

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСОБІВ
МОНІТОРИНГУ ЕКСТРЕМІСТСЬКОГО КОНТЕНТУ
RESEARCH AND SOFTWARE IMPLEMENTATION OF THE
TOOLS FOR MONITORING EXTREMIST CONTENT

Виконав(ла): студент(ка) 2 курсу, групи КНТ-133М
Спеціальності 121 Інженерія програмного
(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ДЕНИСЕНКО С.В.

(ПРІЗВИЩЕ та ініціали)

Керівник ЛЕОЩЕНКО С.Д.

(ПРІЗВИЩЕ та ініціали)

Рецензент СКРУПСЬКИЙ С.Ю.

(ПРІЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2024 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ДЕНИСЕНКА Сергія Володимировича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація засобів моніторингу екстремістського контенту. Research and Software Implementation of the Tools for Monitoring Extremist Content

керівник проєкту (роботи) д-р філософ., ЛЕОЩЕНКО Сергій Дмитрович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “07” жовтня 2024 року № 398

2. Строк подання студентом проєкту (роботи) 02 грудня 2024 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

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

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ЛЕОЩЕНКО С.Д., доцент		
Нормоконтроль	КАЛІНІНА М.В., асистент		

7. Дата видачі завдання “9” вересня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка структури програми.	3 тиждень	Розділ 3
5	Розробка програми.	4-7 тижні	Розділи 3,4
6	Тестування та експериментальне дослідження програмного забезпечення.	8 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	9-10 тижні	Додатки
8	Нормоконтроль та рецензування.	11 тиждень	
9	Захист роботи.	12 тиждень	

Студент(ка)

(підпис) Сергій ДЕНИСЕНКО
(Ім'я ПРІЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Сергій ЛЕОЩЕНКО
(Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
79 с., 5 табл., 19 рис., 3 дод., 28 джерел.

PYTHON, PYSPARK, TENSORFLOW, МОНІТОРИНГ,
ЕКСТРЕМІСТСЬКИЙ КОНТЕНТ, РЕКУРЕНТНА НЕЙРОННА МЕРЕЖА,
РОЗРОБКА, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

Об'єкт дослідження – процес моніторингу екстремістського контенту

Предмет дослідження – програмні засоби для моніторингу екстремістського контенту.

Метою роботи є дослідження та програмна реалізація засобів моніторингу екстремістського контенту.

Матеріали, методи та технічні засоби: мова програмування Python, середовище розробки PyCharm, надбудови PySpark та TensorFlow.

Результати. Розроблено програмне забезпечення моніторингу екстремістського контенту за допомогою мови програмування Python, середовища розробки PyCharm та надбудов PySpark та TensorFlow.

Практична цінність роботи полягає у розробці програмного забезпечення, що реалізує моніторинг екстремістського контенту.

Висновки. Виконано проєктування програмного забезпечення для моніторингу екстремістського контенту. Розроблено програмне забезпечення для моніторингу екстремістського контенту. Здійснено тестування розробленого програмного забезпечення для моніторингу екстремістського контенту.

Галузь використання – соціальні мережі, адміністратори та модератори соціальних платформ.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 79 pages, 5 tables, 19 figures, 3 appendixes, 28 sources.

PYTHON, PYSPARK, TENSORFLOW, MONITORING, EXTREMIST CONTENT, RECURRENT NEURAL NETWORK, DEVELOPMENT, SOFTWARE.

The object of research is the process of monitoring extremist content

The subject of the study is software tools for monitoring extremist content.

The aim of the work is to study and programmatically implement tools for monitoring extremist content.

Materials, methods, and technical tools: Python programming language, PyCharm development environment, PySpark and TensorFlow add-ons.

Results. Software for monitoring extremist content has been developed using the Python programming language, the PyCharm development environment, and the PySpark and TensorFlow add-ons.

The practical value of the work lies in the development of software that implements monitoring of extremist content.

Conclusions. Software for monitoring extremist content was designed. Software for monitoring extremist content has been developed. The developed software for monitoring extremist content was tested.

The scope of use is social networks, administrators and moderators of social platforms.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	7
Вступ	8
1 Аналіз предметної області	12
1.1 Екстремістський контент	12
1.2 Програмні засоби для моніторингу екстремістського контенту	15
1.2.1 Платформа ActiveFence	18
1.2.2 Сервіс Moonshot CVE	19
1.2.3 Інструмент Hatebase	20
1.2.4 Порівняння існуючих рішень	21
1.3 Постановка задачі	23
1.4 Висновки за розділом 1	24
2 Матеріали і методи	25
2.1 Вибір мови програмування	25
2.2 Використання зовнішніх бібліотек	29
2.3 Вибір середовища для розробки	30
3 Опис програми	34
3.1 Загальна схема роботи	34
3.2 Алгоритм використання RNN для моніторингу екстремізму	38
4 Експлуатація, тестування та експериментальне дослідження програми	43
4.1 Призначення й умови застосування програми	43
4.2 Стратегія тестування	45
4.3 Тестування точності роботи	47
4.4 Проблеми та виклики	49
Висновки	50
Перелік джерел посилань	52
Додаток А Технічне завдання	55
Додаток Б Фрагмент тексту програми	61
Додаток В Слайди презентації	73

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

GIFCT – The Global Internet Forum to Counter Terrorism – глобальний інтернет-форум з боротьби з тероризмом;

RNN – Recurrent Neural Network – рекурентні нейронні мережі;

ЗМІ – засоби масової інформації;

МН – машинне навчання;

ОПМ – обробка природної мови;

ПЗ – програмне забезпечення;

ШІ – штучний інтелект.

ВСТУП

Дослідження та впровадження програмного забезпечення (ПЗ) для моніторингу екстремістського контенту є актуальною темою в сучасну цифрову епоху. Поширення екстремістських ідеологій та їх поширення через цифрові платформи створюють серйозні проблеми для національної та громадської безпеки та згуртованості суспільства. Це питання особливо гостро стоїть для України, країни, що стикається з геополітичною напруженістю, триваючим військовим конфліктом і внутрішніми соціально-політичними викликами [1].

Глобальний контекст екстремістського контенту:

а) визначення та форми екстремістського контенту – екстремістський зміст включає повідомлення, засоби масової інформації (ЗМІ) або наративи, що пропагують радикальні ідеології, підбурюють до насильства або розпалюють ненависть до певних груп. Це може бути як пропаганда тероризму, так і кампанії дезінформації, спрямовані на дестабілізацію суспільства [1];

б) вплив на суспільство – екстремістський контент може поляризувати суспільство, підбурювати до насильства, радикалізувати окремих людей і підривати демократичні інститути. Він часто виходить за межі, використовуючи глобальні платформи, такі як соціальні медіа, для розширення свого охоплення [1];

в) нові технології – цифрова епоха змінила способи створення, розповсюдження та споживання екстремістського контенту:

1) соціальні медіа та платформи обміну повідомленнями: екстремістські групи використовують такі платформи, як Telegram, YouTube та TikTok;

2) штучний інтелект (ШІ) і автоматизація: екстремісти використовують інструменти ШІ для створення фальшивок,

поліпшення стратегій вербування і автоматизації поширення пропаганди;

3) зашифровані повідомлення: такі платформи, як Signal та WhatsApp, забезпечують безпечне спілкування для планування екстремістських дій.

Актуальний контекст для України:

а) геополітична нестабільність і конфлікти – Україна стикається з триваючим конфліктом у своїх східних регіонах і з 2014 року є об'єктом російської агресії. Стратегія гібридної війни, що використовується противниками, включає інформаційну війну та розповсюдження екстремістських матеріалів для дестабілізації країни [1]. Наприклад:

а) проросійські сепаратисти використовують екстремістський контент для вербування і радикалізації людей на спірних територіях;

б) пропаганда підриває суверенітет України і прагне позбавити легітимності її уряд на світовій арені;

б) внутрішні проблеми – Україна стикається з внутрішніми розбіжностями, викликаними етнічними, мовними та політичними відмінностями. Екстремістські наративи можуть використовувати ці розбіжності для поглиблення розколу в суспільстві [1]. Наприклад:

1) ультраправі або ультранаціоналістичні групи в Україні можуть брати участь в онлайн-кампаніях ненависті;

2) групи етнічних меншин або маргіналізовані спільноти можуть стати об'єктами нападів, що ще більше посилить напруженість;

в) глобальні наслідки – стратегічне положення України як кордону між Європою і Росією робить її важливою точкою тяжіння для екстремістських угруповань з більш широкими геополітичними амбіціями. Моніторинг такої діяльності не тільки приносить користь Україні, а й зміцнює регіональну безпеку [1].

Актуальність дослідження пояснюється низкою факторів, зокрема і мінливим ландшафтом загроз. Тобто, екстремістські угруповання постійно

адаптують свою тактику, використовуючи нові технології і використовуючи прогалини в цифровому управлінні [1]. Дослідження необхідні для того, щоб випереджати ці загрози шляхом:

- розуміння стратегій вербування екстремістів;
- виявлення тенденцій в поширенні контенту;
- розробка прогностичних моделей для прогнозування майбутніх загроз.

Також, слід враховувати вплив кампаній з дезінформації, адже Україна стала мов випробувальний полігон для тактики дезінформації. Необхідні дослідження, щоб:

- проаналізувати роль ботів і ферм тролів в поширенні екстремістського контенту;
- оцінити вплив неправдивих повідомлень на громадську думку;
- оцінити вразливості інформаційної екосистеми України.

Дослідження в цій галузі підтримують міжнародне співробітництво. Обмін результатами моніторингу екстремістського контенту зміцнює альянси і підвищує колективну стійкість [1].

Впровадження спеціалізованих програмних засобів може сприяти надзвичайному ефекту, зокрема програмні рішення для моніторингу екстремістського контенту відіграють ключову роль у:

- ідентифікації вмісту: алгоритми, керовані штучним інтелектом, виявляють та класифікують екстремістські матеріали, включаючи текст, зображення та відео [1];
- аналізі мережі: інструменти відображають зв'язки між обліковими записами та платформами для виявлення екстремістських мереж;
- сповіщенні в режимі реального часу: системи надають сповіщення про виникаючі загрози в режимі реального часу;
- аналізі настроїв: ШІ аналізує онлайн-настрої, щоб оцінити реакцію громадськості і виявити вогнища екстремістської активності.

З іншого боку, впровадження таких інструментів пов'язане з труднощами:

- проблеми конфіденційності даних – дотримання балансу між спостереженням і громадянськими свободами має вирішальне значення;
- точність і упередженість – необхідно навчити алгоритми, щоб уникнути помилкових спрацьовувань і упередженості;
- мовний і культурний контекст – інструменти повинні бути адаптовані до українських мовних і культурних особливостей, щоб ефективно виявляти екстремістський контент.

Україні слід приділяти пріоритетну увагу фінансуванню академічних і практичних досліджень в області моніторингу екстремістського контенту [1].

Партнерство зі світовими технологічними компаніями забезпечує доступ до передових інструментів та експертних знань.

Оновлення законів для боротьби з цифровим екстремізмом при одночасному захисті громадянських свобод має вирішальне значення.

Навчання громадян розпізнавати екстремістський контент і повідомляти про нього підвищує стійкість суспільства. Навчання ж правоохоронних органів і розвідувальних служб використанню інструментів моніторингу підвищує їх ефективність [1].

В решті решт, дослідження та впровадження програмного забезпечення для моніторингу екстремістського контенту необхідні сучасним країнам, особливо таким, як Україна, в складних умовах безпеки. Інвестуючи в цю сферу, Україна може протистояти як безпосереднім загрозам, так і довгостроковим викликам, захищаючи свій суверенітет і вносячи свій внесок у глобальні зусилля по боротьбі з екстремізмом. Це не просто технологічна чи політична проблема – це питання національної безпеки та соціальної стабільності, яке потребує нагальної уваги та скоординованих дій [1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Екстремістський контент

Екстремістський зміст відноситься до ЗМІ, інформації чи матеріалів, що порушують соціальні, правові чи культурні межі. Це часто викликає сильну реакцію, включаючи шок, обурення, страх або захоплення. Екстремістський зміст може проявлятися в різних сферах, включаючи цифрові медіа, розваги, політичний дискурс та художнє вираження, і може включати насильницькі, відверті, графічні або ідеологічно радикальні матеріали [2].

Розуміння екстремістського змісту вимагає аналізу його визначень, причин, контекстів, впливів та нормативно-правової бази, що регулює його поширення [2].

Екстремальний контент в цілому можна розділити на три категорії:

- відвертий або графічний контент: включає насильницькі, сексуальні або графічні зображення, що виходять за рамки нормативних вимог. Приклади: відеоролики із зображенням насильства, матеріали відвертого сексуального характеру або шокуючі зображення [2];

- ідеологічно екстремальний контент: включає радикальні політичні, соціальні чи релігійні погляди, які закликають до суттєвого відхилення від загальноприйнятих норм або розпалюють ненависть, розкол чи насильство. Приклади: терористична пропаганда, що розпалюють ненависть висловлювання або конспірологічні сюжети [2];

- шокуючий або розсуває межі контент: матеріали, які кидають виклик суспільним табу або нормам заради шокуючого ефекту або художніх експериментів. Приклади: вистави, що зображують страждання, шокуюча реклама або суперечливі фільми [2].

Поширення екстремістського контенту відбувається з багатьох джерел. По-перше, це звісно культурні та мистецькі матеріали. Художники, кінематографісти та творці можуть торкатися заборонених тем, щоб викликати роздуми або критикувати суспільні норми. Екстремальний вміст у мистецтві,

такий як твори Ларса фон Трієра чи Марини Абрамович, часто порушує зони комфорту глядачів [3].

Другим джерелом може стати цифрова демократизація. Платформи соціальних медіа та Інтернет дозволяють будь-якому користувачеві ділитися вмістом без традиційного контролю, що призводить до поширення екстремальних матеріалів. Алгоритми можуть посилювати шокуючий або провокаційний контент через його високу залученість [2].

Третє джерело пов'язано з особистими – психологічними мотивами. Захоплення людей нездоровим або шокуючим призводить до споживання і створення екстремального контенту. Масова культура може використовувати це для створення сенсаційних заголовків та мініатюр, щоб привернути увагу [2].

