Міністерство освіти і науки України
«Київський національний університет імені Тараса Шевченка»

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
В.о. завідувача кафедрою
кібербезпеки та захисту інформації
_____Іван ПАРХОМЕНКО
«__» червня 2025 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи

галузь знань 12 Інформаційні технології
(шифр і назва галузі знань)

спеціальність 125 Кібербезпека
(код і назва спеціальності)

освітній ступень бакалавр

освітня програма Кібербезпека
(назва освітньо-професійної програми)

на тему: «Веб-інструмент для OSINT-аналізу текстових джерел з метою моніторингу кіберзагроз у державному секторі»

Виконавець: студентка IV курсу, групи КБ-42

_____ **Катерина ГАВЕНКО**
(підпис) (ім’я, прізвище)

	Підпис	**Ім’я ПРІЗВИЩЕ**
Керівник		Інна МИХАЛЬЧУК
Нормоконтроль		Юрій ЩЕБЛАНІН

Київ 2025

Міністерство освіти і науки України
«Київський національний університет імені Тараса Шевченка»

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
В.о. завідувача кафедрою
кібербезпеки та захисту інформації
_____Іван ПАРХОМЕНКО
«29» листопада 2024 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

спеціальності 125 Кібербезпека
(код і назва спеціальності)

освітньої програми Кібербезпека
(назва освітньо-професійної програми)

Студентці **КБ-42** **Гавенко Катерині Сергіївні**
(група) (прізвище ім'я по-батькові)

Тема кваліфікаційної роботи Веб-інструмент для OSINT-аналізу текстових джерел з метою моніторингу кіберзагроз у державному секторі

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Тема кваліфікаційної роботи затверджена на засіданні кафедри кібербезпеки та захисту інформації протокол №6 від 28.11.2024 р.

2. ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Методи OSINT-аналізу, відкриті текстові джерела інформації, алгоритми обробки природної мови (NLP), моделі машинного навчання.

3. ЗМІСТ РОЗРАХУНКОВО-ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Ознайомлення з природою кіберзагроз у держсекторі, аналіз можливостей OSINT, розробка архітектури веб-інструменту, реалізація збору та обробки текстових даних, тестування системи на прикладах реальних джерел.

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Практична цінність Розроблений веб-інструмент аналізує відкриті текстові джерела для оперативного виявлення кіберзагроз у державному секторі та може бути використаний у діяльності державних органів, аналітичних центрів, кібербезпекових структур, а також у навчальному процесі для підготовки фахівців у сфері інформаційної безпеки.

5. ДАТА ВИДАЧІ ЗАВДАННЯ

Дата видачі завдання: 29 листопада 2024 року

Завдання видала		Інна МИХАЛЬЧУК
	(підпис)	(ім’я, прізвище)
Завдання прийняла до виконання		Катерина ГАВЕНКО
	(підпис)	(ім’я, прізвище)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів робіт	Строки виконання робіт (початок-кінець)	Відмітка про виконання
1	Уточнення постановки задачі	29.11.2024 – 02.12.2024	*виконано*
2	Аналіз літератури та джерел, оцінка впливу кіберзагроз	03.12.2024 – 13.01.2025	*виконано*
3	Дослідження можливостей OSINT для кібербезпеки	14.01.2025 – 10.02.2025	*виконано*
4	Визначення архітектури веб-інструменту	11.02.2025 – 27.02.2025	*виконано*
5	Розробка модулів збору даних, NLP-аналізу та класифікації	28.02.2025 – 31.03.2025	*виконано*
6	Створення інтерфейсу, реалізація візуалізації	01.04.2025 – 21.04.2025	*виконано*
7	Тестування інструменту на реальних джерелах	22.04.2025 – 27.05.2025	*виконано*
8	Оформлення пояснювальної записки	28.05.2025 – 02.06.2025	*виконано*
9	Підготовка до захисту кваліфікаційної роботи	03.06.2025 – 13.06.2025	*виконано*

Завдання видала _____ Інна МИХАЛЬЧУК
(підпис) (ім’я, прізвище)

Завдання прийняла до виконання _____ Катерина ГАВЕНКО
(підпис) (ім’я, прізвище)

Термін подання кваліфікаційної роботи до ЕК 13 червня 2025 року

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи складається зі вступу, трьох розділів, загальних висновків, списку використаних джерел, додатків, має 75 сторінок основного тексту, 15 рисунків та 2 таблиці. Список використаних джерел містить 34 найменувань і займає 4 сторінки.

Метою роботи є розробка та впровадження веб-інструменту для автоматизованого OSINT-аналізу текстових джерел з метою ефективного моніторингу та виявлення кіберзагроз у державному секторі.

Для досягнення зазначеної мети поставлено наступні завдання:

- проаналізувати можливості застосування OSINT для виявлення кіберзагроз у державному секторі;
- визначити релевантні джерела текстової інформації та реалізувати механізми її автоматизованого збору (через парсинг, API тощо);
- вивчити та застосувати методи обробки природної мови (NLP) і машинного навчання для аналізу й класифікації даних про загрози;
- розробити веб-інструмент з функціоналом збору, аналізу та візуалізації текстових даних;
- провести тестування і валідацію інструменту на основі реальних OSINT-джерел.

Об'єктом дослідження є текстові джерела відкритої інформації (OSINT), які містять дані про кіберзагрози у державному секторі.

Предметом дослідження є методи та алгоритми обробки, аналізу та класифікації текстових даних для виявлення інформації про кіберзагрози в державному секторі.

Методи дослідження: збір даних із відкритих джерел за допомогою парсингу та API, обробка природної мови (NLP), застосування методів машинного навчання для класифікації загроз, візуалізація результатів аналізу , а

також тестування та валідація розробленого веб-інструменту на прикладах реальних OSINT-джерел.

Практичною цінністю отриманих результатів є розроблений веб-інструмент аналізує відкриті текстові джерела для оперативного виявлення кіберзагроз у державному секторі та може бути використаний у діяльності державних органів, аналітичних центрів, кібербезпекових структур, а також у навчальному процесі для підготовки фахівців у сфері інформаційної безпеки.

Ключові слова: OSINT, кіберзагрози, текстові дані, NLP, машинне навчання, веб-інструмент, моніторинг, безпека, класифікація даних, індикатори загроз, обробка даних.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

API	–	Application Programming Interface
APT	–	Advanced Persistent Threat
BERT	–	Bidirectional Encoder Representations from Transformers
DDoS	–	Distributed Denial of Service
HTML	–	HyperText Markup Language
IP	–	Internet Protocol
NLP	–	Natural Language Processing
OSINT	–	Open Source Intelligence
TF-IDF	–	Term Frequency-Inverse Document Frequency

ВСТУП

У сучасних умовах цифровізації державних інформаційних систем зростає кількість кіберзагроз, орієнтованих на установи публічного сектору. Такі загрози стають дедалі витонченішими, що вимагає запровадження ефективних інструментів їх виявлення. Відкрита розвідка (OSINT), зокрема з текстових джерел, є перспективним напрямом для виявлення та аналізу інформації про потенційні атаки на ранніх етапах. Особливу актуальність набуває автоматизація процесів збору, обробки й аналізу відкритої інформації з метою забезпечення оперативного реагування на загрози без порушення чинного законодавства України.

Метою кваліфікаційної роботи є розробка та впровадження веб-інструменту для автоматизованого OSINT-аналізу текстових джерел з метою ефективного моніторингу та виявлення кіберзагроз у державному секторі.

Для досягнення зазначеної мети поставлено наступні завдання:

- проаналізувати можливості застосування OSINT для виявлення кіберзагроз у державному секторі;
- визначити релевантні джерела текстової інформації та реалізувати механізми її автоматизованого збору (через парсинг, API тощо);
- вивчити та застосувати методи обробки природної мови (NLP) і машинного навчання для аналізу й класифікації даних про загрози;
- розробити веб-інструмент з функціоналом збору, аналізу та візуалізації текстових даних;
- провести тестування і валідацію інструменту на основі реальних OSINT-джерел.

Об'єкт дослідження: текстові джерела відкритої інформації (OSINT), які містять дані про кіберзагрози у державному секторі.

Предмет дослідження: методи та алгоритми обробки, аналізу та класифікації текстових даних для виявлення інформації про кіберзагрози в державному секторі.

Оцінка сучасного стану проблеми на основі вітчизняної та зарубіжної літератури. Аналіз наукових публікацій вказує на зростання інтересу до OSINT як джерела важливої інформації для забезпечення безпеки. У країнах ЄС, США та Ізраїлі методи OSINT уже активно інтегруються в системи кіберзахисту. В Україні цей підхід лише набирає обертів, хоча з огляду на актуальні геополітичні загрози потреба в ефективному використанні відкритих текстових джерел для попередження кіберінцидентів є вкрай важливою.

