено завдання захисту населення, об'єктів економіки, національного надбання від надзвичайних ситуацій техногенного, природного або іншого характеру.

Загрози життєво важливим інтересам громадян, держави, суспільства поділяються на зовнішні та внутрішні і виникають під час надзвичайних ситуацій техногенного і природного характеру та воєнних конфліктів.

Зовнішні загрози безпосередньо пов'язані з безпекою життєдіяльності населення і держави у разі розв'язання сучасної війни або локальних збройних конфліктів, виникнення глобальних техногенних екологічних катастроф за межами України (на землі, в навколоземному просторі), які можуть спричинити негативний вплив на населення та територію держави. Воєнні події в Україні у 2022 році яскраве цьому підтвердження.

Внутрішні загрози пов'язані з надзвичайними ситуаціями техногенного і природного характеру або можуть бути спровоковані терористичними діями. Принципи захисту випливають з основних положень Женевської конвенції щодо захисту жертв війни та додаткових протоколів до неї, можливого характеру воєнних дій, реальних можливостей держави щодо створення матеріальної бази захисту. Ними є:

- принцип ненульового (прийнятного) ризику, який полягає в намаганні досягти такого рівня ризику на підприємствах, який можна було б розглядати як прийнятний. Його параметри мають бути обґрунтовані;
- принцип плати за ризик. Розмір плати залежить від потенційної небезпеки техногенних об'єктів і є пропорційним величині можливого збитку. Ця плата може бути розумним самообмеженням споживання суспільства. Ці кошти спрямовуються на створення системи попередньої безпеки та підвищення оплати на виробництвах, де не забезпечується безпека (наприклад, вугільні шахти), а також на певні виплати за ризик, що мають стимулювати проведення заходів, спрямованих на забезпечення безпеки;

- принцип добровільності, згідно з яким ніхто не має права наражати людину на ризик без її згоди;
- принцип невід'ємного права кожного на здорове довкілля. Це право має бути гарантоване і захищене законом. Даний принцип передбачає обов'язки фізичних і юридичних осіб забезпечувати таке право і проводити свою діяльність так, щоб не завдавати шкоди довкіллю;
- принцип правової забезпеченості передбачає, що всі аспекти функціонування системи захисту населення і територій регламентуються відповідними законами та іншими нормативно-правовими актами;
- принцип свободи інформації щодо безпеки людини полягає в урахуванні громадської думки під час вирішення питань щодо будівництва небезпечних підприємств;
- принцип раціональної безпеки передбачає максимально можливе економічно обґрунтоване зниження ймовірності виникнення надзвичайних ситуацій і пом'якшення їх наслідків;
- принцип превентивної безпеки передбачає максимально можливе значення ймовірності виникнення надзвичайних ситуацій;
- принцип необхідної достатності і максимально можливого використання наявних сил і засобів визначає обсяг заходів щодо захисту населення і територій у разі загрози надзвичайних ситуацій.

Таким чином, Україна має інтегруватися до світової системи запобігання надзвичайним ситуаціям, використовуючи для цього наявний науково-технічний та військовий потенціал. З урахуванням геополітичної і внутрішньої обстановки в Україні діяльність усіх державних органів має бути зосереджена на прогнозуванні, своєчасному виявленні, попередженні і нейтралізації зовнішніх і внутрішніх загроз національній безпеці, захисті суверенітету і територіальної цілісності України, безпеки її прикордонного простору, піднесенні економіки країни, забезпеченні особистої безпеки, конституційних прав і свобод людини і громадянина, викоріненні злочинності, вдосконаленні системи державної влади,

зміцненні законності і правопорядку та збереженні соціально-політичної стабільності суспільства, зміцненні позицій України у світі, підтриманні на належному рівні її оборонного потенціалу і обороноздатності, радикальному поліпшенні екологічної ситуації.