Окремо слід розглядати ідеологічні цілі – екстремістські групи, які використовують засоби масової інформації для поширення пропаганди, вербування членів або навіювання страху. Прикладами можуть служити терористичні організації, що поширюють відеозаписи актів насильства, або групи ненависті, що пропагують ідеї, що сіють розкол [2].

Одночасно мотиватором та джерелом такого контенту також можуть стати: прибуток і публічність – творці контенту або ЗМІ можуть виходити за рамки дозволеного, щоб домогтися популярності і фінансової винагороди. Суперечки часто призводять до збільшення переглядів, дискусій та доходів від реклами [3].

Серед основних, та найбільш шокуючих видів та прикладів екстремістського контенту відрізняють:

- насильство та кров – графічні зображення фізичної шкоди або смерті; зустрічається у фільмах, відеоіграх або у відеороликах, що просочилися в соціальні мережі [2];

- відвертий сексуальний підтекст – порнографія, фетиш-контент або відверті зображення в мистецтві та ЗМІ; наприклад, низка серіалів зазнали критики за зображення сексуального насильства [2];

- радикальні ідеології – екстремістський контент пропагує ненависть, насильство або радикалізацію; прикладами можуть служити пропагандистські відеоролики або маніфести прихильників переваги білої раси [4];

- зухвалий художній вираз – мистецтво, спрямоване на те, щоб викликати дискомфорт, як, наприклад, шокуючі кампанії неурядових організацій [4];

- суперечливий гумор – образливі жарти або меми, що виходять за рамки допустимого дискурсу; як приклади можна навести меми з чорним гумором, які спрощують делікатні питання [4];

- фейки та синтетичні медіа – маніпульовані відео або зображення, на яких зображені сфабриковані, часто шокуючі сценарії; підроблена порнографія або сфабриковані політичні промови створюють моральні та юридичні дилеми [5].

Серед найбільших наслідки від розповсюдження екстремістського контенту виділяють:

- соціальні та психологічні наслідки – вплив насильницького або графічного контенту може викликати травму, зниження чутливості або страх. Радикальні ідеології можуть поляризувати суспільство, сприяючи поділу та ворожості [6];

- нормалізація заборонених тим – постійний вплив може нормалізувати екстремістську поведінку або погляди, впливаючи на суспільні стандарти;

- культурний та художній вплив – екстремістський контент може викликати критичні дискусії і розширити межі художнього самовираження. Однак він також може відштовхнути аудиторію і спровокувати негативну реакцію;

- наслідки для цифрової екосистеми – платформи стикаються з труднощами при модерації контенту, забезпечуючи баланс між свободою самовираження. Екстремістський вірусний вміст може завдати шкоди репутації платформи та довірі користувачів [6].

З метою регулювання проблеми використовують різні підходи. Зокрема, модерація контенту на різних соціальних платформах. Такі платформи, як Facebook, YouTube та TikTok, використовують автоматизовані інструменти та модераторів людей для виявлення та видалення екстремістського вмісту [5].

Однак, такий підхід до вирішення проблеми включає балансування між цензурою та свободою слова, управління помилковими спрацьовуваннями та подолання глобальних культурних відмінностей [3].

Також, у багатьох країнах криміналізуються певні види екстремістського контенту, такі як дитяча порнографія, підбурювання до насильства або розпалювання ненависті. Але закони сильно розрізняються, що ускладнює правозастосування в глобальному цифровому просторі [5].

Навмисна цензура екстремістського контенту може придушувати креативність і голоси незгодних. Дискусія про те, чи слід захищати суспільні норми і зберігати свободу творчості, триває [6].

Екстремістський контент залишається суперечливою та багатогранною проблемою. Хоча він кидає виклик суспільним нормам та мистецьким кордонам, він також становить значний ризик для психічного здоров'я, соціальної згуртованості та безпеки. Баланс між правом на свободу вираження поглядів та необхідністю захисту людей від шкоди вимагає тонкого розуміння та спільних рішень. У міру розвитку технологій суспільство повинно залишатися пильним у вирішенні складних проблем, пов'язаних з екстремальним контентом, гарантуючи, що цифрова епоха сприяє прогресу, а не шкодить йому.

1.2 Програмні засоби для моніторингу екстремістського контенту

Моніторинг екстремістського контенту став найважливішим напрямком діяльності урядів, приватних організацій і технологічних компаній, які прагнуть пом'якшити його вплив. Були розроблені різні інструменти та

програми для виявлення, відстеження та управління екстремістським вмістом на цифрових платформах. Ці інструменти використовують такі технології, як ШІ, машинне навчання (МН), обробка природної мови (ОПМ) та модерація людини [7].

Усі існуючі інструменти та програми для моніторингу екстремістського контенту можна розділити на певні групи за їхнім рівнем.

Глобальний Інтернет-форум з боротьби з тероризмом (The Global Internet Forum to Counter Terrorism – GIFCT) – це галузева ініціатива, заснована Facebook, Microsoft, Twitter та YouTube. Вона спрямована на запобігання використанню цифрових платформ терористами і воєнними екстремістами. GIFCT підтримує базу даних хешованого терористичного вмісту (зображень та відео), що дозволяє компаніям-членам швидко виявляти та видаляти відомі екстремістські матеріали. Сприяє співпраці між урядами, науковими колами та групами громадянського суспільства у боротьбі з онлайн-екстремізмом. Забезпечує скоординоване реагування на такі події, як терористичні атаки, з метою запобігання поширенню екстремістської пропаганди [7].

Платформи для модерації контенту. Від так, наприклад Thorn – для платформ соціальних мереж. Виявляти і видаляти інформацію, пов'язану з експлуатацією дітей і екстремістським контентом. Фокусується на виявленні нелегального контенту і захисту вразливих осіб. Інший приклад, ActiveFence – інструмент, який використовує ШІ та людський аналіз для виявлення шкідливого контенту і управління ним. Виявляє екстремістські матеріали, що розпалюють ненависть висловлювання, дезінформацію та інші порушення. Інтегрується з робочими процесами організації для ефективної модерації контенту [8].

Пошукові роботи та автоматизовані засоби виявлення. Наприклад, Moonshot CVE – протидія насильницькому екстремізму за допомогою технологій та досліджень. Виявляє осіб, що піддаються ризику, і перенаправляє їх на альтернативні шляхи. Використовує цільову рекламу для

перехоплення осіб, які шукають екстремістські матеріали. Використовує інструменти штучного інтелекту для моніторингу форумів, платформ соціальних медіа та активності в темному Інтернеті. Функція NetClean – ідентифікує і позначає незаконний контент, зокрема, матеріали, пов'язані з експлуатацією дітей, і екстремістські матеріали. Використовується правоохоронними органами та приватними компаніями для підтримки чистоти мереж [7].

Інструменти компанії для соціальних мереж. В кожній подібній компанії свої механізми, тому слід розглянути їх окремо:

- Meta (Facebook/Instagram) – використовує ШІ та МН для виявлення і видалення екстремістських матеріалів, що розпалюють ненависть висловлювань і контенту, пов'язаного з тероризмом. Також використовує модераторів-людей для ретельної перевірки контенту [8];

- ідентифікатор вмісту YouTube – система, яка перевіряє завантажені відео на наявність у базі даних вмісту, захищеного авторським правом та позначеного як екстремістський. Автоматично видаляє або демонетизує матеріали, що порушують авторські права [7];

- інструменти політики Twitter щодо поведінки, що викликає ненависть – відстежує і припиняє розпалювання ненависті, агресивну риторику і екстремістську пропаганду. Використовує автоматизовані алгоритми в поєднанні з модерацією людиною [7].

Також окремо можна розглянути інструменти урядів та правоохоронних органів:

- проєкт боротьби з екстремізмом – неурядова організація, яка відстежує екстремістський контент по всьому світу. Працює з технологічними компаніями над удосконаленням інструментів штучного інтелекту для виявлення екстремістських матеріалів [8];

- програма PREVENT (Великобританія) – націлена на людей, схильних до ризику радикалізації. Використовує аналіз даних та моніторинг соціальних мереж для раннього виявлення та втручання [8];

– платформа для аналізу терористичного вмісту – створено компанією Tech Against Terrorism. Збирає та аналізує терористичний вміст з відкритим кодом, особливо на невеликих платформах [8].

Моніторинг екстремістського контенту вимагає комплексного підходу, який об'єднує технології, людський нагляд і глобальне співробітництво. Такі інструменти, як GIFCT, Moonshot CVE та платформи на ШІ, відіграють ключову роль у цій боротьбі, але для врахування мінливої тактики екстремістських угруповань необхідна постійна адаптація. Забезпечення етичного використання цих інструментів при одночасному захисті прав користувачів залишається серйозною проблемою в цифрову епоху.

1.2.1 Платформа ActiveFence

ActiveFence – це комплексна платформа, яка використовує штучний інтелект та людський досвід для виявлення та управління шкідливим та екстремістським контентом в Інтернеті. Вона відстежує широкий спектр платформ, включаючи основні соціальні мережі, спеціалізовані форуми та темну павутину [9].

Серед ключових функцій виділяють:

- виявлення та відстеження екстремістської пропаганди, спроби вербування і розпалювання ненависті;
- робота кількома мовами та в різних культурних контекстах;
- звітність про дезінформацію, теорії змови та організовані групи ненависті.

В основному використовується урядами, компаніями соціальних мереж та приватними організаціями для забезпечення політики щодо вмісту та захисту користувачів.

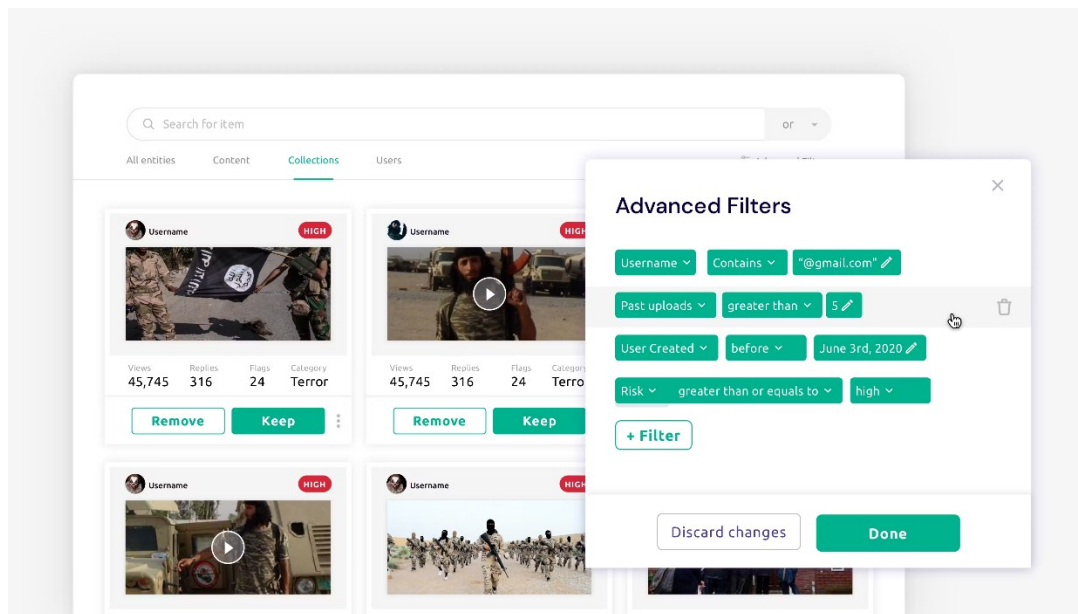


Рисунок 1.1 – Приклад звіту в ActiveFence [10]

1.2.2 Сервіс Moonshot CVE

Moonshot CVE – спеціалізується на протидії насильницькому екстремізму, відстежуючи осіб, які перебувають у групі ризику, і перенаправляючи їх на позитивні розповіді або послуги підтримки. Для втручання в критичні моменти компанія використовує аналіз даних та інформацію про поведінку [11].

Серед ключових функцій виділяють:

- відстеження пошукових запитів й активностей в соціальних мережах, що вказують на радикалізацію;
- проведення цільових рекламних кампаній, щоб надати альтернативні ресурси користувачам, які шукають екстремістський контент;
- перенаправлення користувачів, а не просто блокуванню вмісту.

Широко використовується урядами та організаціями, що займаються дерадикалізацією та боротьбою з тероризмом.

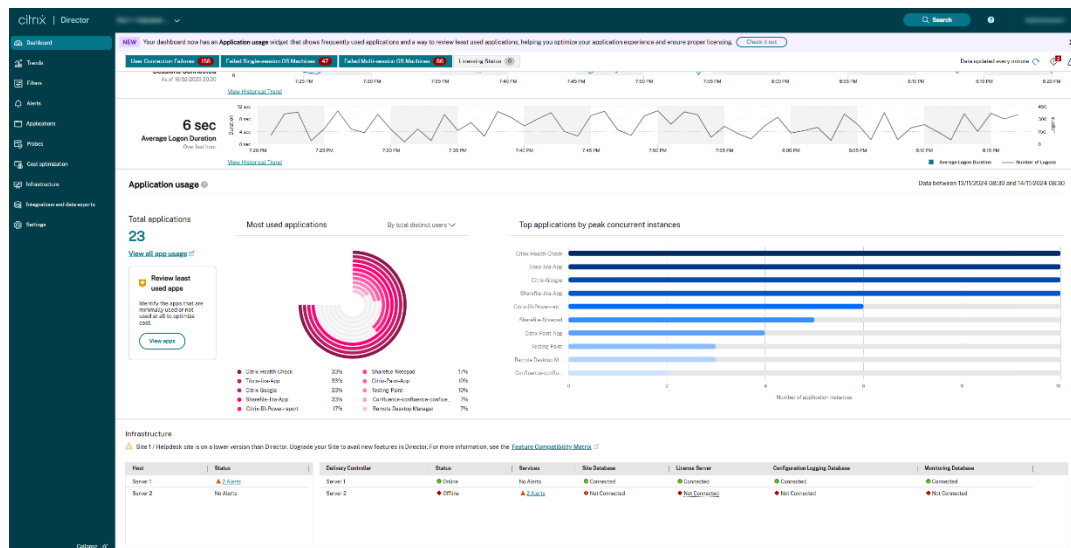


Рисунок 1.2 – Робочий простір Moonshot CVE [11]

1.2.3 Інструмент Hatebase

Hatebase – це краудсорсингова база даних та інструмент моніторингу, який відстежує мову ненависті та екстремістські висловлювання в Інтернеті. Він підтримує аналіз розмов на різних платформах у режимі реального часу для виявлення потенційно небезпечної риторики [12].