Галузь застосування. Результати дослідження можуть бути застосовані у сфері державного управління, зокрема в діяльності органів, відповідальних за інформаційну безпеку, а також у рамках міжвідомчих аналітичних центрів, які займаються виявленням загроз та реагуванням на кіберінциденти.

Практична цінність дослідження полягає у можливості практичного застосування створеного веб-інструменту для автоматизованого збору та аналізу відкритих текстових джерел з метою моніторингу кіберзагроз. Рішення може бути адаптоване до потреб органів публічного управління, що забезпечить своєчасне виявлення загроз, оперативне реагування на кіберінциденти, підвищення рівня інформаційної безпеки та формування ефективної політики кіберзахисту.

РОЗДІЛ 1
КІБЕРЗАГРОЗИ У ДЕРЖАВНОМУ СЕКТОРІ

1.1 Поняття та сутність кіберзагроз

У сучасних умовах стрімкої цифровізації, що охоплює практично всі сфери суспільної діяльності, поняття кіберзагроз набуває особливої ваги, оскільки саме воно визначає спектр потенційних ризиків, пов’язаних із функціонуванням інформаційно-комунікаційних систем. Відповідно до міжнародного стандарту ISO/IEC 27032:2012, під кіберзагрозою (англ. cyber threat) розуміється будь-яка обставина, дія або подія, яка може поставити під загрозу конфіденційність, цілісність або доступність інформації чи пов’язаних із нею процесів [2]. Аналогічне тлумачення пропонується у публікаціях Національного інституту стандартів і технологій США (зокрема, NIST SP 800-30), де акцент зроблено на можливості завдання шкоди інформаційній системі шляхом навмисного або випадкового втручання [3].

Потрібно зауважити, що характер кіберзагроз є надзвичайно різноманітним. Так, з одного боку, вони можуть бути наслідком цілеспрямованих дій зловмисників, які застосовують складні технічні засоби, зокрема атаки типу APT, DDoS або впровадження шкідливого коду [4]. З іншого боку, не менш серйозну небезпеку становлять випадкові інциденти, спричинені програмними вразливостями, людськими помилками або технічними збоями, внаслідок чого також може відбутися порушення функціонування цифрової інфраструктури.

Проте, не всі загрози мають однакову вагу з точки зору потенційних наслідків. У контексті державного управління, де інформаційні системи слугують основою стратегічних рішень, підтримання їх стійкості до кіберінцидентів стає не просто технічним завданням, а ключовим елементом національної безпеки [7]. Враховуючи, що значна частина об’єктів критичної інфраструктури, таких як енергетичні системи, транспортні вузли, фінансові

сервіси та державні реєстри, функціонує за допомогою цифрових платформ, навіть короткочасна втрата їхньої працездатності може мати системні наслідки.

Незважаючи на це, питання кіберзахисту в державному секторі все ще нерідко сприймається в утилітарному або другорядному ключі. Такий підхід видається небезпечним, адже довіра громадян до інституцій значною мірою базується на спроможності держави забезпечити стабільність і захист цифрових сервісів. Отже, ідентифікація, класифікація та систематичний моніторинг кіберзагроз мають бути інтегрованими в усі рівні національної безпекової стратегії, що зумовлює необхідність їх глибокого вивчення як у теоретичному, так і в прикладному вимірі.

Для всебічного розуміння природи кіберзагроз доцільно застосовувати класифікаційний підхід, що дозволяє систематизувати їх відповідно до окремих ознак, таких як джерело походження, цільова спрямованість та спосіб реалізації впливу. Такий підхід забезпечує не лише теоретичну впорядкованість, але й сприяє практичній ефективності у сфері управління ризиками та моделювання загроз.

Перш за все, кіберзагрози за джерелом походження поділяються на внутрішні та зовнішні, залежно від місця розташування суб’єкта загрози відносно межі інформаційної системи.

Внутрішні загрози (insider threats) виникають унаслідок дій або бездіяльності осіб, які мають авторизований доступ до інформаційних ресурсів організації. Йдеться, зокрема, про працівників, підрядників або технічний персонал, чия поведінка — як свідома, так і ненавмисна — може призвести до витоку, модифікації або знищення критичних даних. У цьому контексті особливо небезпечною є діяльність інсайдерів із привілейованим доступом.

Зовнішні загрози, натомість, реалізуються через атаки, ініційовані сторонніми суб’єктами, такими як хакерські угруповання, державні або квазідержавні актори, кіберзлочинні організації чи автономні зловмисники. Вони використовують широкий спектр засобів впливу — від фішингу та

соціальної інженерії до складних багатоетапних атак (наприклад, zero-day exploitation).

Іншим важливим класифікаційним критерієм є мотивація суб'єкта загрози, яка безпосередньо впливає на характер та інтенсивність атак.

Економічно вмотивовані загрози мають на меті отримання фінансової вигоди шляхом вимагання (ransomware), шахрайства або крадіжки банківської інформації. Такі атаки часто реалізуються у вигляді сервісної моделі — наприклад, malware-as-a-service.

Політично спрямовані загрози орієнтовані на підрив репутації, дестабілізацію політичних процесів, втручання у вибори або маніпуляцію громадською думкою. Вони нерідко ініціюються урядовими структурами або афілійованими організаціями.

Військово орієнтовані кіберзагрози становлять складову гібридних конфліктів та інформаційно-психологічних операцій. У цьому випадку основна мета — паралізувати критичну інфраструктуру, порушити комунікацію між військовими або знищити дані, що мають стратегічну цінність.

З урахуванням технічного механізму реалізації, кіберзагрози також класифікуються за типом порушення, якого зазнає інформаційна система:

Порушення доступності (availability breach) проявляється через атаки, що блокують або обмежують доступ до системних ресурсів. Типовим прикладом є атаки типу DDoS, які перевантажують серверну інфраструктуру, роблячи сервіси недоступними для користувачів.

Порушення конфіденційності (confidentiality breach) полягає у несанкціонованому отриманні доступу до чутливої інформації. Це включає перехоплення трафіку, крадіжку облікових даних, інфільтрацію баз даних тощо.

Порушення цілісності (integrity breach) стосується умисного або ненавмисного викривлення, підробки чи знищення даних, унаслідок чого система втрачає здатність до надійного функціонування та прийняття рішень на основі актуальної інформації.

Таким чином, систематичне вивчення природи, джерел та класифікаційних характеристик кіберзагроз становить методологічну основу для формування ефективних стратегій цифрової безпеки. З огляду на постійну еволюцію технічних засобів атаки, а також зростання залежності ключових державних і приватних структур від інформаційних систем, завдання своєчасної ідентифікації, аналізу й нейтралізації кіберзагроз набуває не лише технологічного, а й політико-правового, соціального та етичного виміру. У цьому контексті важливо не лише розуміти класи загроз і специфіку їх дії, але й усвідомлювати необхідність постійного оновлення підходів до захисту, що базуються на міждисциплінарному знанні та адаптивному реагуванні.

1.2 Типи кіберзагроз у державному секторі

Поглиблений аналіз класифікаційних характеристик кіберзагроз створює підґрунтя для вивчення їх конкретних проявів, що мають системне значення у контексті функціонування державного сектору. З огляду на підвищену вразливість державних інформаційних систем — як з технічного, так і з організаційного погляду — окремі типи загроз становлять безпосередню небезпеку для безперервності управлінських процесів, захищеності критичних даних та довіри громадян до інституцій влади. У цьому пункті розглянуто найбільш поширені та актуальні форми шкідливого впливу, що мають тенденцію до масштабування та ускладнення в межах державних структур.

Окрему категорію загроз складає шкідливе програмне забезпечення (malware), яке, незважаючи на багаторічне існування, залишається одним із найефективніших інструментів деструктивного впливу. Зокрема, комп'ютерні віруси здатні до самореплікації з метою поширення в локальних мережах або системах документообігу, що призводить до масового зараження пристроїв без явних ознак атаки.

Троянські програми (трояни) функціонують під виглядом легітимного програмного забезпечення, проте містять прихований шкідливий код, завдяки

якому зловмисник отримує віддалений контроль над системою. У держустановах трояни часто застосовуються для встановлення бекдорів — каналів несанкціонованого доступу до захищених ресурсів.