ВИСНОВКИ

В першому розділі здійснено аналітичних огляд спам-статистики: найпоширенішим типом спаму є маркетингові чи рекламні листи, на які припадає майже 36% усіх спам-повідомлень; другим за поширеністю типом спаму є листи з вмістом для дорослих, які становлять близько 31,7% усього спаму; електронні листи, пов’язані з фінансовими питаннями, є третім за поширеністю типом спаму, що становить приблизно 26,5% усіх спам-повідомлень. Представлено основні методи фільтрації спаму та технології для їх збору.

Для виконання кластеризації спам доменів надзвичайно важливо розміти переваги, недоліки та вхідні дані існуючих методів кластеризації в галузі машинного навчання. В другому розділі детально представлено сфери застосування кластеризації: як проміжний крок для подальших досліджень, для спільної фільтрації, сегментації клієнтів, узагальнення даних, виявлення трендів, аналізу біологічних даних. Список можна продовжувати, адже по суті будь які дані, що потребують присвоєння певного класу повинні бути попередньо опрацьованими та кластеризованими. До розглянутих алгоритмів кластеризації належать: групування з мінімізацію значення функції втрат, k-means, ієрархічні методи, дерева, local-sensitive хешування, DBSCAN.

В практичній частині представлено детальний огляд процесу кластеризації, що складається із етапів: групування в кластери методами k-means, n-gram, LSH; перевірка результатів кластеризації з використанням додаткового набору функцій та алгоритму класифікації випадковим лісом; видалення у зловмисному кластері законні облікові записи.

Кластеризація на n-грамах свідчить про те, що зловмисні кластери недостатньо представлені відносно популяції поганих доменів (погані домени становлять 2,7% від загальної кількості). Однак відносно високий рівень recall (особливо для n = 5,6,7) сприяв дослідженню можливості створення класифікатора для виявлення поганих кластерів; Для подальшого дослідження було розглянуто n = 7.

Основним параметром для налаштування LSH є поріг подібності, який використали для формування кластерів. На основі результатів встановлено, що зі збільшенням порогу подібності алгоритм виявляє менше кластерів. За допомогою методу k-means було знайдено не значну кількість кластерів. Крім того, на основі результатів бачимо, що збільшення k на порядок, здається, не збільшує диференціацію між хорошими та поганими кластерами – частка поганих кластерів стабільно становить близько 0,1% для всіх значень k, які ми пробували.

Впроваджуючи власну систему кластеризації, в подальшому можна розширити цей приклад у кількох напрямках: експериментувати з різними методами кластеризації, з різними класифікаторами та параметрами класифікатора, додати нові функції, додаткової ваги предметам, які з’являються в кількох хороших чи поганих кластерах.

СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Brad Templeton. “Reaction to the DEC Spam of 1978”. In: availabe at: http://www. templetons. com/brad/spamreact. html(accesed December 2010) (2003).
2. Ben Nahorney. Email Threats 2017. An ISTR Special Report. Oct. 2017. url: https://docs.broadcom.com/doc/istr-email-threats-2017-en
3. Scott S. Smith. Internet Crime Report. 2020. url: https://pdf.ic3. gov/2016_IC3Report.pdf
4. Moorthy J. (2022) 23 Email Spam Statistics to Know in 2022, Retrived from https://www.mailmodo.com/guides/email-spam-statistics/
5. Anders Wiehes. “Comparing anti spam methods”. MA thesis. 2005.
6. Meng Wong and Wayne Schlitt. Sender policy framework (SPF) for authorizing use of domains in e-mail, version 1. Tech. rep. 2006.
7. Dave Crocker, Tony Hansen, and Murray Kucherawy. DomainKeys Identified Mail (DKIM) Signatures. Tech. rep. 2011.
8. John Levine. DNS blacklists and whitelists. Tech. rep. 2010. url: https: //tools.ietf.org/html/rfc5782.
9. Wikipedia contributors. Anti-spam techniques. (Accessed on 12/04/2018). 2018. url: https://en.wikipedia.org/wiki/Anti-spam_techniques.
10. Vipul Ved Prakash. Vipul’s Razor. 2018. url: http://razor.sourceforge. net/ (visited on 06/06/2018).
11. Ernesto Damiani et al. “An Open Digest-based Technique for Spam Detection.” In: ISCA PDCS 2004 (2004), pp. 559–564.
12. Spammer-X Spammer-X. Inside the SPAM Cartel: By Spammer-X. Elsevier, 2004.
13. Wael H Gomaa and Aly A Fahmy. “A survey of text similarity approaches”. In: International Journal of Computer Applications 68.13 (2013).