Серед ключових функцій виділяють:

- багатомовна база даних ненависницьких термінів і фраз.
- API для інтеграції з системами модерації та аналізу вмісту.
- відстеження тенденції ненависті в Інтернеті, щоб передбачити ескалацію та насильство.

Використовується дослідниками та технологічними компаніями для аналізу мови ненависті та забезпечення дотримання політики [13].

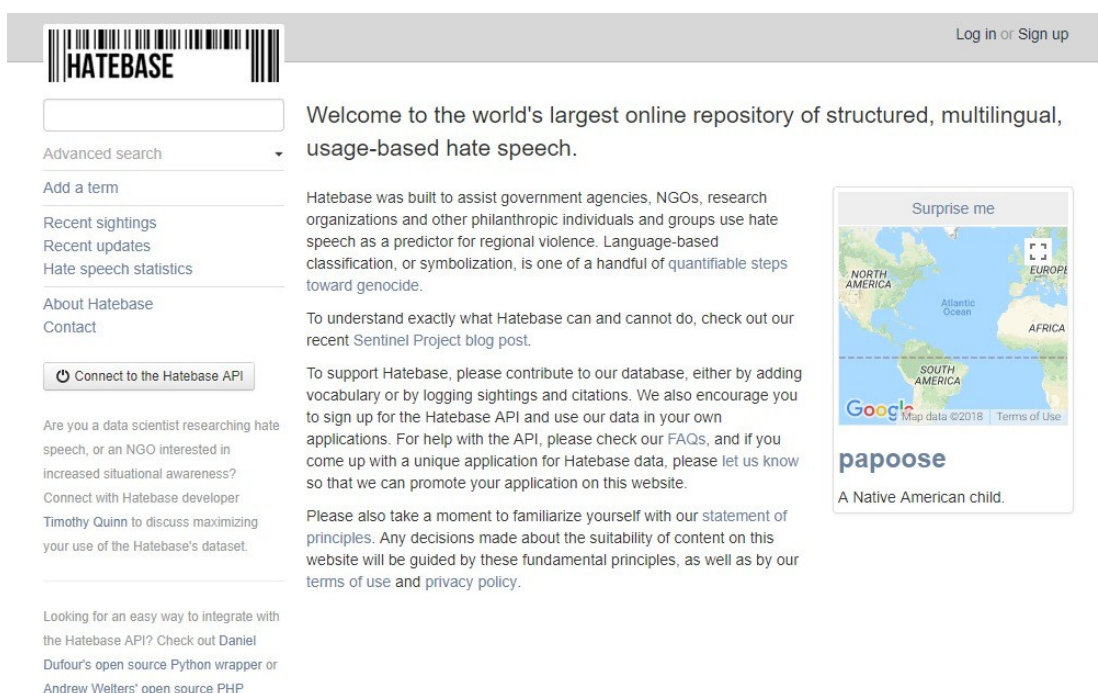


Рисунок 1.3 – Головна сторінка Hatebase [12]

1.2.4 Порівняння існуючих рішень

Порівняння ПЗ для моніторингу екстремістського контенту у вигляді таблиці наведено у таблиці 1.1.

ActiveFence – це універсальне та масштабоване рішення для моніторингу екстремістського та шкідливого вмісту на різних типах медіа та платформах.

Moonshot CVE спеціалізується на поведінковій аналітиці та дерадикалізації за допомогою активного втручання, приділяючи більше уваги окремим особам, ніж моніторингу вмісту.

Hatebase є чудовим у лінгвістичному аналізі, пропонуючи дослідникам та установам ресурс для відстеження моделей та тенденцій ненависті у всьому світі.

Кожен інструмент найкраще підходить для конкретних сценаріїв залежно від цілей (широкий моніторинг, втручання або дослідження).

Таблиця 1.1 – Порівняння методів для пошуку спільнот користувачів у соціальних мережах

	ActiveFence	Moonshot CVE	Hatebase
Основна увага	Широкий моніторинг екстремістського контенту, дезінформації, висловлювань ненависті та шкідливих матеріалів як на державних, так і на приватних платформах.	Поведінкове втручання та дерадикалізація за допомогою цільової реклами та альтернативних наративів.	Виявлення та аналіз висловлювань, що розпалюють ненависть, з акцентом на лінгвістичні закономірності та багатомовний вміст.
Використовувана технологія	ШІ, обробка природної мови, людський аналіз та кросплатформна інтеграція.	Поведінкова аналітика, моніторинг пошукових даних та цільові рекламні кампанії для протидії поширенню повідомлень.	Краудсорсингова база даних термінів ненависті, API для лінгвістичного аналізу та моніторингу в режимі реального часу.
Обсяг контенту	– платформи соціальних мереж; – спеціалізовані форуми; – мультиформатний контент (текст, зображення, відео)	– запити в пошукових системах; – поведінка в соціальних мережах; – орієнтація на показники радикалізації	– текстові вирази ненависті – багатомовні бази даних для виявлення риторики ненависті
Сильні сторони	– масштабоване виявлення, кероване ШІ; – можливість багатомовності; – інтеграція з робочими процесами організації	– проактивна дерадикалізація; – ефективний для перенаправлення вразливих осіб до ресурсів; – поведінкове втручання	– велика база даних термінів ненависті; – мовний моніторинг в режимі реального часу; – відмінна дослідницька цінність
Обмеження	Висока залежність від ШІ може призвести до помилкових спрацьовувань	Обмежений у моніторингу візуального або нетекстового контенту	Більш вузька спрямованість (тільки розпалювання ненависті)
Користувач	Уряди, технологічні компанії, приватні організації та правоохоронні органи.	Уряди та організації зосередили свою увагу на дерадикалізації та боротьбі з екстремізмом.	Дослідники, компанії соціальних медіа та навчальні заклади.

Ці інструменти являють собою спектр підходів до боротьби з екстремістським контентом, від попереджувального виявлення і видалення (ActiveFence) до поведінкового втручання (Moonshot CVE) і лінгвістичного аналізу (Hatebase). Кожен з них відіграє унікальну роль у підтримці безпеки онлайн-середовища.

Саме тому актуальною задачею є розробка нових засобів моніторингу екстремістського контенту.

1.3 Постановка задачі

Метою роботи є розробка та дослідження нових засобів моніторингу екстремістського контенту.

Під час роботи буде розроблено новий засіб моніторингу екстремістського контенту на основі використання рекурентних нейронних мереж (recurrent neural network – RNN) для:

- аналізу тексту – виявлення висловлювань, що розпалюють ненависть, та екстремістської лексики;
- аналізу мовлення – аналіз аудіо на наявність шкідливої риторики;
- моніторингу відео – виявлення екстремістського контенту в підписах або аудіо.

Розроблення нового засобу моніторингу екстремістського контенту дозволить:

- виявляти та вирішувати конфліктні ситуації в режимі реального часу;
- точніше та детальніше розуміти контент;
- автоматизувати та масштабувати моніторинг на різних платформах.

Нові рішення покращать моніторинг екстремістського вмісту шляхом аналізу послідовних даних, забезпечуючи точні, масштабовані та проактивні системи виявлення.

1.4 Висновки за розділом 1

В ході виконання першого розділу було досліджено проблемну область екстремістського контенту.

Додатково було проаналізовано найпопулярніші програмні рішення. На основі їхнього порівняння було прийнято рішення про розробу нового більш універсального рішення.

Основую для нового засобу було обрано RNN.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

Python став популярною мовою програмування для програм, пов'язаних з наукою про дані, завдяки своїй універсальності, простоті використання та Великій екосистемі інструментів, призначених для аналізу даних, візуалізації та машинного навчання. Цей документ розглядає причини, чому Python є кращою мовою для роботи з даними, і дає уявлення про його функції, бібліотеки та підтримку спільноти [14].

Простий та інтуїтивно зрозумілий синтаксис Python робить його ідеальною мовою як для початківців, так і для експертів. Фахівці з обробки даних можуть приділяти більше уваги вирішенню проблем з даними, а не розбиратися зі складнощами самої мови програмування [14].

Скорочений шаблонний код Python мінімізує обсяг шаблонного коду, дозволяючи фахівцям з обробки даних виконувати завдання з меншою кількістю рядків коду [14].

Швидке прототипування: Python забезпечує швидку розробку та тестування, що необхідно для експериментів з даними [14].

Python пропонує багату екосистему бібліотек, спеціально розроблених для обробки даних, аналізу, візуалізації та машинного навчання. Ці бібліотеки значно спрощують складні робочі процеси в галузі науки про дані [14].

Pandas: надає потужні інструменти для обробки та аналізу даних, включаючи підтримку фреймворків даних [15].

NumPy: забезпечує ефективні чисельні обчислення та маніпулювання великими наборами даних [15].

Matplotlib: основна бібліотека для створення статичних, анімованих та інтерактивних візуалізацій [15].

Seaborn: створений на базі Matplotlib, що спрощує створення естетично привабливих і складних графіків [15].

Plotly: пропонує інтерактивні візуалізації, ідеально підходять для інформаційних панелей та вебперегляду даних [15].

Scikit-learn: надає інструменти для стандартних завдань машинного навчання, таких як класифікація, регресія та кластеризація [15].

TensorFlow і PyTorch: розроблені для додатків глибокого навчання, ці бібліотеки підтримують складні моделі нейронних мереж [15].

XGBoost: Оптимізовано для методів підвищення градієнта, які часто використовуються в таких змаганнях, як Kaggle [15].

Універсальність Python дозволяє йому легко інтегруватися з різними сферами та технологіями.

Широке поширення Python сприяло формуванню активної спільноти, що надає великі ресурси та підтримку. Відкритий код: Python та його бібліотеки доступні для безкоштовного використання та активного супроводу. Форуми та навчальні посібники: такі платформи, як Stack Overflow, GitHub та спеціалізовані форуми Python, пропонують рішення поширених проблем. Постійний розвиток: регулярні оновлення гарантують, що Python залишається актуальним та ефективним [16].

Хмарна інтеграція: Python легко взаємодіє з AWS, Google Cloud та Azure для розгортання моделей машинного навчання. Автоматизація: такі інструменти, як Airflow та Luigi, дозволяють автоматизувати завдання та керувати конвеєром даних. Передові технології: Python підтримує розробку передових технологій штучного інтелекту, забезпечуючи можливість запуску моделей на пристроях Інтернету речей [16].

Кросплатформність Python забезпечує його послідовну роботу в різних операційних системах, таких як Windows, macOS та Linux, що робить його придатним для різних обчислювальних середовищ [16].

Jupyter Notebooks: дозволяє науковцям даних об'єднувати код, візуалізацію та описовий текст в одному документі [16].

Google Colab: надає хмарну платформу для запуску коду на Python із вбудованою підтримкою графічних процесорів [16].

Python чудовий у різних галузях науки про дані. Дослідницький аналіз даних: такі інструменти, як Pandas та Seaborn, полегшують аналіз та візуалізацію наборів даних. Прогностичне моделювання: такі бібліотеки, як Scikit-learn та XGBoost, допомагають створювати надійні прогностичні моделі. Обробка мовлення: NLTK і spaCy спрощують завдання обробки тексту, такі як аналіз настроїв [16].

Глибоке навчання: TensorFlow і PyTorch використовують складні моделі нейронних мереж для таких додатків, як розпізнавання зображень. Аналіз часових рядів: бібліотеки, такі як Statsmodels, надають надійні інструменти для аналізу даних, що залежать від часу [16].

Простота Python, його великі бібліотеки та активна спільнота зробили його наріжним каменем додатків для обробки даних. Його здатність справлятися з широким спектром завдань, від попередньої обробки даних до просунутого машинного навчання, робить його незамінним для сучасних фахівців з обробки даних. Завдяки постійному розвитку екосистеми, Python продовжує визначати майбутнє науки про дані, гарантуючи, що він залишається кращим вибором як для початківців, так і для досвідчених професіоналів [16].

Обґрунтування вибору мови програмування для розробка та дослідження нових засобів моніторингу екстремістського контенту наведено у таблиці 2.1.

Підбиваючи певні підсумки порівняння, можна зробити наступні висновки: Python – найкраще підходить для простоти використання, створення прототипів та використання великих бібліотек для роботи з даними. Ідеально підходить для завдань дослідження даних і машинного навчання. Java – підходить для додатків корпоративного масштабу, що вимагають високої продуктивності, хоча спочатку і не орієнтований на роботу з даними. Scala – оптимізовано для роботи з великими обсягами даних і Apache Spark; ідеально підходить для ентузіастів розподіленої обробки даних і функціонального програмування.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки методів реконструкції тривимірних зображень

Критерій порівняння мов програмування	Мова програмування		
	Python	Java	Scala
Простота навчання	Простий синтаксис, зрозумілий для початківців.	Більш висока швидкість навчання, детальний синтаксис.	Помірна складність; лаконічний, але вимагає знайомства з функціональним програмуванням.
Екосистема та бібліотеки	Багаті бібліотеки для науки про дані: Pandas, NumPy, TensorFlow, Scikit-learn.	Обмежена кількість власних бібліотек для обробки даних; потрібна інтеграція із зовнішніми інструментами (наприклад, Weka).	Потужна підтримка інструментів обробки великих даних: Spark, Breeze і Akka.
Сумісність з великими даними	Хороша підтримка через PySpark, Dask.	Часто використовується в екосистемах Hadoop.	Відмінно підходить для Apache Spark, оскільки Scala є його рідною мовою.
Підтримка спільноти	Велика активна спільнота з великими ресурсами.	Розвинене співтовариство, в меншій мірі орієнтоване на науку про дані.	Зростає, особливо в областях великих даних і функціонального програмування.
Інтеграція	Легко інтегрується з різними інструментами і технологіями.	Високий ступінь інтеграції, широко використовується в корпоративних додатках.	Повна інтеграція з бібліотеками Java та платформами обробки великих даних.
Масштабованість	Підходить для невеликих і середньомасштабних додатків обробки даних.	Ідеально підходить для великих систем обробки даних корпоративного рівня.	Оптимізовано для розподілених і масштабованих додатків з великими даними.
Машинне навчання	Відмінна підтримка бібліотек: TensorFlow, Python, Scikit-learn.	Можливості обмежені, хоча можлива інтеграція з такими бібліотеками, як DL4J.	Може інтегруватися із засобами машинного навчання, але не має великої екосистеми Python.