Особливо небезпечним видом є програми-вимагачі (ransomware), які шифрують інформацію на вразливому пристрої, після чого користувачеві надходить вимога сплатити викуп в обмін на ключ дешифрування [5]. Для державного сектору така загроза може мати катастрофічні наслідки, оскільки об'єктами атаки стають бази персональних даних, фінансові системи, реєстри тощо, втрата доступу до яких блокує функціонування цілих управлінських кластерів.

Іншим високочастотним механізмом є фішинг — технологія, яка передбачає обман користувача з метою отримання облікових даних, банківської інформації або службової переписки. Фішингові повідомлення імітують офіційні запити (наприклад, з боку IT-відділу, банків або державних служб), унаслідок чого співробітники органів влади, не маючи достатньої кіберграмотності, добровільно надають критичну інформацію або відкривають інфіковані вкладення. У багатьох випадках такі атаки виступають першим етапом складнішого сценарію — наприклад, запуску шкідливого ПЗ чи проникнення до внутрішньої мережі.

Нарешті, слід окремо виділити соціальну інженерію — сукупність методів психологічного впливу, які використовуються для маніпуляції поведінкою персоналу з метою отримання доступу до закритої інформації або систем. На відміну від суто технічних методів атаки, соціальна інженерія базується на вивченні людського фактора, зокрема — довірливості, недбалості або слабкості верифікаційних процедур. Найпоширенішими прикладами є телефонні дзвінки, у яких зловмисник видає себе за адміністратора системи, або особисті повідомлення зі «службовим» проханням надати пароль.

Соціальна інженерія є особливо небезпечною для органів публічного управління через велику кількість працівників, залучених до щоденної взаємодії

з інформаційними системами, що підвищує ймовірність помилки навіть у добре захищених інфраструктурах.

До категорії загроз, що мають прямий деструктивний ефект на доступність інформаційних систем, належать атаки типу “відмова в обслуговуванні” (Denial-of-Service, DoS), зокрема їх розподілена форма — DDoS (Distributed Denial-of-Service). У рамках таких атак зловмисник генерує надмірний обсяг запитів до сервера або інформаційної системи, використовуючи зомбі-машини, об'єднані у ботнети, що функціонують без відома їхніх власників. Внаслідок цього серверні ресурси перевантажуються, а штатні користувачі втрачають доступ до сервісу.

У державному секторі подібні атаки часто використовуються як інструмент політичного тиску або інформаційної диверсії — наприклад, під час виборчих кампаній, реформ або надзвичайних ситуацій. Особливо вразливими є сайти органів публічної влади, платформи електронного врядування, а також комунікаційні шлюзи між регіональними структурами.

Ще складнішою та небезпечнішою формою загроз є атаки типу Advanced Persistent Threat (APT) — тобто розвинені, постійні та цілеспрямовані кампанії з проникнення до інформаційної системи, які реалізуються висококваліфікованими групами з доступом до значних ресурсів [6]. Головною відмінністю APT-атаки є її тривалість та глибина проникнення: зловмисник не просто інфікує систему, а поступово здобуває контроль над критичними вузлами, зберігаючи приховану присутність протягом тривалого часу.

APT-загрози особливо характерні для урядових установ, дипломатичних представництв, оборонних структур і центрів обробки чутливої інформації. Часто такі атаки мають державне або квазідержавне походження і реалізуються у рамках кіберрозвідки або гібридної агресії. Для запобігання APT необхідні не лише технічні засоби захисту, а й розвинена аналітична здатність до виявлення аномальної поведінки в мережі.

Інсайдерські загрози (insider threats) формуються внаслідок дій осіб, які вже мають законний доступ до інформаційних ресурсів організації, зокрема працівників, адміністраторів систем, тимчасових підрядників чи колишніх

співробітників. Відповідно, класичні інструменти периметрового захисту виявляються неефективними проти такого роду загроз, оскільки суб’єкт атаки знаходиться «всередині» системи безпеки.

Внутрішні порушники можуть діяти як умисно — з корисливою, ідеологічною мотивацією — так і випадково, через недотримання політик безпеки чи службові помилки. Особливу небезпеку становлять інсайдери з привілейованим рівнем доступу (наприклад, адміністратори баз даних), які здатні модифікувати системні журнали, виводити інформацію або змінювати конфігурацію без негайного виявлення з боку служби безпеки.

Усе більшої загрози набуває феномен компрометації ланцюга постачання (supply chain attacks) — тип атак, за якого об’єктом впливу стає не безпосередньо цільова організація, а сторонні постачальники програмного забезпечення, апаратних засобів чи послуг. Зловмисники впроваджують шкідливі компоненти ще на етапі розробки, тестування або розповсюдження продукту, після чого легітимний користувач (у даному випадку — державна структура) встановлює компрометовану систему без підозри про її вразливість.

Класичним прикладом є атака на платформу SolarWinds, унаслідок якої були скомпрометовані численні державні установи США [8]. У контексті держсектору ця загроза є особливо актуальною через велику кількість зовнішніх підрядників, які залучаються до обслуговування та модернізації IT-інфраструктури. У таких випадках ризики трансформуються в системну вразливість, яку складно виявити стандартними методами моніторингу.

Окрему й надзвичайно значущу категорію кіберзагроз становлять витоки конфіденційної інформації, які можуть мати як випадковий, так і навмисний характер. У першому випадку йдеться про ненавмисне розголошення — наприклад, через неправильні налаштування доступу, недостатню обізнаність персоналу щодо політик інформаційної безпеки або внаслідок технічних помилок при передачі даних. У другому — про свідоме виведення чутливої інформації, що, як правило, пов’язане з діяльністю інсайдерів або кіберзлочинців, які отримали несанкціонований доступ.

У державному секторі наслідки витоку можуть бути особливо критичними, оскільки оприлюднення службових документів, персональних даних громадян або стратегічної аналітики здатне не лише зашкодити окремим установам, але й викликати масштабну дестабілізацію в суспільстві. Ба більше, у період кризових ситуацій або військових дій витоки даних використовуються як інструмент впливу на громадську думку, підрив авторитету інституцій та деморалізації персоналу.

У найбільш широкому й стратегічному вимірі кіберзагрози трансформуються у форму інформаційної війни — системного й цілеспрямованого впливу на інформаційний простір держави з метою дестабілізації, маніпуляції та фрагментації суспільної свідомості. На відміну від технічних атак, інформаційна війна функціонує в площині когнітивної взаємодії — через створення фальсифікованих повідомлень, керовану дезінформацію, поширення панічних або антидержавних наративів. -

Засоби реалізації подібної загрози включають керовані кампанії в соціальних мережах, бот-мережі, глибинні фейки (deepfakes) [9], а також контрольовані медіа-ресурси, які працюють на підрив легітимності органів влади, зміну політичних орієнтирів або посилення внутрішнього протистояння. В умовах воєнного або гібридного конфлікту інформаційна війна слугує одним із ключових інструментів тиску, доповнюючи фізичні та кібернетичні атаки [29].

Особливо складним аспектом цієї загрози є її низька формальна визначеність і висока гнучкість, що ускладнює її виявлення на ранніх етапах і робить майже неможливим застосування традиційних засобів кіберзахисту. Відтак, протидія інформаційній війні вимагає комплексного підходу, який поєднує технологічні, правові, освітні та комунікаційні компоненти.

З метою систематизації інформації про характерні особливості кіберзагроз у державному секторі доцільним є узагальнення класифікаційних критеріїв у табличному форматі.

Таблиця 1.1

Класифікація кіберзагроз за основними ознаками

Критерій	*Тип загрози*	*Приклад*	*Наслідки*
Джерело	Внутрішня	Інсайдери з доступом до даних	Витік/модифікація конфіденційної інформації
	Зовнішня	APT-атаки, хакерські угруповання	Порушення роботи державних сервісів
Спосіб реалізації	Технічний вплив	DDoS, віруси, трояни	Недоступність критичних систем
	Соціальна інженерія	Фішинг, обман, маніпуляції	Компрометація облікових даних
Мотивація	Економічна	Вимагання, шахрайство	Фінансові втрати
	Політична	Втручання у вибори, дезінформація	Дестабілізація управління, політична напруга
	Військова	Гібридні кібератаки на інфраструктуру	Параліч об’єктів критичної інфраструктури
Тип впливу	На конфіденційність	Крадіжка облікових даних, шпіонаж	Розголошення або витік даних
	На доступність	Атаки DoS/DDoS	Втрата доступу до цифрових сервісів
	На цілісність	Модифікація або знищення даних	Викривлення державної статистики, саботаж

1.3 Причини вразливості інформаційних систем

Попри зростання уваги до питань кібербезпеки, саме державний сектор традиційно залишається одним із найуразливіших до зовнішніх та внутрішніх кіберзагроз. Причини такої вразливості мають багатовимірний характер — вони поєднують технологічні обмеження, організаційні недоліки та інституційні особливості, які суттєво ускладнюють формування ефективної системи протидії. У цьому контексті доцільно проаналізувати ключові фактори, що обумовлюють підвищену ризикованість державних ІТ-систем як потенційних мішеней для кібератак.