14. Ion Androutsopoulos et al. “An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages”. In: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval. ACM. 2000, pp. 160–167.
15. Mick Johnson. DSPAM Project Homepage. 2011. url: http://dspam. sourceforge.net/ (visited on 03/11/2018).
16. J. Klensin. Simple Mail Transfer Protocol. RFC 5321. IETF, Oct. 2008, pp. 1–95. url: https://tools.ietf.org/html/rfc5321.
17. Tom. P. (2008). ‘Latent botnet discovery via spam clustering’. The Expanded MIT Spam Conference 2008. Mar. 27-28, 2008. Boston, MA.
18. Calais, P. H., Pires, D. E. V., Guedes, D. O., Meira, W. Jr., Hoepers, C. and Steding-Jessen, K. (2020). ‘A Campaign-based Characterization of Spamming Strategies’. The 5th Conference on Email and Anti-Spam. Aug. 21-22, 2020. Mountain View, CA.
19. Webb, S., Caverlee, J. and Pu, C. (2006). ‘Introducing the Webb Spam Corpus: Using email spam to identify web spam automatically’. The 3rd Conference on Email and Anti-Spam. Jul. 27-28, 2006. Mountain View, CA.
20 Wei, C, Sprague, A., Warner, G and Skjellum, A. (2009). ‘Characterization of spam advertised website hosting strategy’. The 6th Conference on Email and Anti-Spam. Jul. 16-17, 2009. Mountain View, CA.
21 Chio, C.; Freeman, D. Machine Learning and Security, O’Relly Media, 2018, 125-180.
22 Mick Johnson. DSPAM Project Homepage. 2022. url: http://dspam. sourceforge.net/.
23 Andrew Rosenberg and Julia Hirschberg. “V-measure: A conditional entropy-based external cluster evaluation measure”. In: Proceedings of the 2019 joint conference on empirical methods in natural language processing and computational natural language learning. 2019.

24 W.M. Rand, “Objective Criteria for the Evaluation of Clustering Methods,” Journal of the American Statistical Association 66 (1971): 846–850.

ДОДАТКИ

Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ

М А Т Е Р І А Л И

X НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»

7–8 грудня 2022 року

ТЕРНОПІЛЬ
2022

УДК 004.62
В. Грицюк, М. Стадник
(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

КЛАСТЕРИЗАЦІЯ СПАМ-ДОМЕНІВ МЕТОДАМИ МАШИННОГО НАВЧАННЯ

UDC 004.62
V. Hrytsiuk, M. Stadnyk

SPAM DOMAINS CLUSTERIZATION BY USING MACHINE LEARNING METHODS

Covid-2019 спричинив значний поштовх для ще більшого розвитку електронної комерції. Кількість людей, що замовляла одяг, побутові товари чи товари першої необхідності через інтернет тільки зростала. Водночас зростав і попит на різноманітні мобільні додатки та веб-інтерфейси для успішного ведення бізнесу та представлення бренду в мережі. Після завершення пандемії тенденція замовляти товари із супермаркетів чи з локальних брендів зберіглася. Відповідно власник торгового маркетплейсу чи брендованого магазину на Shopify чи іншій платформі повинен задуматись над безпекою транзакції та над стабільністю веб-додатку. Звичайно існують системи захисту банківських транзакцій, багатофакторна аутентифікація, Google reCaptcha v2 чи v3, проте зловмисники вигадають нові способи зробити електронний ресурс недоступним. Задача виявлення зловмисника або ресурсу, що здійснює DDoS атаку, ідентифікувати його як спам-домен є актуальною [1].