Отже, для реалізації ПЗ для моніторингу екстремістського контенту обрано мову програмування Python.

2.2 Використання зовнішніх бібліотек

TensorFlow і PySpark можна об'єднати, щоб використовувати сильні сторони кожного з них для розробки масштабованих, розподілених та ефективних програм для обробки даних. Така інтеграція особливо корисна для обробки великомасштабних наборів даних та навчання моделей машинного навчання в розподілених середовищах [17]-[21].

PySpark, API Python для Apache Spark, чудово підходить для обробки та перетворення масивних наборів даних у розподілених кластерах. Він ідеально підходить для попередньої обробки даних, очищення та розробки функцій у великомасштабних наборах даних [17].

TensorFlow – це потужна платформа глибокого навчання, здатна створювати, навчати та розгортати складні моделі машинного навчання. Вона вирішує такі завдання, як розпізнавання зображень, обробка природної мови і проектування нейронних мереж [19].

Поєднання цих інструментів дозволяє розробникам попередньо обробляти великі набори даних за допомогою PySpark, а потім використовувати TensorFlow для навчання та оцінки складних моделей [18].

PySpark: готує та обробляє великі набори даних, що зберігаються в розподілених системах зберігання даних (наприклад, Hadoop HDFS, AWS S3) [17].

TensorFlow: навчає моделі машинного навчання на основі попередньо оброблених даних, використовуючи розподілене навчання для забезпечення масштабованості [20].

Розподілені обчислення PySpark можна поєднати зі стратегією `tf.distribute` від TensorFlow для розподілу навчання по декількох вузлах або графічним процесорам [20].

Потокове передавання даних у реальному часі та прогнозування. PySpark: передає дані в режимі реального часу для вилучення ознак.

TensorFlow: застосовує попередньо навчені моделі для прогнозування в режимі реального часу [21].

Управління ресурсами. Обидва інструменти вимагають великих витрат ресурсів. Для оптимізації продуктивності необхідна правильна конфігурація кластера [18].

Розподілені системи ускладнюють налагодження. Такі інструменти, як TENSORBOARD та Spark UI, можуть допомогти у моніторингу та усуненні несправностей [21].

Оброблення та аналіз терабайтів даних за допомогою PySpark, а потім навчання масштабної моделі, використовуючи розподілені можливості TensorFlow. Скорочення проблеми попередньої обробки та навчання, використовуючи паралельність PySpark та прискорення TensorFlow на GPU/TPU. Створення універсального конвеєру для пакетної обробки, потокової передачі даних або виведення даних в режимі реального часу [17].

Поєднання TensorFlow і PySpark створює потужну платформу для розробки надійних, масштабованих і ефективних додатків для обробки даних. Хоча PySpark забезпечує широкомасштабну обробку даних, TensorFlow забезпечує глибоке навчання та розширену аналітику. Використовуючи сильні сторони, розробники можуть створювати комплексні рішення для широкого спектру завдань в області обробки даних [20].

2.3 Вибір середовища для розробки

Розроблене компанією JetBrains, PyCharm – це потужне інтегроване середовище розробки (integrated development environment – IDE), створене спеціально для розробки на Python. Воно широко відоме як одне з кращих IDE для Python завдяки своїм надійним функціям, зручному дизайну і широкій підтримці сучасних методів розробки [22].

PyCharm розроблено з урахуванням особливостей Python, пропонуючи глибоку інтеграцію та підтримку функцій Python. Підсвічування синтаксису

забезпечує чітке візуальне розділення синтаксису Python. Завершення коду – інтелектуальне завершення, адаптоване для Python, включаючи модулі, бібліотеки та спеціальні функції. Інструменти рефакторингу полегшують перейменування змінних, методи видалення та інші перетворення коду, характерні для Python. Відповідність PEP-8 забезпечує відповідність стандартам програмування Python, висвітлюючи проблеми стилю [23].

PyCharm надає надійні засоби налагодження, що полегшують усунення помилок:

- графічний налагоджувач – дозволяє встановлювати точки зупинки, перевіряти змінні та візуально виконувати покрокове виконання коду.
- вбудована налагодження – відображає значення змінних безпосередньо в редакторі під час налагодження.
- віддалена налагодження – підтримує налагодження програм, що працюють на віддалених серверах або в контейнерах Docker.

PyCharm підтримує безліч плагінів і має вбудовані інструменти для підвищення продуктивності. Інтеграція з системою контролю версій – легко працює з Git, SVN, Mercurial та іншими інструментами VCS. PyCharm Professional включає в себе інструменти для роботи з базами даних безпосередньо з IDE. Також доступні тисячі плагінів для додаткової функціональності, включаючи мовну підтримку, теми оформлення та засоби підвищення продуктивності [23].

PyCharm включає функції, які полегшують розробку та підвищують ефективність. Інтелектуальна навігація по коду – швидкий перехід до визначень класів, методів або використання. Дозволяє повторно використовувати блоки коду для прискорення розробки. Виявляє помилки та надає інформацію про потенційні проблеми в режимі реального часу. Підтримує такі завдання, як запуск сценаріїв, управління віртуальними середовищами та тестування безпосередньо з IDE [23].

PyCharm полегшує управління середовищами та залежностями Python. Підтримка віртуального середовища – автоматично визначає та налаштовує

віртуальні середовища, такі як `venv` та `conda`. Інтеграція менеджера пакетів – вбудовані інструменти для управління пакетами Python, включаючи `pip` та `conda` [23].

PyCharm підтримує тестові платформи та полегшує підтримку надійного коду. Модульне тестування – пряма інтеграція з такими платформами, як `unittest`, `pytest` та `nose`. Test Runner – візуальний інтерфейс для виконання та моніторингу результатів тестування [23].

PyCharm – одна з найкращих IDE для Python завдяки своїм спеціалізованим функціям, великій екосистемі та здатності задовольняти запити як новачків, так і професіоналів. Її орієнтація на продуктивність, надійні засоби налагодження та плавна інтеграція з сучасними робочими процесами Python роблять її безцінним інструментом для розробників у різних областях. Незалежно від того, чи створюєте ви простий сценарій або масштабну програму, PyCharm може значно покращити ваш досвід розробки [22].

Підбиваючи підсумок порівняння варто зазначити, що Visual Studio найкраще підходить для розробників, яким потрібна багатомовна IDE і які працюють над кросплатформенними проектами корпоративного рівня. Це добре, якщо Python є лише однією з кількох використовуваних мов. PyCharm ідеально підходить для розробки, орієнтованої на Python, пропонуючи глибоку інтеграцію та спеціалізовані функції для робочих процесів Python, особливо у веб-розробці та обробці даних [23].

Обґрунтування вибору середовища розробки для розробки методів реконструкції тривимірних зображень наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки

Критерій порівняння мов програмування	Мова програмування	
	Visual Studio	Anaconda
Підтримки Python	Потрібне розширення робочого навантаження на Python; підтримка хороша, але менш спеціалізована.	Спеціально розроблена для розробки на Python з глибокими мовними можливостями.
Простоті використання	Зручна для загальної розробки; може здатися складною для проектів, що використовують тільки Python.	Оптимізована для розробки на Python, інтуїтивно зрозуміла для робочих процесів на Python.
Підтримки Платформи	Доступно в Windows, macOS; підтримка Linux обмежена.	Доступна для Windows, macOS і Linux.
Уявлення	Може бути ресурсоємним, особливо при використанні декількох розширень.	Легкий у порівнянні з Visual Studio, але може сповільнювати роботу при роботі з великими проектами або плагінами.
Навігація по коду	Розширені засоби навігації з IntelliSense для Python та інших мов.	Чудова навігація, оптимізована для Python, з такими функціями, як визначення переходу та ієрархія методів.
Налагодження	Надійні засоби налагодження з керуванням точками зупинки та перевіркою змінних для Python.	Однаково надійний, з акцентом на специфічну для Python налагодження та вбудовані засоби візуальної налагодження.
Тестування фреймворків	Підтримує тестування на Python з розширеннями; вимагає додаткової настройки.	Вбудована підтримка фреймворків тестування, таких як unittest, pytest та nose.
Контроль версій	Інтегрована підтримка Git, SVN і TFS; зручні інструменти для спільної роботи.	Потужна інтеграція з Git, Mercurial та SVN, ідеально підходить для проектів Python.
Розширення / плагіни	Велика бібліотека розширень для різних мов та інструментів.	Тисячі плагінів, орієнтованих на Python та пов'язаних з ним робочі процеси.
Управління віртуальним середовищем	Базова підтримка віртуальних середовищ Python за допомогою конфігурації.	Автоматичне визначення та налаштування віртуальних середовищ (наприклад, venv, conda).
Підтримка наукових досліджень даних	Підходить для ноутбуків Jupyter і робочих процесів в області науки про дані з	Потужна інтеграція з Jupyter Notebooks та бібліотеками, такими як Pandas, NumPy та

	розширеннями.	Matplotlib.
--	---------------	-------------

3 ОПИС ПРОГРАМИ

3.1 Загальна схема роботи

На схемі – рис. 3.1, показано робочий процес програми на базі RNN, призначеної для моніторингу екстремістського контенту. Ось що являє собою кожен компонент і з'єднання.

Збір даних – джерела даних, такі як платформи соціальних мереж, вебсайти та форуми, є вхідними даними для системи. Ці компоненти збирають вихідний текст або мультимедійні матеріали, які потенційно можуть містити екстремістський контент.

Попереднє оброблення – цей крок включає очищення, розмітку та нормалізацію вихідних даних для підготовки їх до подальшого аналізу. Ці операції забезпечують узгодженість і видаляють непотрібний шум, роблячи дані придатними для використання.

Витяг ознак. Включає в себе створення структурованих числових уявлень попередньо оброблених даних. Такі методи, як вбудовування та векторизація, перетворюють текст у формати, придатні для моделей машинного навчання.

Аналіз RNN. Основний аналіз, проведений періодичними нейронними мережами (RNN), який обробляє послідовні дані (наприклад, пропозиції). Діаграма включає в себе кілька рівнів RNN, які витягують шаблони, що вказують на екстремістський контент [24].

Класифікація. Вихід RNN передається класифікатору, щоб класифікувати вміст як екстремістський або неекстремістський. Цей крок дозволяє отримати практичні результати [24].

Звітність. Для інформування осіб, які приймають рішення, створюються кінцеві результати, включаючи оповіщення, візуалізацію і

корисну інформацію. Ці результати можуть бути використані для подальших дій, моніторингу або розслідування.

Потік даних. Стрілки відображають послідовний потік даних через систему від збору до складання остаточного звіту.

Схема чітко впорядковує та пояснює, як дані передаються через додаток для моніторингу на базі RNN, показуючи, як кожен компонент взаємодіє для виявлення та категоризації екстремістського вмісту. Вона ілюструє наскрізний процес від отримання даних до практичних результатів.

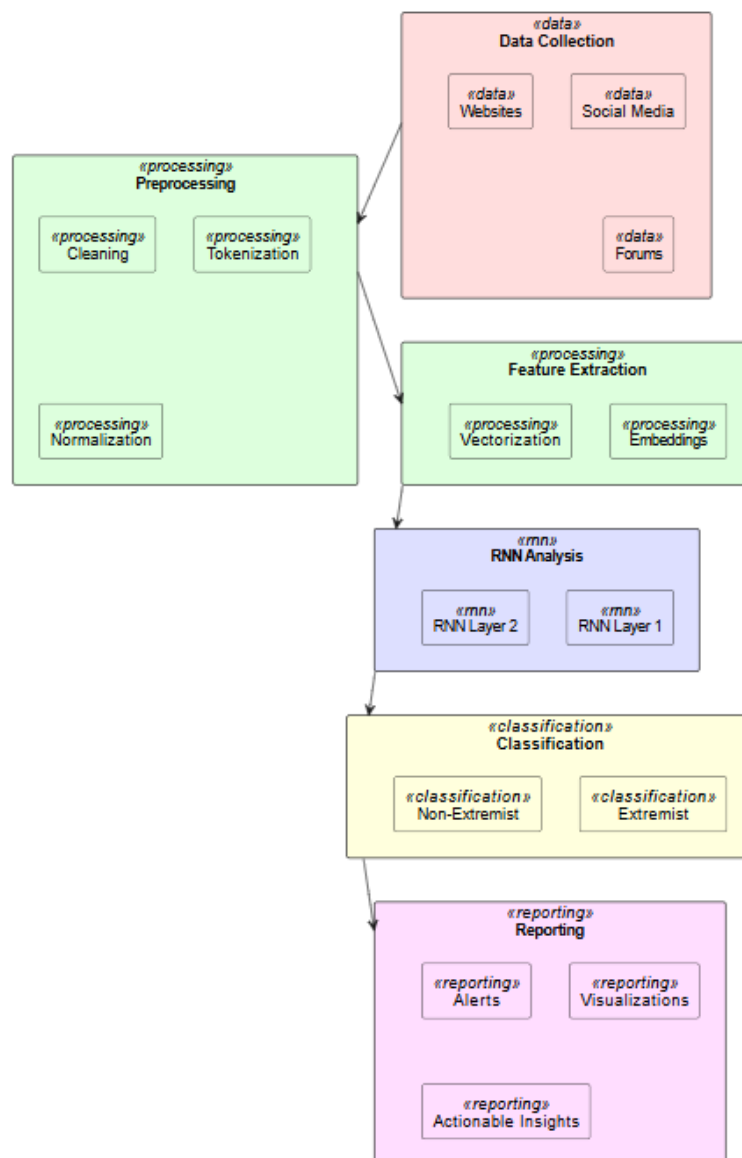


Рисунок 3.1 – Загальна схема роботи

Діаграма класів на рис. 3.2 – це структурне представлення проекту програми на базі RNN для моніторингу екстремістського контенту, що показує, як взаємодіють різні модулі (представлені як класи).

Клас `DataCollection` – надсилає зібрані необроблені дані до класу попередньої обробки.

Клас `Preprocessing` Class` – виводить оброблені дані в клас `FeatureExtraction`.

Клас `FeatureExtraction` – надає витягнуті функції класу `RNNModel`.

Клас `RNNModel` – передає проаналізовані дані в клас `Класифікація``.

Клас `Classification` Class` – відправляє результати класифікації в клас звіти.