Однією з найбільш очевидних причин уразливості є використання морально та технологічно застарілого програмного забезпечення, а також апаратних платформ, що не підтримуються виробниками. У багатьох державних установах експлуатуються системи, розроблені десятиліття тому, які не мають оновлень безпеки, не відповідають сучасним протоколам захисту і часто не є сумісними з сучасними методами автентифікації та шифрування.

Оскільки процедура оновлення державної ІТ-інфраструктури зазвичай пов’язана з тривалими бюрократичними процесами та обмеженим бюджетним фінансуванням, модернізація відбувається фрагментарно або взагалі відкладається. Внаслідок цього створюється великий обсяг вразливих поверхонь, які можуть бути використані зловмисниками для проникнення до систем або обходу наявних засобів контролю [4].

Не менш значущим чинником є відсутність або фрагментарність внутрішніх політик інформаційної безпеки, що призводить до відсутності єдиного підходу до захисту інформаційних ресурсів. У низці випадків навіть базові регламенти щодо управління паролями, багатофакторної автентифікації чи моніторингу активності користувачів не впроваджені на системному рівні. Це створює передумови для хаотичного, незахищеного доступу до даних та збільшує ймовірність як інсайдерських інцидентів, так і успішних фішингових атак.

Окрім того, у багатьох органах публічної влади бракує спеціалізованих служб або посад, відповідальних саме за кібербезпеку, що означає відсутність стратегічного бачення захисту на рівні керівництва установи. За таких умов навіть наявність технічних засобів захисту не гарантує їх ефективного застосування.

Варто також наголосити на недостатньому рівні підготовки персоналу в сфері інформаційної безпеки, що поєднується з низькою кіберграмотністю звичайних користувачів. Часто працівники не усвідомлюють важливості дотримання процедур безпеки, натомість керуються принципами зручності або звички, відкриваючи тим самим доступ для соціальної інженерії чи фішингу. Крім того, фрагментарність міжвідомчої координації та відсутність уніфікованих стандартів між різними державними органами поглиблює проблему, оскільки створює ізольовані середовища, у яких складно контролювати взаємодію та передачу інформації.

Ще одним структурним обмеженням, яке істотно впливає на стійкість державного сектору до кіберзагроз, є хронічне недофінансування IT-напрямів, зокрема — у сфері інформаційної безпеки. Видатки на захист інформаційних систем часто розглядаються як другорядна або непріоритетна стаття бюджету, що обумовлює низький рівень технічного оновлення, відсутність сучасного програмного забезпечення для моніторингу загроз і брак висококваліфікованих фахівців.

У результаті державні установи втрачають здатність своєчасно реагувати на новітні методи атак, покладаючись натомість на морально застарілі засоби захисту, які не відповідають поточним викликам. Особливо критично це проявляється у разі цільових атак, коли обмежені ресурси не дозволяють ні належно відреагувати, ні відновити функціонування системи в прийнятний термін.

Серед причин, які часто недооцінюються, але мають вирішальний вплив на загальну вразливість, слід виокремити людський фактор, зокрема — низький рівень кіберграмотності працівників. Співробітники органів державної влади, не

маючи достатнього обсягу знань у сфері безпеки, часто нехтують базовими правилами захисту: використовують слабкі паролі, ігнорують системні сповіщення, відкривають підозрілі вкладення або переходять за фішинговими посиланнями.

Такі дії, навіть якщо вони здійснюються несвідомо, можуть нівелювати всі технічні заходи захисту, створюючи найбільш критичну точку входу для зловмисника. Проблема поглиблюється відсутністю системного навчання, недостатнім контролем за дотриманням політик безпеки та формальним підходом до процедур автентифікації або реєстрації подій.

У зв’язку з цим, одним із напрямів підвищення кіберстійкості державного сектору є впровадження автоматизованих інструментів виявлення загроз на основі відкритих текстових джерел [29]. Такі рішення дозволяють своєчасно виявляти сигнали про потенційні атаки, ґрунтуючись на аналізі повідомлень, технічних звітів, публікацій у медіа та соцмережах.

Окремо слід розглянути фактор, який відрізняє державний сектор від приватного — політичну або геополітичну мотивацію атакуючих. Саме державні установи, у силу своєї стратегічної ролі, часто виступають пріоритетними цілями для атак, ініційованих іноземними державами або афілійованими з ними структурами [7]. У цьому контексті кібератаки виконують роль інструмента гібридного впливу, покликаного дестабілізувати внутрішню ситуацію, посіяти недовіру до влади або вплинути на прийняття політичних рішень.

Особливу увагу зловмисники приділяють елементам критичної інфраструктури, виборчим системам, базам персональних даних і каналам державної комунікації, що робить атаку не лише технічним актом агресії, а й елементом інформаційно-психологічного тиску в умовах конфлікту.

Отже, вразливість державного сектору до кіберзагроз формується внаслідок комплексної взаємодії технологічних, організаційних та геополітичних факторів, які не лише створюють численні точки входу для потенційного зловмисника, але й ускладнюють формування системної відповіді на загрози. Умови бюджетної обмеженості, недостатній рівень внутрішньої

кіберкультури та постійний зовнішній тиск з боку політично мотивованих акторів обумовлюють необхідність перегляду поточних підходів до цифрової безпеки. Подолання цієї вразливості потребує не лише модернізації технічної бази, але й глибоких інституційних змін — від підвищення обізнаності персоналу до побудови ефективної системи управління ризиками на міжвідомчому рівні.

1.4 Приклади реалізованих атак та їх наслідки

Розгляд теоретичних моделей і класифікаційних підходів до кіберзагроз створює необхідну основу для розуміння їхньої природи, однак лише емпіричний аналіз реальних кейсів дозволяє повною мірою усвідомити масштаб, складність і системні наслідки таких інцидентів. У цьому контексті особливо показовими є резонансні кібератаки проти державного сектору, які спричинили значні функціональні, фінансові та політичні втрати. Їх детальний розгляд не лише ілюструє потенційну шкоду, але й дає змогу ідентифікувати типові сценарії атак і вразливі точки.

Одним із найбільш руйнівних прикладів сучасної кіберагресії проти держави стала атака NotPetya, здійснена в червні 2017 року. За характером дії це був шкідливий програмний код, який мав ознаки програми-вимагача, але за своєю суттю виконував функцію кіберзброї, спрямованої на масове виведення з ладу інформаційних систем.

Атака розпочалася з компрометації ланцюга постачання — зловмисники зуміли модифікувати оновлення бухгалтерського ПЗ M.E.Doc, широко використовуваного українськими державними структурами. Після того як шкідливий код було завантажено й інстальовано легітимними користувачами, вірус почав швидко поширюватися в мережах за допомогою експлойтів EternalBlue та EternalRomance, використовуючи відомі вразливості операційної системи Windows.

Важливо, що на відміну від класичних ransomware-атак, NotPetya не передбачав реального механізму розшифрування даних після сплати викупу, що

дозволяє говорити про його деструктивну мету — параліч критичних інфраструктур.

Наслідки:

- Порушено функціонування державних установ, включаючи Міністерство фінансів, енергетичні компанії, залізничний транспорт, судову адміністрацію, поштову службу тощо.
- Зашифровано або знищено дані у десятках тисяч пристроїв, що вивело з ладу як локальні сервери, так і централізовані бази.
- Порушено логістику та комунікацію, включно з тимчасовим припиненням операцій в аеропорту «Бориспіль».
- Загальні збитки в Україні, за різними оцінками, сягнули від $300 млн до $1 млрд, що робить цю атаку однією з найдорожчих в історії.

У глобальному вимірі атака NotPetya стала прецедентом нового типу кібервійни, в якій головною метою є не здобуття доступу до ресурсів або шантаж, а тотальне руйнування функціональних можливостей держави. Інцидент продемонстрував, що уразливість навіть одного компонента в інфраструктурі постачання може призвести до системної катастрофи, масштаб якої виходить далеко за межі інформаційної сфери.

Ще одним фундаментальним прикладом високорівневої кібератаки, спрямованої на державний сектор, стала атака на компанію SolarWinds, виявлена у грудні 2020 року. Цей інцидент вважається класичною APT-атакою (Advanced Persistent Threat), яка реалізовувалася з використанням складної багаторівневої стратегії з довготривалою прихованою присутністю в інформаційних системах державних органів США.