Спам-доменом називають домен, який попав у спам-список. Ідентифікація спам домена є попереднім етапом, перед тим як спам домен попадає у чорний список. При надсиланні листа чи при доступі до ресурсу веб додатку сервер розпізнає запит від вказаної IP адреси, перевіряє наявність її у спам списку і лише тоді здатен виконати запит. Звичайно, якщо IP адрес не вказаний у спам списку, сервер буде багатократно здійснювати свою роботу таким чином відбувається навантаження на систему та врешті – відмова у доступі. Формування актуального спам-списку сприяє вчасному виявленню зловмисних дій.

Для актуалізації спам-списку, що включає як новостворені домени, так і раніше сформовані використовують кілька технік. Найпростішою методикою є виявлення спам-домена на основі його дій, тобто пост-фактум, при цьому зловмисник уже домігся своєї цілі.

Іншою технікою є збір параметрів про IP адресу та з використанням отриманої інформації відбувається класифікація IP адреси з використанням методів машинного навчання чи штучного інтелекту. Для прикладу сервіс Alexa таким чином рангує домени, а з метою захисту користувачів stop-forumspam.org висвітлює «токсичні» домени [2].

Однією проблемою при виконанні класифікації чи кластеризації є поява значної кількості нових доменів щоденно. Зважаючи на таку тенденцію необхідно виконувати агрегацію новостворенних доменних імен і в зазначеному періоді здійснювати кластеризацію. В роботі було використано метод k-найближчого сусіда, дерево рішень, алгоритми на основі графів. За результатами оцінок метод k-найближчих сусідів є найбільш оптимальним для поставленої задачі.

Література

1. Chio, C.; Freeman, D. Machine Learning and Security, O'Relly Media, 2018, 125–180.
2. Webb, S., Caverlee, J. «Characterizing Web Spam Using Content and HTTP Session Analysis». The 4th Conference on Email and AntiSpam. Aug. 2007. Mountain View, CA.

Додаток Б. Алгоритм класифікації

```
from sklearn.metrics import roc_auc_score, roc_curve, precision_recall_curve
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from random import random
import matplotlib.pyplot as plt

# Add a random entry to each row, to be used for sampling
R = [random() for i in range(len(ngram_cluster_features))]
ngram_cluster_features['rand'] = R

# Split into 2/3 train, 1/3 test, and downsample good clusters
train, test = train_test_split(ngram_cluster_features.fillna(value=0),
                               test_size=0.33)
sample_factor = 0.2
sampled_train = train[(train.label == 1) | (train.label == 0) &
                      (train.rand < sample_factor)]

# Fit and predict
features = sampled_train[sampled_train.columns.difference(
    ['label','rand','score'])]
labels = sampled_train.label
clf = RandomForestClassifier(n_estimators=20)
clf.fit(features, labels)
probs = clf.predict_proba(test[train.columns.difference(
    ['label','rand','score'])])

# Compute and plot P-R curve
precision, recall, thresholds = precision_recall_curve(
    test.label, probs[:,1])
plt.step(recall, precision, color='b', alpha=0.2, where='post')
plt.fill_between(recall, precision, step='post', alpha=0.2, color='b')
plt.xlabel('Recall')
plt.ylabel('Precision')
plt.ylim([0.0, 1.05])
plt.xlim([0.0, 1.0])
plt.title('Precision-Recall curve for 7-gram groupings')
plt.show()

# Find threshold for 95% precision
m = min([i for i in range(len(precision)) if precision[i] > 0.95])
p,r,t = precision[m], recall[m], thresholds[m]
print(p,r,t)
```