Клас `Reporting` – це кінцева точка, де зацікавлені сторони отримують доступ до результатів, які можна використовувати для вжиття заходів.

Така структура забезпечує модульність, що полегшує розробку, тестування та підтримку програми.

Діаграма класів UML – це структурний проект програми на основі RNN для моніторингу екстремістського вмісту, що показує, як взаємодіють різні модулі (представлені як класи). Ось як це працює:

Клас `DataCollection` – надсилає зібрані необроблені дані до класу попередньої обробки. Методи `collectSocialMedia()`, `collectWebsites ()`, `collectForums ()` збирають необроблені дані з різних джерел.

Клас `Preprocessing` – виводить оброблені дані в клас `FeatureExtraction`.

Методи:

- функція `cleanData ()` видаляє шум і непотрібну інформацію;
- функція `tokenizeData ()` розбиває текст на менші компоненти, такі як слова чи речення;
- `normalizeData ()` стандартизує текстовий формат (наприклад, нижній регістр, початковий рядок).

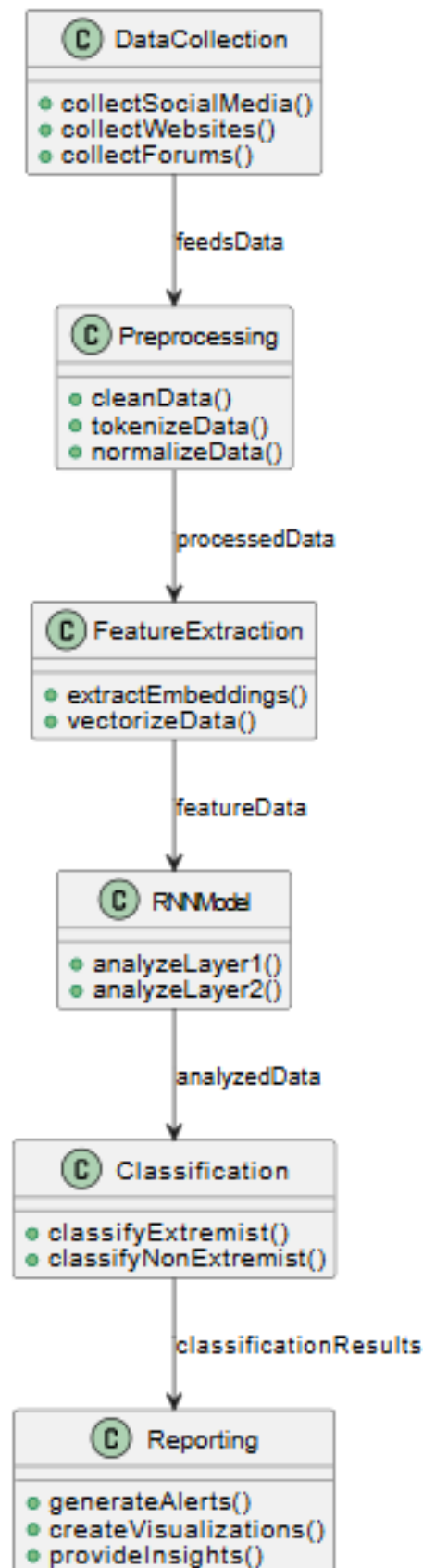


Рисунок 3.2 – Діаграма класів

Клас FeatureExtraction – надає витягнуті функції класу RNNModel.

Методи:

- функція extractEmbeddings () перетворює слова або фрази в щільні числові вектори;
- функція vectorizeData () перетворює текст у структуровані числові формати, придатні для машинного навчання.

Клас RNNModel – передає проаналізовані дані в клас класифікації.

Методи:

- analyzeLayer1 () обробляє дані об'єктів на першому рівні RNN;
- analyzeLayer2 () уточнює аналіз, використовуючи більш глибокий рівень RNN.

Клас Classification – надсилає результати класифікації до звітного класу.

Методи:

- функція classifyExtremist () визначає вміст, який відповідає екстремістським шаблонам;
- функція classifyNonExtremist () класифікує безпечний контент.

Клас Reporting – це кінцева точка, де зацікавлені сторони отримують практичні результати.

Методи:

- функція generateAlerts () видає попередження при виявленні екстремістського контенту.
- функція createVisualizations () надає візуальне представлення результатів.
- функція provideInsights () надає корисні резюме або аналітичні матеріали.

3.2 Алгоритм використання RNN для моніторингу екстремізму

Використання RNN для моніторингу екстремістського вмісту є ефективним підходом завдяки їх здатності аналізувати послідовні дані, такі як текст, і фіксувати контекстуальні зв'язки між словами з часом. RNN особливо добре підходять для таких завдань, як аналіз настроїв, класифікація вмісту та виявлення тенденцій екстремістської поведінки в публікаціях у соціальних мережах, блогах чи форумах. Нижче буде описано основний алгоритм використання RNN в контексті моніторингу екстремістського контенту [25].

Перший етап – збір даних. Напрацювання вхідних даних – текстові дані (наприклад, записи в соціальних мережах, статті, коментарі, повідомлення) з різних платформ, на яких може поширюватися екстремістський контент. У якості джерел розглядають: дані можна збирати з таких платформ, як соціальні мережі або ж форуми. Після чого починається попередня обробка, під час якої необроблені дані потрібно очистити та відформатувати [25]. Це включає в себе:

- видалення стоп-слів, розділових знаків і спеціальних символів;
- приведення тексту в нижній регістр для стандартизації;
- лексемізація – розбивка тексту на більш дрібні частини (наприклад, слова або підслова);
- лематизація або виділення кореня – приведення слів до їх кореневої форми для обробки варіацій;
- робота з незбалансованими класами (при наявності непропорційно великої кількості неекстремістського контенту).

Другий етап – трансформація слів. Оскільки RNN мають справу з послідовностями даних, дуже важливо перетворити оригінальний текст у числові подання, зрозумілі моделі. Зазвичай це робиться за допомогою вбудовувань, таких як Word2Vec, GloVe, або навіть контекстних вбудовувань від трансформаторів, таких як BERT [25]:

- Word2Vec – кожне слово в тексті представлено у вигляді вектора в багатовимірному просторі, де семантично схожі слова розташовані ближче;
- GloVe – глобальне векторне представлення, яке вивчається на

основі збігу слів у корпусі.

Третій етап – опрацювання за допомогою RNN. RNN обробляє послідовність слів (або токенів) крок за кроком. На кожному кроці RNN оновлює свій прихований стан на основі поточного слова та попереднього прихованого стану, що дозволяє йому фіксувати послідовні шаблони. Можна використовувати простий RNN, але найчастіше перевага надається довготривалій короткочасній пам'яті (long short-term memory – LSTM) або керованим рекурентним модулям (gated recurrent units – GRU) через їх здатність обробляти довгострокові залежності та вирішувати проблеми з градієнтом вимирання [26]:

- LSTM/GRU – це спеціалізовані RNN, призначені для вивчення довгострокових залежностей шляхом підтримки стану пам'яті. Вони мають елементи керування, які керують потоком інформації, допомагаючи моделі запам'ятовувати або забувати певні частини послідовності;

- вхідне значення – визначає, яку інформацію дозволити вводити;
- значення забуття – визначає, яку інформацію слід видалити;
- вихідне значення – визначає, яка інформація виводиться з поточного кроку часу.

Наступний етап – вилучення ознак або ж виконання контекстуального аналізу. RNN витягує контекстуальні характеристики з послідовності слів, використовуючи прихований стан на кожному кроці часу. Ця контекстуальна інформація дозволяє мережі вловлювати ширший зміст речень або абзаців, що важливо для виявлення екстремістських ідеологій, розпалювання ненависті та агресивної риторики [26]:

- аналіз настроїв – модель може витягувати характеристики настроїв, аналізуючи, чи містить повідомлення агресивні або запальні висловлювання;

- розпізнавання іменованих об'єктів – RNN може допомогти ідентифікувати ключові об'єкти, такі як політичні діячі, організації або терористичні групи, згадані в тексті, які мають вирішальне значення для розуміння значущості змісту [25];

– механізм привернення уваги на рівні слів – щоб зосередитися на конкретних словах або фразах, які мають більше значення для визначення екстремізму, в рамках RNN можуть бути застосовані механізми залучення уваги.

Передостанній етап – класифікаційний рівень. Повністю підключений шар – після обробки вхідних даних через шари RNN модель зазвичай передає остаточний прихований стан або агреговані дані послідовності на повністю підключений (щільний) шар для класифікації. Цей шар видає прогноз на основі витягнутих об'єктів. Активація Softmax – для класифікації за кількома класами (наприклад, для віднесення тексту до екстремістських, неекстремістських або інших заздалегідь визначених категорій) зазвичай використовується функція активації softmax для обчислення розподілу ймовірності по всіх класах. Активація сигмовидної форми – у бінарній класифікації (наприклад, екстремістській чи неекстремістській) функція активації сигмовидної форми може бути використана для отримання оцінки ймовірності від 0 до 1. Порогове значення – на основі ймовірності висновку встановлюється порогове значення для класифікації вмісту як екстремістського чи неекстремістського. Наприклад, якщо оцінка ймовірності віднесення до класу екстремістських перевищує певний поріг (наприклад, 0,5), вміст позначається [25].

Останнім етапом є подальша обробка та виведення даних. Постобробка – після класифікації позначений вміст може бути додатково оброблений для перевірки користувачем або автоматизованих дій (наприклад, видалення записів, повідомлення органів влади або видалення вмісту). Цикл зворотного зв'язку – впровадження циклу зворотного зв'язку, де позначений вміст перевіряється експертами, може допомогти вдосконалити модель. Позначені дані можуть бути повернуті в конвеєр навчання для підвищення точності моделі з плином часу. Оповіщення в режимі реального часу – система моніторингу може бути налаштована для безперервного аналізу контенту та

оповіщення влади або модераторів в режимі реального часу про виявлення екстремістського контенту [26].

Повторювані нейронні мережі, зокрема LSTM та GRU, є потужними інструментами моніторингу екстремістського вмісту завдяки своїй здатності фіксувати довгострокові залежності та послідовну інформацію в тексті. Використовуючи шар вбудовування для перетворення тексту у векторне представлення, обробляючи дані за допомогою шару RNN та застосовуючи методи класифікації, ці моделі можуть ефективно виявляти екстремістські ідеології, що розпалюють ненависть висловлювання та Контент, що пропагує насильство. Однак для ефективної реалізації в реальних додатках необхідно вирішити такі проблеми, як незбалансованість даних, упередженість моделей і розуміння тонкощів мови, що використовується екстремістами [26].

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Метою роботи є дослідження та розробка програмних засобів для моніторингу екстремістського контенту.

Головні функціональні вимоги можна представити наступним чином:

а) вимоги до обробки даних – стосуються збору, попередньої обробки та управління вхідними даними:

- 1) можливість збору даних з декількох джерел, включаючи платформи соціальних мереж та форуми;
- 2) інтеграція з API для автоматизованого отримання даних;
- 3) підтримка як текстового, так і мультимедійного контенту (зображень, відео і т.д.);
- 4) безпечне зберігання зібраних даних за допомогою шифрування для захисту конфіденційної інформації;
- 5) масштабованість для обробки великих наборів даних із безперервних потоків даних;
- 6) зберігання метаданих, таких як мітки часу, ідентифікатори джерел та географічні дані;
- 7) очищення тексту для усунення перешкод (наприклад, HTML-тегів, спеціальних символів);
- 8) токенизація для поділу тексту на значущі елементи (слова, пропозиції);
- 9) розпізнавання і нормалізація мови (наприклад, стеммінг, лемматизація);

б) аналітичні вимоги – визначають основні функціональні можливості виявлення екстремістського контенту:

- 1) перетворення текстових даних у вкладення (наприклад, за допомогою Word2Vec, GloVe або BERT);

- 2) використання контекстно-залежного представлення об'єктів для послідовного аналізу;
- 3) впровадження RNN (або його варіантів, таких як LSTM або GRU) для аналізу послідовних текстових шаблонів;
- 4) навчання моделі виявляти ознаки екстремістського контенту, включаючи:
 - розпалюють ненависть висловлювання;
 - погрози застосування насильства або підбурювання;
 - ідеологічну пропаганду;
- 5) можливість адаптації та перепідготовки на основі нових наборів даних для підвищення точності;
- 6) класифікація вмісту за заздалегідь визначеними категоріями (екстремістський, не екстремістський, неоднозначний (потребує подальшого розгляду), присвоєння класифікаціям ступеня достовірності);
- 7) оцінка вмісту для визначення рівня загрози або серйозності;
- 8) створення показників пріоритетності для негайних дій;
- 9) вимоги до звітності та оповіщення – стосуються того, як додаток представляє та передає результати:
- 10) сповіщення в режимі реального часу про виявлений екстремістський контент;
- 11) порогові значення сповіщень, що налаштовуються, засновані на ступені серйозності або достовірності;
- 12) інформаційні панелі, що відображають тенденції в змісті, розподіл джерел і рівні загрози;
- 13) визначення актуальних тем або ключових слів;
- в) вимоги до безпеки та відповідності вимогам – забезпечують дотримання етичних норм і безпеку операцій:
 - 1) управління доступом на основі ролей для користувачів;
 - 2) знеособлення персональних даних;

г) вимоги до масштабованості та продуктивності – стосуються здатності програми справлятися зі зростаючими вимогами:

1) обробляти зростаючі обсяги даних за допомогою розподіленої обробки;

2) оброблення та класифікація контенту в режимі реального часу для своєчасного реагування;

3) відновлення після збоїв в процесі збору даних або виведення моделі;

д) вимоги до адаптивності та технічного обслуговування – спрямовані на забезпечення того, щоб застосунок залишався ефективним з часом:

1) періодична перепідготовка для адаптації до мінливих формулювань або тактики екстремістів.

Основні функціональні вимоги до програми на базі RNN для моніторингу екстремістського контенту спрямовані на забезпечення надійної обробки даних, ефективне виявлення екстремістських проявів, своєчасну звітність та дотримання етичних та правових норм. Реалізація цих функцій може зробити додаток цінним інструментом для боротьби з екстремістською діяльністю.

4.2 Стратегія тестування

Під час модульного тестування відбувається перевірка окремих компонентів, таких як збір даних, попередня обробка та вилучення об'єктів. Необхідно переконатися, що кожна функція (наприклад, `tokenize Data ()`, `cleanData()`) працює належним чином.