Атака була організована через компрометацію ланцюга постачання: зловмисники змогли внести шкідливий код до оновлення програмного продукту Orion — популярної платформи для управління мережевою інфраструктурою, яку широко використовували в урядових та корпоративних середовищах. Після інсталяції зараженого оновлення на сервері SolarWinds, бекдор SUNBURST

активувався й надавав зловмиснику можливість віддаленого доступу, витоку даних та моніторингу систем упродовж багатьох місяців.

Загрозливість атаки полягала не лише в технічній витонченості, а й у її масштабності: за офіційними даними, було скомпрометовано щонайменше 9 агентств федерального уряду США [8], включаючи:

- Міністерство фінансів (U.S. Department of the Treasury),
- Міністерство енергетики (Department of Energy),
- Міністерство національної безпеки (Department of Homeland Security),
- Міністерство оборони (Department of Defense),
- Національні лабораторії,
- А також десятки приватних компаній, пов’язаних з держконтрактами.

Наслідки:

- Тривалий несанкціонований доступ до внутрішніх систем державних агентств, який, за оцінками, тривав до 9 місяців до моменту виявлення.
- Масштабний витік службових даних — як метаданих, так і конфіденційної кореспонденції.
- Порушення довіри до принципів кіберстійкості державних структур, що викликало необхідність глибокого аудиту всіх ланцюгів постачання.

Реакція на рівні національної безпеки США: створено нові політики, ініційовано перегляд процедур авторизації постачальників ПЗ, підвищено рівень міжвідомчої координації в сфері кіберзахисту.

Цей кейс продемонстрував вразливість навіть найбільш ресурсно забезпечених та технологічно розвинених держав, якщо атака здійснюється стратегічно через довіру до зовнішніх постачальників. Водночас інцидент актуалізував дискусію щодо національної цифрової суверенності та необхідності переосмислення політики кіберімпорту в держсекторі.

Хоча формально Colonial Pipeline є приватною компанією, що володіє однією з найбільших у США мереж трубопроводів для транспортування пального, її діяльність має прямий вплив на функціонування об’єктів критичної інфраструктури, відтак — на національну безпеку держави. Атака, здійснена у травні 2021 року, стала яскравим прикладом того, як кіберінцидент у приватному секторі може трансформуватися у кризу державного масштабу, що вимагає втручання органів вищого рівня управління.

Механізм атаки полягав у використанні програми-вимагача (ransomware), створеної угрупованням DarkSide [5]. Зловмисники проникли у внутрішню мережу компанії та зашифрували критичні бізнес-системи, внаслідок чого керівництво Colonial було змушене превентивно зупинити всю роботу трубопровідної мережі, аби не допустити поширення шкідливого коду до операційних систем реального часу.

Незважаючи на те, що мішенню стала приватна компанія, її інфраструктура забезпечувала до 45% постачання пального на східному узбережжі США, що автоматично перевело інцидент у сферу державної відповідальності. Уряд США був змушений оголосити надзвичайний стан у сфері транспортування енергоносіїв, активувати міжвідомчу координацію та залучити структури Міністерства енергетики та кіберкомандування.

Наслідки:

- Паливна криза в низці штатів, з чергами на автозаправках, панічними закупівлями та локальними перебоями у постачанні.
- Виплата викупу у криптовалюті, що згодом частково була відстежена й повернена Міністерством юстиції США.

Посилення політики національної безпеки щодо приватних компаній, які обслуговують критичну інфраструктуру — зокрема, було запроваджено нові вимоги до звітності про кібератаки та міжвідомчі процедури реагування.

Узагальнюючи, цей інцидент засвідчив, що межа між приватним і державним у сфері кібербезпеки є умовною, якщо йдеться про обслуговування критичних інфраструктур. Функціональна нездатність однієї компанії швидко

перетворюється на системний ризик, який зачіпає інтереси мільйонів громадян, що, у свою чергу, вимагає від держави активної ролі у кіберзахисті навіть приватних операторів у стратегічних секторах.

Проаналізовані кейси чітко демонструють, що навіть одиничний кіберінцидент здатен мати багатовимірні наслідки — від технічного порушення доступу до сервісів і тривалих збоїв у роботі критичних систем до витоку персональних або службових даних, які можуть бути використані як у комерційних цілях, так і для розвідки чи дезінформації. Крім того, прямі та непрямі втрати для державного бюджету, пов’язані з відновленням систем, розслідуванням, компенсацією збитків і впровадженням надзвичайних заходів, можуть сягати сотень мільйонів доларів. Проте ще більш загрозливою є втрата довіри громадян до інституцій [10], яка виникає у випадку невміння держави ефективно захистити власні системи, забезпечити безперервність публічних сервісів і зберегти контроль над інформаційними потоками. Така ситуація виводить проблему кібербезпеки далеко за межі IT-сфери, перетворюючи її на фактор політичної, соціальної та національної стабільності.

У ряді випадків, про підготовку до масштабних атак, таких як NotPetya чи SolarWinds, свідчили непрямі згадки у відкритих джерелах — аналітичних блогах, технічних форумах або звітах спеціалістів з інформаційної безпеки, зокрема у публікаціях OSINT-дослідників, таких як Скрипнік [11] чи Harding [31]. Це вказує на значний потенціал використання текстової OSINT-інформації як інструменту раннього попередження, що й обґрунтовує доцільність розробки відповідних аналітичних рішень.

1.5 Вплив кіберзагроз на ефективність державного управління

З огляду на зростаючу інтеграцію інформаційно-комунікаційних технологій у структури публічного адміністрування, кіберзагрози дедалі частіше трансформуються з технічної проблеми у системний виклик для державного управління. Вони здатні не лише порушити функціональність окремих

інформаційних систем, але й серйозно вплинути на якість прийняття рішень, стабільність інституцій та легітимність публічної влади. Нижче розглянуто ключові вектори впливу.

Одним із найбільш очевидних наслідків кібератак є тимчасове або тривале порушення роботи органів державного управління, зокрема центральних органів виконавчої влади, місцевих адміністрацій, судових та правоохоронних структур. У випадках, коли атаки спрямовані на сервери або бази даних, відбувається втрата доступу до внутрішніх документів, збої у міжвідомчій взаємодії, унеможливлюється нормальне функціонування документообігу чи реєстраційних процедур. Як наслідок, адміністративні процеси паралізуються, що безпосередньо позначається на здатності держави виконувати свої базові функції.

Особливо чутливим до кіберзагроз є виборчий процес, як один із ключових елементів легітимності публічної влади. Кібератаки можуть спричинити втручання у підрахунок голосів, саботаж електронних реєстрів виборців або дискредитацію результатів виборів через поширення дезінформації. У низці країн фіксувалися спроби втручання іноземних держав у національні вибори, що засвідчує, наскільки політично вмотивованими можуть бути цифрові інциденти. Збої або недовіра до прозорості виборчої процедури ставлять під загрозу стабільність політичної системи.

У цифрову епоху державні послуги дедалі частіше надаються через електронні платформи, такі як портали адміністративних послуг, електронні реєстри, системи податкової звітності, онлайн-кабінети громадян тощо. Кібератаки, спрямовані на ці сервіси, можуть призвести до масових збоїв, втрати даних, порушення безперервності обслуговування, а в деяких випадках — до несанкціонованого доступу до персональної інформації. Відмова в наданні послуг, навіть у короткостроковій перспективі, суттєво знижує рівень довіри до держави, особливо якщо мова йде про сервіси, критично важливі для громадян (пенсії, субсидії, соціальні виплати тощо).

Нарешті, у стратегічному вимірі кіберзагрози є прямим чинником дестабілізації національної безпеки. Зломи інформаційних систем органів оборони, інфраструктури управління надзвичайними ситуаціями, об’єктів критичної інфраструктури (енергетика, транспорт, зв’язок) здатні створити ситуацію оперативної безпорадності держави перед зовнішніми чи внутрішніми загрозами. Крім того, атаки інформаційно-психологічного характеру, спрямовані на формування недовіри, паніки або дезорієнтації населення, активно використовуються в умовах гібридних конфліктів та воєнного протистояння, набуваючи статусу інструмента стратегічного впливу [9].

Особливої уваги заслуговує зростання ролі інформаційних кампаній у цифровому середовищі, спрямованих на послаблення інституційної довіри до органів влади. Навмисна дезінформація, маніпулятивні вкидання, використання бот-мереж і deepfake-контенту стають частиною комплексних атак, які спрямовані не на IT-системи, а на свідомість громадян. Це веде до політичної поляризації, делегітимізації демократичних інститутів та ускладнення процесу ухвалення рішень.