Інтеграційне тестування спрямовано на перевірку взаємодії між компонентами (наприклад, як результати попередньої обробки використовуються при вилученні об'єктів). Додатково перевіряється безперебійність проходження даних по конвеєру.

Тестування моделі спрямовано на оцінку мережі RNN, що використовується під час роботи

Тестування продуктивності спрямовано на оцінку здатності програми обробляти великі набори даних. Вимірювання затримки при класифікації та генерації попереджень.

Тестування безпеки в рамках ПЗ, що розробляється допоможе переконатися, що конфіденційні дані зашифровані. Більш того, засвідчить дотримання правил конфіденційності даних.

Генеральний сценарій тестування відбувається на основі схеми роботи застосунку – рис. 4.1.

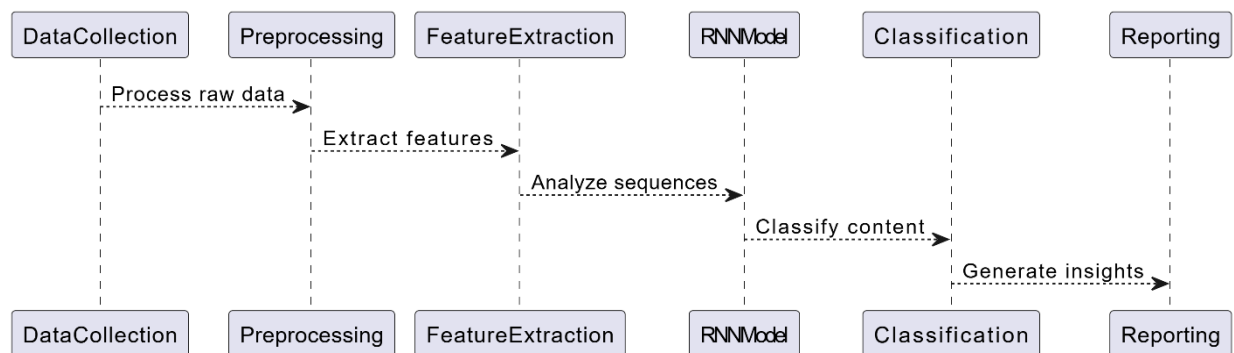


Рисунок 4.1 – Генеральний сценарій тестування на основі схеми роботи

Загальний перебіг складається з:

а) збір та попередня обробка даних:

- 1) збір текстових даних з платформ і їх очищення;
- 2) маркування, видалення стоп-слів і підготовка даних до впровадження;

б) представлення тексту за допомогою впроваджень:

- 1) використання попередньо підготовлених вставок слів або контекстних вставок для перетворення необробленого тексту у вектори;

в) архітектура моделі RNN:

- 1) реалізація рівнів LSTM та GRU для обробки послідовного

характеру тексту;

2) навчання моделі вловлювати контекстуальні зв'язки в тексті;

г) витяг ознак і контекстуальний аналіз:

1) витягування релевантних об'єктів, використовуючи механізми залучення уваги і аналіз настроїв;

2) визначення ключових об'єктів та контекстних підказок, які вказують на екстремістський вміст;

д) класифікація;

е) подальша обробка:

1) визначення екстремістського контенту і вживання заходів (наприклад, видалення, перегляд).

4.3 Тестування точності роботи

Для тестування точності було обрано вибірку даних з відкритих джерел, а саме: How ISIS Uses Twitter [27]. Приклад вибірки наведено на рис. 4.2, а детальна інформація у табл. 4.1.

Варто відзначити, що вибірка оцінюється на ознаки екстремізму на основі різних підходів, зокрема [28]:

– кластерний аналіз соціальних медіа: хто є основними гравцями в мережі Twitter [28];

– аналіз ключових слів: які ключові слова на основі назви, імені користувача, опису, місцезнаходження та твітів найчастіше використовували терористи: «baqiya», «dabiq», «wilayat», «amaq» [28];

– класифікація посилань: на які вебсайти посилаються терористи та прихильники радикалізму [28];

– аналіз настроїв: кого зі священнослужителів найчастіше цитують терористи і кого з них вони ненавидять найбільше [29].

Таблиця 4.1 – Загальна інформація про вибірку для тестування точності

Джерело даних	Соціальна мережа Twitter
Число повідомлень	17000
Число користувачів	150
Подія-тригер	Терористична атака в Парижі – 2015 р.
Інформативні ознаки	Заголовок Ім'я користувача Опис Місце розташування Кількість підписників на момент завантаження твіту Кількість статусів, присвоєних Користувачем на момент завантаження твіту Дата і часова мітка твіту Текст твіту

name	username	description	location	# followers	# n
Rami	Uncle_SamCoco	[null]	[null]	16	1
War BreakingNews	RamiAlLolah	Here to defend the...	Read my blog	34.7k	1
Other (14744)	Other (14355)	Other (13148)	Other (9957)	640	49
GunsandCoffee	GunsandCoffee70	ENGLISH TRANSLATIONS: http://t.co/QLdJ0fte ws		640	49
GunsandCoffee	GunsandCoffee70	ENGLISH TRANSLATIONS: http://t.co/QLdJ0fte ws		640	49
GunsandCoffee	GunsandCoffee70	ENGLISH TRANSLATIONS: http://t.co/QLdJ0fte ws		640	49

Рисунок 4.2 – Приклад вибірки

Результати роботи застосунку наведено у табл. 4.2.

Таблиця 4.2 – Результати тестування застосунку

Час роботи	6025,3 с
Точність роботи під час навчання	97%
Виявлено потенційно екстремістського контенту (етап навчання)	86% твітів з загальної вибірки
Точність роботи на етапі тестування	95,4%
Виявлено потенційно екстремістського контенту (етап тестування)	82,3% твітів з загальної вибірки (7,1% були визначені – як підозрілі)

З результатів роботи видно прийнятну точність запропонованого підходу, проте під час тестової роботи застосунок відсіяв 7,1% твітів, як таких, що викликають сумніви та вимагають додаткового опрацювання. Усі вони були відповідями на коментарі інших людей, тобто в системи були проблеми із розумінням загального контексту набору повідомлень.

4.4 Проблеми та виклики

Під час роботи були виявлені наступні виклики та проблеми:

- дисбаланс даних – екстремістський вміст часто становить невелику частку від загального обсягу даних, що може призвести до незбалансованості наборів даних. Можуть допомогти такі методи, як надмірна вибірка, недостатня вибірка або використання вагових коефіцієнтів класів у функції втрат;
- упередженість – модель повинна бути ретельно підготовлена, щоб уникнути упередженості, особливо при виявленні екстремістського змісту, пов'язаного з певними політичними чи релігійними групами. Справедливість і прозорість при прийнятті рішень за моделлю є ключовими факторами;
- мовні нюанси – мова, що використовується в екстремістському контенті, часто включає сленг, зашифровані вирази або евфемізми. Модель повинна бути навчена роботі з різними даними, щоб ефективно розуміти ці нюанси.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи магістра було досліджено та розроблено нові засоби моніторингу екстремістського контенту.

Тож, головною метою роботи було визначено дослідження та розробка методів моніторингу екстремістського контенту.

У роботі розроблявся застосунок для моніторингу екстремістського контенту.

В основній частині було представлено аналітичну частину роботи з результатами огляду найбільш відомих застосунків моніторингу екстремістського контенту. Проаналізовані аналоги мали низку проблем, для вирішення яких було обрано розробити новий підхід та реалізувати його.

Для дослідження викладених методик використовувалася реальна вибірка даних під час опрацювання якої запропонований підхід продемонстрував прийнятні результати.

За результатами роботи слід зробити висновок про те, що новий спосіб моніторингу екстримістського контенту продемонстрував хороші результати..

В якості можливих напрямків розвитку теми можна опрацювання помітних проблем:

- дисбаланс даних – екстремістський вміст часто становить невелику частку від загального обсягу даних, що може призвести до незбалансованості наборів даних. Можуть допомогти такі методи, як надмірна вибірка, недостатня вибірка або використання вагових коефіцієнтів класів у функції втрат;

- упередженість – модель повинна бути ретельно підготовлена, щоб уникнути упередженості, особливо при виявленні екстремістського змісту, пов'язаного з певними політичними чи релігійними групами. Справедливість і прозорість при прийнятті рішень за моделлю є ключовими факторами;

- мовні нюанси – мова, що використовується в екстремістському контенті, часто включає сленг, зашифровані вирази або евфемізми. Модель

повинна бути навчена роботі з різними даними, щоб ефективно розуміти ці нюанси.

Новизна роботи полягає в тому, що запропоновано новий підхід моніторингу екстремістського контенту на основі рекурентних нейронних мереж.

Практичне значення роботи полягає в тому, що розроблено програмне забезпечення для моніторингу екстремістського контенту.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Іванова А. Політичний екстремізм в контексті глобалізаційних викликів сучасності [Текст] / А. Іванова // *Evropský politický a právní diskurz*. - 2018. - Sv. 5, Vyd. 3. - С. 101-106.
2. Гаман І. Г. Організаційно-правові засади протидії екстремізму та шляхи вдосконалення [Текст] / І. Г. Гаман // *Чесť і закон*. - 2023. - № 3. - С. 61-64.
3. Звоздецька О. Особливості проявів правого екстремізму та тероризму в європейському інформаційному просторі [Текст] / О. Звоздецька, О. Лучик // *Історико-політичні проблеми сучасного світу*. - 2024. - Т. 49. - С. 186-201.
4. What is terrorist and violent extremist content? [Electronical resource]. – Access mode: <https://www.dia.govt.nz/Countering-Violent-Extremism-What-is-terrorist-and-violent-extremist-content>.
5. Terrorist and violent extremist content [Electronical resource]. – Access mode: <https://www.oecd.org/en/topics/terrorist-and-violent-extremist-content.html>.
6. Preventing violent extremism through promoting inclusive development, tolerance and respect for diversity. United Nations Development Programme a development response to addressing radicalization and violent extremism [Electronical resource]. – Access mode: <https://www.undp.org/sites/g/files/zskgke326/files/publications/Discussion%20Paper%20%20Preventing%20Violent%20Extremism%20by%20Promoting%20Inclusive%20%20Development.pdf>
7. Online radicalisation. What we know [Electronical resource]. – Access mode: https://home-affairs.ec.europa.eu/system/files/2023-11/RAN-online-radicalisation_en.pdf.
8. Violent extremism [Electronical resource]. – Access mode: <https://dig.watch/topics/violent-extremism>.

9. ActiveFence [Electronical resource]. – Access mode: <https://www.weforum.org/organizations/activefence/>.
10. ActiveFence. Fostering Healthy and Compliant Online [Electronical resource]. – Access mode: <https://www.activefence.com/>.
11. Moonshot CVE [Electronical resource]. – Access mode: <https://moonshotteam.com/>.
12. Hatebase [Electronical resource]. – Access mode: <https://hatebase.org/>.
13. Hatebase [Electronical resource]. – Access mode: <https://www.unesco.org/en/world-media-trends/hatebase>.
14. Python [Electronical resource]. – Access mode: <https://www.python.org/>.
15. Real Python Tutorials [Electronical resource]. – Access mode: <https://realpython.com/>.
16. Python Introduction [Electronical resource]. – Access mode: https://www.w3schools.com/python/python_intro.asp.
17. PySpark Overview [Electronical resource]. – Access mode: <https://spark.apache.org/docs/latest/api/python/index.html#pyspark-overview>.
18. PySpark [Electronical resource]. – Access mode: <https://www.databricks.com/glossary/pyspark>.
19. TensorFlow [Electronical resource]. – Access mode: <https://www.tensorflow.org>.
20. TensorFlow і машинне навчання [Electronical resource]. – Access mode: <https://foxminded.ua/tensorflow-shcho-tse/>.
21. TensorFlow fundamentals [Electronical resource]. – Access mode: <https://learn.microsoft.com/en-us/training/paths/tensorflow-fundamentals/>.
22. PyCharm. The Python IDE for data science and web development [Electronical resource]. – Access mode: <https://www.jetbrains.com/pycharm/>.
23. PyCharm як найкраща IDE для розробки ПЗ на Python [Electronical resource]. – Access mode: <https://foxminded.ua/pycharm-tse/>.

24. What is a recurrent neural network (RNN)? [Electronical resource]. – Access mode: <https://www.ibm.com/topics/recurrent-neural-networks>.
25. An Introduction to Recurrent Neural Networks (RNNs) [Electronical resource]. – Access mode: <https://medium.com/@researchgraph/an-introduction-to-recurrent-neural-networks-rnns-802fcfee3098>.
26. Attention as an RNN [Electronical resource]. – Access mode: <https://arxiv.org/abs/2405.13956>.
27. How ISIS Uses Twitter [Electronical resource]. – Access mode: <https://www.kaggle.com/datasets/fifthtribe/how-isis-uses-twitter/data>.
28. Multi-Ideology, Multiclass Online Extremism Dataset, and Its Evaluation Using Machine Learning [Electronical resource]. – Access mode: <https://onlinelibrary.wiley.com/doi/10.1155/2023/4563145>.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення для моніторингу екстремістського контенту.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Дослідження та програмна реалізація засобів моніторингу екстремістського контенту», затверджене наказом Національного університету «Запорізька політехніка» № 398 від 07 жовтня 2024 р.

A.2 Призначення розробки

Метою роботи є розробка та дослідження нових засобів моніторингу екстремістського контенту.

Під час роботи буде розроблено новий засіб моніторингу екстремістського контенту на основі використання рекурентних нейронних мереж (recurrent neural network – RNN) для:

- аналізу тексту – виявлення висловлювань, що розпалюють ненависть, та екстремістської лексики;
- аналізу мовлення – аналіз аудіо на наявність шкідливої риторики;
- моніторингу відео – виявлення екстремістського контенту в підписах або аудіо.

Розроблення нового засобу моніторингу екстремістського контенту дозволить:

- виявляти та вирішувати конфліктні ситуації в режимі реального часу;
- точніше та детальніше розуміти контент;
- автоматизувати та масштабувати моніторинг на різних платформах.

Нові рішення покращать моніторинг екстремістського вмісту шляхом аналізу послідовних даних, забезпечуючи точні, масштабовані та проактивні системи виявлення.

А.3 Основні вимоги до програми, що розробляється

А.3.1 Вимоги до функціональних характеристик

Головні функціональні вимоги можна представити наступним чином:

а) вимоги до обробки даних – стосуються збору, попередньої обробки та управління вхідними даними:

- 1) можливість збору даних з декількох джерел, включаючи платформи соціальних мереж та форуми;
- 2) інтеграція з API для автоматизованого отримання даних;
- 3) підтримка як текстового, так і мультимедійного контенту (зображень, відео і т.д.);
- 4) безпечне зберігання зібраних даних за допомогою шифрування для захисту конфіденційної інформації;
- 5) масштабованість для обробки великих наборів даних із безперервних потоків даних;
- 6) зберігання метаданих, таких як мітки часу, ідентифікатори джерел та географічні дані;
- 7) очищення тексту для усунення перешкод (наприклад, HTML-тегів, спеціальних символів);
- 8) токенизація для поділу тексту на значущі елементи (слова, пропозиції);
- 9) розпізнавання і нормалізація мови (наприклад, стеммінг, лемматизація);

б) аналітичні вимоги – визначають основні функціональні можливості виявлення екстремістського контенту:

- 1) перетворення текстових даних у вкладення (наприклад, за

допомогою Word2Vec, GloVe або BERT);

2) використання контекстно-залежного представлення об'єктів для послідовного аналізу;

3) впровадження RNN (або його варіантів, таких як LSTM або GRU) для аналізу послідовних текстових шаблонів;

4) навчання моделі виявляти ознаки екстремістського контенту, включаючи:

- розпалюють ненависть висловлювання;
- погрози застосування насильства або підбурювання;
- ідеологічну пропаганду;

5) можливість адаптації та перепідготовки на основі нових наборів даних для підвищення точності;

6) класифікація вмісту за заздалегідь визначеними категоріями (екстремістський, не екстремістський, неоднозначний (потребує подальшого розгляду), присвоєння класифікаціям ступеня достовірності);

7) оцінка вмісту для визначення рівня загрози або серйозності;

8) створення показників пріоритетності для негайних дій;

9) вимоги до звітності та оповіщення – стосуються того, як додаток представляє та передає результати:

10) сповіщення в режимі реального часу про виявлений екстремістський контент;

11) порогові значення сповіщень, що налаштовуються, засновані на ступені серйозності або достовірності;

12) інформаційні панелі, що відображають тенденції в змісті, розподіл джерел і рівні загрози;

13) визначення актуальних тем або ключових слів;

в) вимоги до безпеки та відповідності вимогам – забезпечують дотримання етичних норм і безпеку операцій:

1) управління доступом на основі ролей для користувачів;

2) знеособлення персональних даних;

г) вимоги до масштабованості та продуктивності – стосуються здатності програми справлятися зі зростаючими вимогами:

1) обробляти зростаючі обсяги даних за допомогою розподіленої обробки;

2) оброблення та класифікація контенту в режимі реального часу для своєчасного реагування;

3) відновлення після збоїв в процесі збору даних або виведення моделі;

д) вимоги до адаптивності та технічного обслуговування – спрямовані на забезпечення того, щоб застосунок залишався ефективним з часом:

1) періодична перепідготовка для адаптації до мінливих формулювань або тактики екстремістів.

Основні функціональні вимоги до програми на базі RNN для моніторингу екстремістського контенту спрямовані на забезпечення надійної обробки даних, ефективне виявлення екстремістських проявів, своєчасну звітність та дотримання етичних та правових норм. Реалізація цих функцій може зробити додаток цінним інструментом для боротьби з екстремістською діяльністю.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс ПЗ для засобів моніторингу екстремістського контенту повинен бути ергономічним та зрозумілим для користувачів.

А.3.3 Вимоги до надійності

Вимоги до безпеки та відповідності вимогам – забезпечують дотримання етичних норм і безпеку операцій:

- управління доступом на основі ролей для користувачів;
- знеособлення персональних даних;

А.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для реконструкції тривимірних зображень необхідна наявність персонального комп'ютера у користувача.

А.3.5 Вимоги до складу та параметрів технічний засобів

Для експлуатації програмного забезпечення для реконструкції тривимірних зображень необхідно таке апаратне та програмне забезпечення:

- від 8 ГБ оперативної пам'яті;
- від 2-х ядер, обчислювальної потужності – 2.0-5.6 ГГц;
- 2 ГБ жорсткий диск;
- 64-бітна платформа;
- встановлені пакети PySpark та TensorFlow (або наявність безперебійного доступу до мережі).

А.3.6 Вимоги до маркування і пакування

ПЗ моніторингу екстремістського контенту може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення для засобів моніторингу екстремістського контенту».

ДОДАТОК Б
Фрагмент тексту програми

```

from binascii import unhexlify
from Plum import Plum

import datetime
from operator import attrgetter
from pathlib import Path
from typing import Set, Optional

import os
import monit
from cryptography.hazmat.primitives.serialization
import Encoding

from tru_storS_obs.certS_repository import
HMCatcertSRepository

from tru_storS_obs.HMCat_recD import
HMCatCertificaterecD

class platPlum(Plum):
    """The list of plats supported by TSO."""

    APPLE = 1
    GOOGLE_AOSP = 2
    MICROSOFT_WINDOWS = 3
    MOZILLA_NSS = 4
    ORACLE_JAVA = 5
    OPENJDK = 6

    # TODO (AD)

```

```
# DEBIAN or UBUNTU?
```

```
class truStore:
    """The set of HMCat certS that compose the tru
store of one plat at a specific time."""

    def __init__(
        self,
        plat: platPlum,
        ver: Optional[str],
        url: str,
        date_fet: datetime.date,
        trued_certS: Set[HMCatCertificaterecD],
        blocked_certS:
Optional[Set[HMCatCertificaterecD]] = None,
    ) -> None:
        if blocked_certS is None:
            blocked_certS = set()
        self.plat = plat

        if ver is not None:
            ver = ver.strip()
            self.ver = ver

        self.url = url.strip()
        self.date_fet = date_fet
        self.trued_certS = trued_certS
        self.blocked_certS = blocked_certS
```

```

        # TODO(AD): Track additional constraints
such as whether the cert is trued, blocked, or "always
ask" (Apple)

```

```

        # or Disabled / notBefore (MSFT) -
basically anything that has to do with what the
certificate can do but that

```

```

        # is not stored as a field in the
certificate itself

```

```

def __eq__(self, other: object) -> bool:
    if not isinstance(other, truStore):
        return False

    # Two storS are equal if all their fields
except for date_fet are equal

```

```

        self_dict = self.__dict__.copy()
        self_dict.pop("date_fet")
        other_dict = other.__dict__.copy()
        other_dict.pop("date_fet")
        return other_dict == self_dict

```

```

import tensorflow as tf

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense,
Dropout

```

```

from tensorflow.keras.optimizers import Adam
import numpy as np
import librosa
import os

```

```

# Load and preprocess data

```

```

def load_audio_files(data_path, max_samples=1000):

```

```

X, y = [], []
labels = {}
for idx, folder in
enumerate(os.listdir(data_path)):
    labels[folder] = idx
    folder_path = os.path.join(data_path,
folder)
    for file in
os.listdir(folder_path)[:max_samples]:
        file_path = os.path.join(folder_path,
file)
        signal, sr = librosa.load(file_path,
sr=None)
        mfcc = librosa.feature.mfcc(signal,
sr=sr, n_mfcc=20)
        mfcc = np.mean(mfcc.T, axis=0) #
Average MFCCs over time
        X.append(mfcc)
        y.append(idx)
    return np.array(X), np.array(y), labels

# Normalize features
def normalize_data(X):
    return (X - np.mean(X, axis=0)) / np.std(X,
axis=0)

# Prepare dataset
data_path = "./speech_dataset" # Replace with your
dataset path
X, y, labels = load_audio_files(data_path)

```

```

X = normalize_data(X)

# Train-test split
from sklearn.model_selection import
train_test_split
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.2, random_state=42)

# One-hot encode labels
from tensorflow.keras.utils import to_categorical
y_train = to_categorical(y_train,
num_classes=len(labels))
y_test = to_categorical(y_test,
num_classes=len(labels))

# Reshape for LSTM input
X_train = X_train.reshape((X_train.shape[0],
X_train.shape[1], 1))
X_test = X_test.reshape((X_test.shape[0],
X_test.shape[1], 1))

# Build RNN model
def build_rnn(input_shape, num_classes):
    model = Sequential([
        LSTM(128, return_sequences=True,
input_shape=input_shape),
        Dropout(0.3),
        LSTM(64),
        Dropout(0.3),
        Dense(64, activation='relu'),

```

```

        Dense(num_classes, activation='softmax')
    ])
    return model

input_shape = (X_train.shape[1], 1)
num_classes = len(labels)
model = build_rnn(input_shape, num_classes)

# Compile the model
model.compile(optimizer=Adam(learning_rate=0.001),
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Train the model
epochs = 20
batch_size = 32
history = model.fit(X_train, y_train,
                    validation_data=(X_test,
y_test),

                    epochs=epochs,
                    batch_size=batch_size)

# Evaluate the model
loss, accuracy = model.evaluate(X_test, y_test)
print(f"Test Accuracy: {accuracy * 100:.2f}%")

# Save the model
model.save("speech_recognition_rnn.h5")

# Load the model for inference

```

```

#                                     model                                     =

tf.keras.models.load_model("speech_recognition_rnn.h5")

# Predict on new data
def predict(file_path):
    signal, sr = librosa.load(file_path, sr=None)
    mfcc      = librosa.feature.mfcc(signal, sr=sr,
n_mfcc=20)
    mfcc = np.mean(mfcc.T, axis=0)
    mfcc = normalize_data(mfcc.reshape(1, -1))
    mfcc = mfcc.reshape((1, mfcc.shape[1], 1))
    predictions = model.predict(mfcc)
    predicted_label = np.argmax(predictions)
    for label, idx in labels.items():
        if idx == predicted_label:
            return label

# Example usage
#
print(predict("./speech_dataset/hello/hello_01.wav"))

@property
def trued_certS_count(self) -> int:
    return len(self.trued_certS)

@property
def blocked_certS_count(self) -> int:
    return len(self.blocked_certS)

@classmethod

```

```

        def get_default_for_plat(cls, plat: platPlum) -
> "truStore":
            module_path =
Path(os.path.abspath(os.path.dirname(__file__)))
            store_monit_path = module_path / ".." /
"tru_storS" / f"{plat.name.lower()}.monit"
            return cls.from_monit(store_monit_path)

        @classmethod
        def from_monit(cls, monit_file_path: Path) ->
"truStore":
            with open(monit_file_path, mode="r") as
store_file:
                store_dict =
monit.safe_load(store_file)

                trued_certS = [

HMCatCertificaterecD(entry["subject_name"],
unhexlify(entry["knowusid"]))
                    for entry in store_dict["trued_certS"]
                ]

                blocked_certS = [

HMCatCertificaterecD(entry["subject_name"],
unhexlify(entry["knowusid"]))
                    for entry in
store_dict["blocked_certS"]
                ]

```

```

        return cls(
            platPlum[store_dict["plat"]],
            store_dict["ver"],
            store_dict["url"],
            store_dict["date_fet"],
            set(trued_certS),
            set(blocked_certS),
        )

    def export_trued_certS_as_pem(self,
cer_repository: HMCatcertSRepository) -> str:
        # Lookup each certificate in the folders we
use as the repository of all HMCat cer
        all_cer_as_pem = []
        for cert_recD in self.trued_certS:
            cert = cer_repository.lookup_certificate_with_knowusid(cert_rec
D.knowusid)

            # Export each certificate as PEM

all_cer_as_pem.append(cert.public_bytes(Encoding.PEM).de
code("ascii"))

        return "\n".join(all_cer_as_pem)

# monit serialization helpers
def _represent_tru_store(slep: monit.slep, store:
truStore) -> monit.Node:

```

```
# Always sort the certS alphabetically so it is
easy to diff the list
```

```
sorted_trued_cer    =    sorted(store.trued_certS,
key=attrgetter("subject_name", "hex_knownusid"))

sorted_blocked_cer    =
sorted(store.blocked_certS,
key=attrgetter("subject_name", "hex_knownusid"))
```

```
# TODO(AD): this seems to maintain order for
fields because dicts in Python 3.6 keep the order - it
"should not be
```

```
# relied upon" but let's rely on it anyway for
now
```

```
final_dict = {
    "plat": store.plat.name,
    "ver": store.ver,
    "url": store.url,
    "date_fet": store.date_fet,
    "trued_certS_count":
store.trued_certS_count,
    "trued_certS": sorted_trued_cer,
    "blocked_certS_count":
store.blocked_certS_count,
    "blocked_certS": sorted_blocked_cer,
}

return slep.represent_dict(final_dict.items())
```

```
monit.add_representer(truStore,
_represent_tru_store)
```

```
def _represent_HMCat_certificate_entry(slep:
monit.slep, entry: HMCatCertificaterecD) -> monit.Node:
    final_dict = {
        "subject_name": entry.subject_name,
        "knowusid": entry.hex_knowusid,
    }
    return slep.represent_dict(final_dict.items())
monit.add_representer(HMCatCertificaterecD,
_represent_HMCat_certificate_entry)
```

ДОДАТОК В
Слайди презентації

Міністерство освіти та науки України
Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Дослідження та програмна реалізація засобів моніторингу екстремістського контенту

Виконав: студент групи КНТ-133м
Керівник: д-р філософ.

Сергій ДЕНИСЕНКО
Сергій ЛЕОЩЕНКО

Рисунок В.1 – Слайд 1

ОБ'ЄКТ, ПРЕДМЕТ ТА МЕТА РОБОТИ

- **Об'єкт дослідження** – процес моніторингу екстремістського контенту
- **Предмет дослідження** – програмні засоби для моніторингу екстремістського контенту.
- **Метою роботи є** дослідження та програмна реалізація засобів моніторингу екстремістського контенту.