У низці випадків масштабні кібератаки змушують державу витрачати значні ресурси на відновлення інфраструктури, що впливає на державний бюджет, затримує реалізацію реформ та знижує ефективність управління в умовах кризи. Повільне або неадекватне реагування на інциденти може бути сприйнято як свідчення управлінської неспроможності, що додатково підірвує довіру до держави з боку громадян, бізнесу та міжнародних партнерів.

З огляду на постійне зростання обсягів інформації у відкритому доступі, зокрема у формі новин, звітів, технічної документації та публікацій у соціальних мережах, текстові джерела інформації дедалі більше використовуються для виявлення потенційних кіберзагроз. Це відповідає висновкам дослідників [11], які описують ефективність застосування OSINT-інструментів для ідентифікації цифрових ризиків. Саме на підставі таких джерел можна оперативно ідентифікувати індикатори компрометації, вектори атак чи тенденції в цифровому середовищі [29]. Тому для збереження стійкості публічного

управління в умовах цифрових загроз критично важливо впроваджувати адаптивні, автоматизовані аналітичні системи на основі OSINT, NLP та машинного навчання. Такий підхід дозволяє не лише реагувати на інциденти, а й здійснювати проактивний моніторинг цифрового простору, що значно підвищує рівень державної готовності до нових викликів [13].

Висновки за розділом 1

У результаті проведеного аналізу встановлено, що кіберзагрози в державному секторі є не лише технічною проблемою, а й серйозним управлінським та безпековим викликом. Їхній вплив охоплює не лише інформаційні системи, а й важливі соціальні, економічні та політичні процеси. Особливу небезпеку становить динамічний та адаптивний характер загроз, що унеможливлює їх ефективне стримування виключно традиційними засобами захисту. Приклади реальних атак, таких як NotPetya чи SolarWinds, демонструють, що наслідки кіберінцидентів можуть бути масштабними й довготривалими.

Ключовими факторами вразливості державного сектору є застаріла інфраструктура, недостатнє фінансування, нестача фахівців, слабка нормативна база та низький рівень міжвідомчої взаємодії. Це формує складне середовище з високим ризиком, де навіть локальна атака може мати системний ефект.

Таким чином, забезпечення кіберстійкості держави вимагає переходу до проактивної, комплексної моделі захисту, що передбачає постійний моніторинг, швидке реагування та міжсекторальну співпрацю. Такий підхід має стати основою сучасної політики кібербезпеки в публічному управлінні.

РОЗДІЛ 2
OSINT ЯК ІНСТРУМЕНТ КІБЕРБЕЗПЕКИ

2.1 Основи, переваги та правові аспекти OSINT

Під терміном «розвідка на основі відкритих джерел» (Open Source Intelligence, OSINT) у сучасному інформаційному просторі розуміється структурований процес збору, фільтрації та аналітичної обробки публічної інформації, яка знаходиться у вільному доступі та отримується без порушення чинного законодавства. Згідно з позицією Європейського порталу даних, OSINT є інструментальною практикою, спрямованою на отримання релевантних знань для підтримки рішень у сфері національної безпеки, кримінального правозастосування та корпоративної розвідки.

Попри відсутність прямого визначення поняття OSINT у національному правовому полі, його застосування в Україні регламентується, зокрема, Законом України «Про доступ до публічної інформації» та Законом України «Про захист персональних даних». Ці документи формують правові межі допустимого збору, зберігання і використання відкритої інформації, особливо в частині, що стосується персоніфікованих даних і прав суб’єктів інформації.

Історичне становлення OSINT як самостійної дисципліни почалося задовго до ери цифрових технологій. Вже під час Другої світової війни аналогічні підходи використовувалися у діяльності Управління стратегічних служб США, яке, серед іншого, аналізувало відкриті публікації — наприклад, некрологи — з метою виявлення непрямих ознак функціонування ворожої інфраструктури. У подальшому, особливо після подій 2001 року, OSINT отримав формалізоване визнання у Сполучених Штатах через створення відповідних структур на рівні національної розвідки.

У європейських інституційних рамках розвиток OSINT активно підтримується через діяльність таких органів, як Європейське агентство з

кібербезпеки (ENISA), що сприяє інтеграції відкритих джерел до систем інформаційної безпеки та державного управління.

Розвідка на основі відкритих джерел (OSINT) у сучасних умовах розглядається як інтегрований інструмент аналізу, що забезпечує систематизований доступ до релевантної інформації, необхідної для прийняття рішень у сфері безпеки, аналітики ризиків, медіа чи корпоративного моніторингу. Його сутність полягає не лише у виявленні загальнодоступних даних, а передусім у формуванні структурованої, змістовно цінної розвідки, яка здатна оперативно відображати ситуаційний контекст. Практичні приклади методів OSINT наведені у класичних роботах, зокрема у Harding [35].

Функціональна логіка OSINT-бази ґрунтується на поетапній обробці інформаційного масиву: від збору до поширення. На початковій стадії здійснюється вилучення даних із публічних джерел, до яких належать вебресурси, соціальні мережі, медіаархіви, відкриті реєстри та академічні бази. Далі інформація проходить стадію фільтрації, під час якої усуваються дублікати, невідповідності та шумові елементи.

Наступним рівнем є аналітична інтерпретація, що охоплює виявлення зв’язків, трендів і потенційних аномалій із використанням методів дедуктивного аналізу або машинного навчання. Завершальним компонентом виступає передача отриманих результатів у форматі, зручному для кінцевого користувача — від інформаційних звітів до інтерактивних панелей.

Таким чином, OSINT не є суто технічним інструментом доступу до даних, а радше гнучкою платформою для створення інтелектуального ресурсу, який забезпечує міждисциплінарну взаємодію на основі відкритої інформації. Його ефективність детермінується якістю аналітичної обробки та відповідністю результатів до конкретних задач дослідження чи моніторингу.

В умовах загострення викликів у сфері інформаційної безпеки, оперативного управління ризиками та стратегічного моніторингу, OSINT демонструє низку переваг, що зумовлюють його активне впровадження в аналітичну практику як державного, так і приватного сектору. Однією з

ключових характеристик є відкритість джерел — доступ до яких не потребує спеціальних дозволів чи процедур автентифікації, що, своєю чергою, значно знижує інституційні та технологічні бар'єри використання інструментарію OSINT.

Висока оперативність, властива цьому підходу, обумовлена можливістю швидкого збору та обробки інформації з великої кількості публічних джерел, особливо у випадках, коли часові рамки є критично важливими. У цьому контексті особливу роль відіграє безперервний характер моніторингу, який дає змогу реагувати на події в режимі реального часу та, за необхідності, формувати аналітичні припущення щодо їхнього подальшого розвитку.

Крім того, масштабованість процесів OSINT, яка досягається за рахунок технічної інтеграції у цифрові екосистеми, дозволяє ефективно працювати як із локалізованими джерелами, так і з глобальними інформаційними потоками. Це робить OSINT надзвичайно гнучким інструментом — придатним для використання як у рамках великих урядових структур, так і в умовах обмежених ресурсів незалежних аналітичних ініціатив.

Не менш важливою перевагою є відповідність OSINT правовим вимогам, що, за умови дотримання чинного законодавства у сфері захисту персональних даних та інтелектуальної власності, надає цьому методу значної легітимності [30]. Додаткова цінність полягає у верифікованості інформації: відкритість джерел дозволяє здійснювати незалежну перевірку достовірності фактів, що сприяє формуванню прозорої, достовірної та обґрунтованої аналітики.

Незважаючи на те, що OSINT ґрунтується виключно на використанні відкритої інформації, питання правомірності та етичної допустимості відповідної діяльності залишаються принципово важливими. Законність обробки публічних даних не є абсолютною: вона залежить як від змісту інформації, так і від методу її отримання та подальшого застосування.

У межах українського законодавства визначальним нормативним актом є Закон України «Про захист персональних даних», який встановлює принципи правомірної обробки персональної інформації — навіть тоді, коли вона отримана

з відкритих джерел. Йдеться, зокрема, про дотримання умов пропорційності, цільового призначення, мінімізації втручання у приватність та обов’язкову згоду у разі обробки чутливих відомостей, що дозволяють ідентифікувати конкретну особу.

У міжнародному контексті особливу роль відіграє Загальний регламент про захист даних ЄС (GDPR), який встановлює уніфіковані вимоги до прозорості, правової визначеності та безпеки обробки даних. Відповідно до положень GDPR, навіть у разі відкритості джерел, обробка інформації має здійснюватися з урахуванням принципів privacy by design і privacy by default, що передбачає вбудований захист приватності у всі етапи роботи з даними.