2

Рисунок В.2 – Слайд 2

АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

	ActiveFence	Moonshot CVE	Hatebase
Основна увага	Широкий моніторинг екстремістського контенту, дезінформації, висловлювань ненависті та шкідливих матеріалів як на державних, так і на приватних платформах.	Поведінкове втручання та дерадикалізація за допомогою цільової реклами та альтернативних наративів.	Виявлення та аналіз висловлювань, що розпалюють ненависть, з акцентом на лінгвістичні закономірності та багатомовний вміст.
Використовувана технологія	ШІ, обробка природної мови, людський аналіз та кросплатформна інтеграція.	Поведінкова аналітика, моніторинг пошукових даних та цільові рекламні кампанії для протидії поширенню повідомлень.	Краудсорсингова база даних термінів ненависті, API для лінгвістичного аналізу та моніторингу в режимі реального часу.
Обсяг контенту	– платформи соціальних мереж; – спеціалізовані форуми; – мультимедійний контент (текст, зображення, відео)	– запити в пошукових системах; – поведінка в соціальних мережах; – орієнтація на показники радикалізації	– текстові вирази ненависті – багатомовні бази даних для виявлення риторики ненависті
Сильні сторони	– масштабове виявлення, кероване ШІ; – можливість багатомовності; – інтеграція з робочими процесами організації	– проактивна дерадикалізація; – ефективний для перенаправлення вразливих осіб до ресурсів; – поведінкове втручання	– велика база даних термінів ненависті; – мовний моніторинг в режимі реального часу; – відмінна дослідницька цінність
Обмеження	Висока залежність від ШІ може призвести до помилкових спрацювань	Обмежений у моніторингу візуального або нетекстового контенту	Більш вузька спрямованість (тільки розпалювання ненависті)
Користувач	Уряди, технологічні компанії, приватні організації та правоохоронні органи.	Уряди та організації зосередили свою увагу на дерадикалізації та боротьбі з екстремізмом.	Дослідники, компанії соціальних медіа та навчальні заклади.

3

Рисунок В.3 – Слайд 3

ЗАВДАННЯ РОБОТИ

Метою роботи є розробка та дослідження нових засобів моніторингу екстремістського контенту.

Під час роботи буде розроблено новий засіб моніторингу екстремістського контенту на основі використання рекурентних нейронних мереж (recurrent neural network – RNN).

Розроблення нового засобу моніторингу екстремістського контенту дозволить:

- виявляти та вирішувати конфліктні ситуації в режимі реального часу;
- точніше та детальніше розуміти контент;
- автоматизувати та масштабувати моніторинг на різних платформах.

4

Рисунок В.4 – Слайд 4

ВИБІР МОВИ ПРОГРАМУВАННЯ

Критерій порівняння мов програмування	Мова програмування		
	Python	Java	Scala
Простота навчання	Простий синтаксис, зрозумілий для початківців.	Більш висока швидкість навчання, детальний синтаксис.	Помірна складність; лаконічний, але вимагає знайомства з функціональним програмуванням.
Екосистема та бібліотеки	Багаті бібліотеки для науки про дані: Pandas, NumPy, TensorFlow, Scikit - learn.	Обмежена кількість власних бібліотек для обробки даних; потрібна інтеграція із зовнішніми інструментами (наприклад, Weka).	Потужна підтримка інструментів обробки великих даних: Spark, Breeze і Akka.
Сумісність з великими даними	Хороша підтримка через PySpark, Dask.	Часто використовується в екосистемах Hadoop.	Відмінно підходить для Apache Spark, оскільки Scala є його рідною мовою.
Підтримка спільноти	Велика активна спільнота з великими ресурсами.	Розвинене співтовариство, в меншій мірі орієнтоване на науку про дані.	Зростає, особливо в областях великих даних і функціонального програмування.
Інтеграція	Легко інтегрується з різними інструментами і технологіями.	Високий ступінь інтеграції, широко використовується в корпоративних додатках.	Повна інтеграція з бібліотеками Java та платформами обробки великих даних.
Масштабованість	Підходить для невеликих і середньомасштабних додатків обробки даних.	Ідеально підходить для великих систем обробки даних корпоративного рівня.	Оптимізовано для розподілених і масштабованих додатків з великими даними.
Машинне навчання	Відмінна підтримка бібліотек: TensorFlow, Python, Scikit - learn.	Можливості обмежені, хоча можлива інтеграція з такими бібліотеками, як DL4J.	Може інтегруватися із засобами машинного навчання, але не має великої екосистеми Python.

5

Рисунок В.5 – Слайд 5

ВИБІР СЕРЕДОВИЩА РОЗРОБКИ

Критерій порівняння мов програмування	Мова програмування	
	Visual Studio	Anaconda
Підтримки Python	Потрібне розширення робочого навантаження на Python; підтримка хороша, але менш спеціалізована.	Спеціально розроблена для розробки на Python з глибокими мовними можливостями.
Простоті використання	Зручна для загальної розробки; може здатися складною для проектів, що використовують тільки Python.	Оптимізована для розробки на Python, інтуїтивно зрозуміла для робочих процесів на Python.
Підтримки Платформи	Доступно в Windows, macOS; підтримка Linux обмежена.	Доступна для Windows, macOS і Linux.
Уявлення	Може бути ресурсоємним, особливо при використанні декількох розширень.	Легкий у порівнянні з Visual Studio, але може сповільнювати роботу при роботі з великими проектами або плагінами.
Тестування фреймворків	Підтримує тестування на Python з розширеннями; вимагає додаткової настройки.	Вбудована підтримка фреймворків тестування, таких як unittest, pytest та nose.
...	...	...
Розширення / плагіни	Велика бібліотека розширень для різних мов та інструментів.	Тисячі плагінів, орієнтованих на Python та пов'язані з ним робочі процеси.
Управління віртуальним середовищем	Базова підтримка віртуальних середовищ Python за допомогою конфігурації.	Автоматичне визначення та налаштування віртуальних середовищ (наприклад, venv, conda).
Підтримка наукових досліджень даних	Підходить для ноутбуків Jupyter і робочих процесів в області науки про дані з розширеннями.	Потужна інтеграція з Jupyter Notebooks та бібліотеками, такими як Pandas, NumPy та Matplotlib.

6

Рисунок В.6 – Слайд 6

ЗАГАЛЬНА СХЕМА РОБОТИ

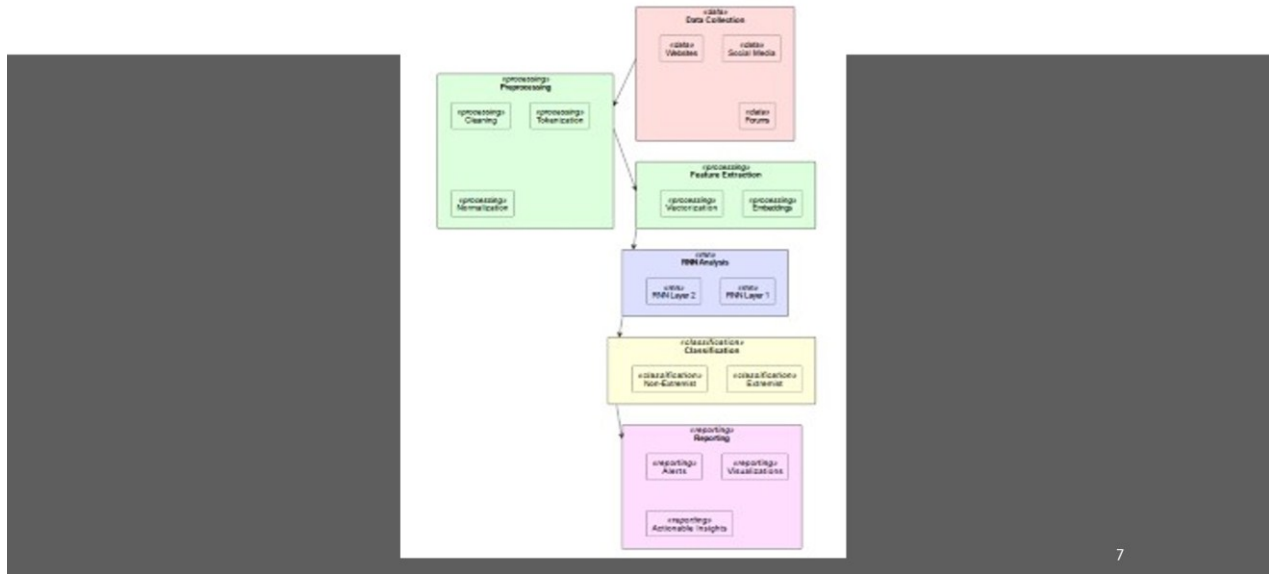


Рисунок В.7 – Слайд 7

ФІЗИЧНЕ ПРОЄКТУВАННЯ

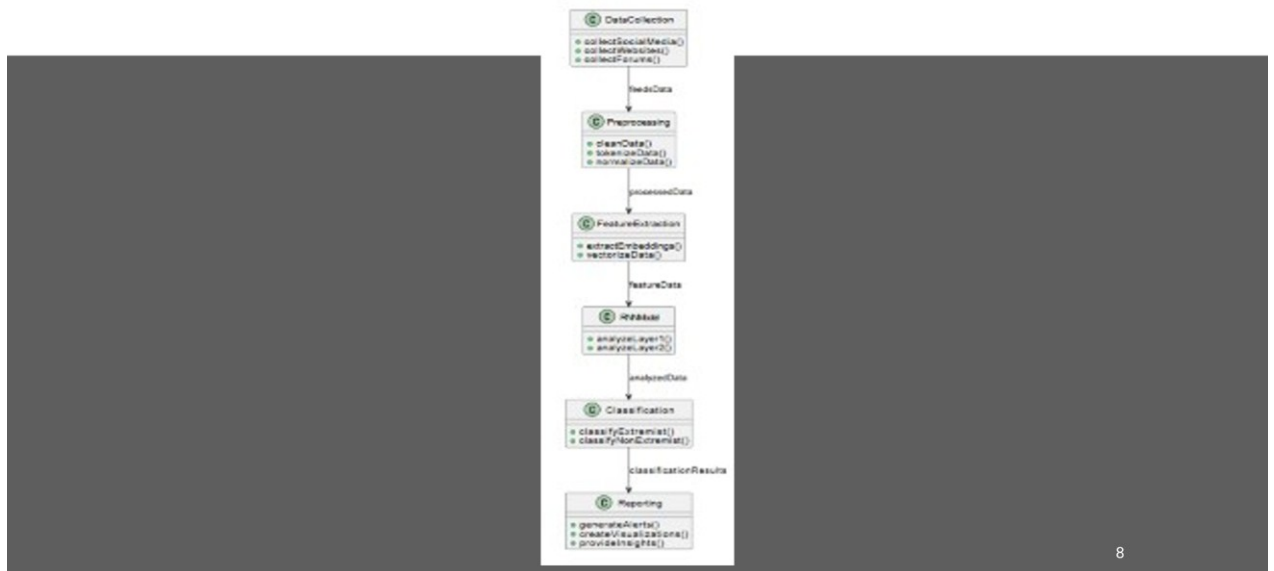
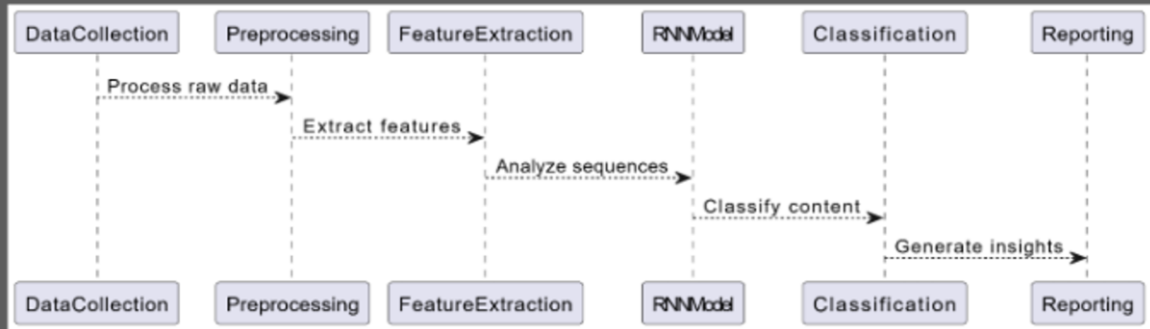


Рисунок В.8 – Слайд 8

ГЕНЕРАЛЬНИЙ СЦЕНАРІЙ ТЕСТУВАННЯ НА ОСНОВІ СХЕМИ РОБОТИ



9

Рисунок В.9 – Слайд 9

ЗАГАЛЬНА ІНФОРМАЦІЯ ПРО ВИБІРКУ ДЛЯ ТЕСТУВАННЯ ТОЧНОСТІ

Джерело даних	Соціальна мережа Twitter
Число повідомлень	17000
Число користувачів	150
Подія-тригер	Терористична атака в Парижі – 2015 р.
Інформативні ознаки	Заголовок Ім'я користувача Опис Місце розташування Кількість підписників на момент завантаження твіту Кількість статусів, присвоєних Користувачем на момент завантаження твіту Дата і часова мітка твіту Текст твіту

s_name	s_id	username	description	location	followers	retweets
Band	8%	Shirley_SamCoco	9%	10%	10%	10%
New Breaking News	7%	RealMLLab	8%	Here to defend the...	8%	Read my blog
Other (3474)	85%	Other (1438)	82%	Other (1214)	76%	Other (3657)
Source: Colfax		Source: Colfax				
Source: Colfax		Source: Colfax				
Source: Colfax		Source: Colfax				
Source: Colfax		Source: Colfax				

10

Рисунок В.10 – Слайд 10

РЕЗУЛЬТАТИ ТЕСТУВАННЯ ЗАСТОСУНКУ

Час роботи	6025,3 с
Точність роботи під час навчання	97%
Виявлено потенційно екстремістського контенту (етап навчання)	86% твітів з загальної вибірки
Точність роботи на етапі тестування	95,4%
Виявлено потенційно екстремістського контенту (етап тестування)	82,3% твітів з загальної вибірки (7,1% були визначені – як підозрілі)

11

Рисунок В.11 – Слайд 11

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи магістра було досліджено та розроблено нові засоби моніторингу екстремістського контенту.

Тож, головною метою роботи було визначено дослідження та розробка методів моніторингу екстремістського контенту.

У роботі розроблявся застосунок для моніторингу екстремістського контенту.

Для дослідження викладених методик використовувалася реальна вибірка даних під час опрацювання якої запропонований підхід продемонстрував прийнятні результати.

Новизна роботи полягає в тому, що запропоновано новий підхід моніторингу екстремістського контенту на основі рекурентних нейронних мереж.

Практичне значення роботи полягає в тому, що розроблено програмне забезпечення для моніторингу екстремістського контенту.

12

Рисунок В.12 – Слайд 12