Окремим виміром постає етична складова OSINT-практик, яка виходить за межі юридичних формальностей і стосується дотримання принципів доброчесності, невтручання, добровільності та поваги до гідності суб’єктів даних. Випадки використання відкритої інформації для маніпуляції, тиску або дискредитації суперечать стандартам відповідального аналітичного підходу та підривають суспільну довіру до таких інструментів.

З урахуванням цього, ефективне застосування OSINT вимагає не лише технічної обізнаності, а й юридичної компетентності — зокрема у частині дотримання локального та міжнародного комплаєнсу. Постійний моніторинг змін у нормативному полі та розробка внутрішніх політик відповідальності є запорукою того, що аналітична діяльність на основі відкритих джерел залишатиметься не лише корисною, але й легітимною.

2.2 Методи збору та обробки даних

У межах розвідки з відкритих джерел (OSINT) особливу роль відіграють як методи збору, так і подальша обробка інформації, що у сукупності формують основу для побудови достовірної та релевантної аналітики. Ці методи ґрунтуються на класифікації типів джерел та відповідному підборі

інструментарію, що дозволяє здійснювати збір як поверхневих, так і прихованих (але не захищених) цифрових слідів.

Згідно з усталеною практикою, джерела даних в OSINT-процесах поділяються на кілька функціонально відмінних категорій [31, 35]. По-перше, це контентні джерела, які включають текстові повідомлення, публікації, відео- та аудіоматеріали, графіку й інші форми медіаконтенту. Цей тип даних найчастіше використовується для тематичного аналізу, виявлення наративів, а також моніторингу громадських настроїв через соціальні мережі та медіа-платформи.

По-друге, технічні джерела становлять інформацію, що походить з мережевих інфраструктур: IP-адреси, записи DNS, WHOIS-дані, цифрові сертифікати, відомості про відкриті порти чи конфігурації серверів. Ці дані є критично важливими у розслідуваннях, пов’язаних із кіберзагрозами, зокрема для виявлення структури мережі, географічного розташування вузлів або зв’язків між суб’єктами цифрового середовища.

Третю категорію формують метадані, які, хоча і не містять основного контенту, несуть інформацію про самі дані — як-от час створення, джерело, геолокація, налаштування пристрою тощо. Метадані набувають особливого значення у випадках, коли контентний шар недостатній для встановлення контексту або автентичності інформації.

Ефективна реалізація методів збору OSINT значною мірою залежить від інструментів, які дозволяють автоматизувати чи напів автоматизувати ключові операції. До таких інструментів належать, зокрема:

- Maltego — платформа для візуалізації взаємозв’язків між об’єктами, зокрема доменами, IP-адресами, акаунтами в соцмережах, яка дозволяє формувати графові структури даних;
- Shodan — пошуковий сервіс, орієнтований на індексацію підключених до Інтернету пристроїв, що використовується для виявлення вразливих систем;
- TheHarvester — утиліта для збору електронних адрес, доменів, субдоменів, IP-адрес та іншої інформації шляхом парсингу відкритих джерел;

- Google Dorks — методика використання розширених операторів пошуку Google для знаходження специфічних даних або конфігурацій, які можуть бути помилково оприлюднені;
- Spiderfoot — фреймворк автоматизованого збору OSINT-даних, що підтримує інтеграцію з великою кількістю API та модулів для аналізу технічної інформації.

Кожен із цих інструментів виконує вузькоспеціалізовані функції, однак у комплексному використанні вони забезпечують повний цикл обробки відкритої інформації — від виявлення джерел до побудови структурованих аналітичних моделей.

Таким чином, методи OSINT вимагають не лише технічної обізнаності, а й чіткого розуміння природи джерел, а також дотримання етичних і правових меж, що дозволяє забезпечити аналітичну обґрунтованість без порушення норм приватності та інформаційної безпеки.

Ефективність OSINT-операцій безпосередньо залежить від чіткості методологічного підходу до організації процесів збору та обробки інформації, що перебуває у відкритому доступі. У цьому контексті важливим є не лише визначення релевантних джерел, а й правильний вибір засобів доступу до даних, а також способів їхньої валідації, структурування та подальшої інтерпретації.

Процес збору може реалізовуватись у двох основних режимах — ручному та автоматизованому. Ручний підхід передбачає безпосередню участь аналітика в ідентифікації джерел, фіксації інформації та попередній класифікації контенту, що, попри свою трудомісткість, забезпечує високий ступінь точності та критичної інтерпретації даних. Натомість автоматизовані методи збору базуються на застосуванні скриптів, ботів, API та програмних агентів, які здатні здійснювати моніторинг у реальному часі, охоплюючи значні обсяги інформаційного простору з мінімальним залученням людських ресурсів.

Особливу увагу слід приділити використанню інтерфейсів прикладного програмування (API), які дозволяють інтегруватися з соціальними мережами, пошуковими платформами або відкритими базами даних задля

систематизованого вилучення контенту. У випадках, коли API недоступне або обмежене, застосовуються методи парсингу — програмного зчитування структури HTML-сторінок для витягнення даних, що відображаються користувачеві, з урахуванням особливостей DOM-структури та специфікацій цільового ресурсу.

Зібрана інформація не може одразу розглядатися як придатна до аналізу. Передусім вона піддається попередній фільтрації — процесу виявлення і вилучення нерелевантних, застарілих, помилкових або дублікатних записів. На цьому етапі також здійснюється валідація даних, тобто перевірка їхньої достовірності, логічної узгодженості та технічної коректності, зокрема через перехресну перевірку з альтернативними джерелами або застосування евристичних правил.

Далі відбувається нормалізація — приведення даних до єдиного формату, що включає уніфікацію часових міток, географічних координат, текстових позначень тощо. У поєднанні з агрегуванням — об’єднанням однотипних об’єктів або атрибутів у зведені категорії — це забезпечує структурованість і уможливлює автоматизований аналіз великих масивів.

Таким чином, методологія OSINT не обмежується суто технічними операціями збору, а охоплює повноцінний цикл обробки даних — від вилучення до трансформації — з обов’язковим урахуванням точності, цілісності та відповідності інформації до потреб конкретного аналітичного завдання.

Для узагальнення інформації про ключові методи збору відкритих текстових джерел, таблиця 2.1 демонструє співвідношення між використовуваними техніками, джерелами даних та інструментами для автоматизації процесу.

Таблиця 2.1

Методи OSINT-збору текстових даних та відповідні інструменти

Метод	*Джерело*	*Інструмент*	*Примітка*
Парсинг	Новинні сайти	BeautifulSoup, Scrapy	Потребує гнучкої логіки та підтримки структури HTML
API	Соціальні мережі, новини	Twitter API, NewsAPI	Швидкий доступ до структурованих даних
Ручний аналіз	Блоги, форуми	Google, Reddit	Корисно при виявленні нестандартних повідомлень
NLP-аналіз	Текстові статті, звіти	spaCy, NLTK, BERT	Застосовується для класифікації та інтерпретації змісту повідомлень про загрози

2.3 Автоматизовані методи збору: API, парсинг і скрейпінг

У межах автоматизованих OSINT-процедур ключову роль відіграють програмні механізми збору великих обсягів даних із відкритих джерел. До них насамперед належать: інтерфейси прикладного програмування (API), парсинг HTML-сторінок та веб-скрейпінг. Кожен із цих підходів має свої переваги, обмеження і сфери доцільного застосування.

API (Application Programming Interface) — це набір визначених запитів і відповідей, який дозволяє отримувати структуровану інформацію без необхідності обробки веб-інтерфейсу. Більшість великих платформ, зокрема Twitter, YouTube, Reddit, OpenCorporates [26 – 28], надають відкриті або частково

відкриті API, через які можна витягувати пости, метадані, коментарі, геолокаційні позначки, часові мітки тощо. Формат даних зазвичай стандартизований — JSON або XML, що значно полегшує їх обробку. В роботі з API активно використовуються бібліотеки Python, зокрема requests, httpx, tweepy, aiohttp [17, 18, 20].

Перевагами API є:

- надійність доступу до оновлених даних;
- відповідність юридичним політикам платформи;
- висока швидкість і стабільність роботи;
- структурованість та обмеження шуму.

Проте API має і обмеження — обмеження на кількість запитів (rate limits), платні рівні доступу, часткове закриття функціоналу або повна його відсутність для деяких платформ.

У таких випадках застосовуються парсинг та скрейпінг.

Парсинг (англ. parsing) — це процес аналізу HTML-документів вебсторінок із метою вилучення цільової інформації, наприклад: заголовків новин, авторів, дат публікацій, основного тексту, тегів тощо. Парсинг вимагає точного знання структури DOM (Document Object Model) сторінки. Як правило, використовуються бібліотеки BeautifulSoup, lxml, re (регулярні вирази) для ручного або частково автоматизованого вилучення інформації [26 – 28].

Скрейпінг — ширший термін, який охоплює не лише парсинг однієї сторінки, а й масовий збір даних із багатьох сторінок або навіть сайтів. Для цього залучаються бібліотеки Scrapy, Selenium (для сайтів з JavaScript-контентом), Playwright. Скрейпінг дає змогу формувати масштабні бази даних із відкритої інформації, але він також стикається з низкою труднощів [27]:

- захист від ботів (CAPTCHA, JavaScript challenge, rate-limiting);
- часті зміни структури сайту;
- правові ризики при порушенні умов використання (ToS).

Таким чином, вибір між API, парсингом і скрейпінгом визначається доступністю даних, складністю реалізації та правовими обмеженнями. У рамках розробленого веб-інструменту застосовуються всі три методи, залежно від типу джерела та поставлених аналітичних задач: API — для регулярного доступу до соціальних мереж, парсинг — для витягу з новинних порталів, скрейпінг — для масового збору з HTML-ресурсів без API.

2.4 Аналіз текстів (NLP-аналіз) і класифікація загроз

У сфері OSINT-аналітики, орієнтованої на обробку великих масивів неструктурованої інформації, особливого значення набуває використання технологій Natural Language Processing (NLP), що забезпечують формалізацію текстових повідомлень, їхню лінгвістичну обробку, семантичну інтерпретацію та автоматизовану класифікацію. Завдяки цим технологіям відкривається можливість оперативного аналізу інформаційних потоків у режимі реального часу, що критично важливо в умовах динамічного кіберпростору.

Однією з базових процедур є виявлення ключових слів і фраз (keyword extraction), що реалізується шляхом частотного аналізу, статистичного моделювання (TF-IDF, RAKE) або застосуванням трансформерних моделей із вбудованим контекстним розумінням (наприклад, BERT, RoBERTa) [13, 14]. Ця процедура дозволяє виокремити тематику повідомлень, ідентифікувати потенційно небезпечні сигнали (наприклад, заклики до дій, згадки про атаки чи плани протестів), а також структурувати інформацію для подальшої класифікації.

Аналіз тональності (sentiment analysis) виступає наступним етапом, що спрямований на оцінку емоційного забарвлення тексту — позитивного, негативного або нейтрального. В окремих випадках використовуються більш складні моделі, що дозволяють ідентифікувати злісні наміри, риторику погроз, а також індикатори радикалізації. Такий аналіз є корисним у виявленні токсичних

інформаційних кампаній, деструктивних соціальних трендів або організованої дезінформації.

Важливою підсистемою NLP є витяг іменованих сутностей (Named Entity Recognition, NER) [13, 14, 20], який передбачає автоматичне розпізнавання згадок про об’єкти певних категорій — осіб, організації, локації, події, дати, технічні терміни тощо. У контексті OSINT це дозволяє не лише структурувати повідомлення, але й формувати семантичні мережі зв’язків, які можуть використовуватися для візуалізації кіберзагроз або виявлення спільних об’єктів у різних джерелах.

Для реалізації цих підходів застосовуються як універсальні фреймворки, такі як spaCy, NLTK та Stanza, так і високоточні трансформерні моделі, зокрема BERT, RoBERTa, DistilBERT, які здатні працювати з контекстною інформацією. В українському сегменті також з’являються адаптовані рішення — наприклад, Ukrainian NLP, lang-uk, ukr-roberta, які дозволяють ефективно аналізувати україномовний контент, включаючи повідомлення в соціальних мережах, новинних агрегаторах, телеграм-каналах та блогах.

Результати NLP-аналізу знаходять подальше застосування в класифікації загроз, що є одним із ключових напрямів OSINT. Загрози можуть класифікуватися:

За типом джерела — наприклад, діяльність хакерських груп (анонімних або ідеологічно вмотивованих), APT-груп (Advanced Persistent Threat), або суб’єктів кібертероризму, що діють у межах політично мотивованих чи державних структур.

За рівнем небезпеки — тобто оцінка масштабу можливих наслідків, імовірності реалізації загрози та потенціалу впливу на критичні активи. Для цього можуть застосовуватись матриці ризику, шкали CVSS або сценарні моделі оцінювання.

За векторами атаки — включаючи фішингові кампанії, експлуатацію вразливостей, соціотехнічні методи, атаки на інфраструктуру або інформаційні

ресурси. NLP-аналіз дає змогу автоматично зіставляти описові тексти із шаблонами векторів атак (наприклад, MITRE ATT&CK) [33].

Таким чином, поєднання NLP-підходів із методами таксономічної класифікації дозволяє створювати системи автоматичного розпізнавання, що не лише фіксують факт появи загрози, а й відразу ж приписують їй контекст, тип і ступінь небезпеки. Це, своєю чергою, забезпечує своєчасність реагування, зниження рівня невизначеності в інформаційному середовищі та підвищення якості рішень у сфері інформаційної безпеки.

2.5 Прогнозування та практичне застосування

В умовах зростаючої складності інформаційного середовища, що характеризується високою динамікою цифрових загроз, OSINT набуває значення не лише як засіб ретроспективного аналізу, але і як інструмент прогнозування потенційних інцидентів. Центральним елементом такої прогностичної моделі є поєднання технологій Natural Language Processing (NLP) із системами машинного навчання, що дозволяє виявляти патерни загроз, оцінювати ризики, а також будувати сценарні моделі реагування [13].

Один із прикладних механізмів, який реалізується на основі NLP-аналізу, — це розрахунок рівня загального ризику, що ґрунтується на інтеграції таких факторів, як емоційне забарвлення тексту (sentiment), кількість і тип іменованих сутностей (NER), а також тематичний контекст загроз. У функціональному плані це може реалізовуватись через умовну процедуру compute_risk(), яка агрегує текстові параметри в уніфікований ризиковий індекс, що дозволяє здійснювати автоматизовану фільтрацію або сортування інформаційних повідомлень за ступенем потенційної небезпеки [13, 14].

Одним із безпосередніх напрямів практичного застосування таких рішень є контентна фільтрація за рівнем загроз, що дозволяє виділяти критичні повідомлення із загального потоку відкритих даних. Зокрема, така фільтрація

може використовуватись для пріоритизації аналізу в системах медіамоніторингу або кіберрозвідки, де оперативність реагування має вирішальне значення.

Додаткову аналітичну цінність забезпечує візуалізація даних, зокрема через динамічні панелі інтерфейсу типу /analytics, що відображають частотність загроз, географічну локалізацію, ключові об’єкти згадок та динаміку ризикових повідомлень. Такі візуалізації дозволяють швидко виявляти пікові навантаження, нетипову активність або інформаційні атаки, які можуть бути неочевидними в текстовому вигляді [29].

Ще одним напрямом прикладної реалізації OSINT-аналізу є кластеризація новин за темами, що може здійснюватись на основі алгоритмів векторного подання текстів (TF-IDF, Word2Vec, BERT embeddings) у поєднанні з кластеризаційними методами (наприклад, K-means, DBSCAN) [13, 14]. У середовищі типу /clustering така функціональність дозволяє групувати контент за тематичними доменами: кібератаки, політична дестабілізація, соціальні протести тощо, що значно підвищує ефективність аналітичної роботи.

Окрему перспективу становить інтеграція подібних рішень у державні інформаційно-аналітичні сервіси, зокрема в контексті національної безпеки. Наприклад, використання OSINT для автоматизованого моніторингу публічних джерел, соціальних мереж або ЗМІ може забезпечити оперативне виявлення поширення дезінформації, організаційних координацій ворожих кампаній або спроб впливу на інформаційне поле. Такі системи можуть стати базою для створення національних центрів інформаційного захисту або реагування на загрози гібридного характеру.

Крім згаданих напрямів, слід відзначити потенціал адаптивного самонавчання OSINT-систем у контексті безперервного вдосконалення точності прогнозів. Використовуючи принципи reinforcement learning або online learning [31], такі системи можуть оновлювати вагові коефіцієнти моделей на основі реального відгуку — наприклад, залежно від того, наскільки релевантними виявилися попередні оцінки ризиків. Це відкриває шлях до створення

адаптивних аналітичних платформ, здатних ефективно працювати навіть у нестабільному або цілеспрямовано зміненому інформаційному середовищі.

Особливу роль у підвищенні прогностичної ефективності відіграє інтеграція з зовнішніми інформаційними каналами — такими як threat intelligence feeds, CERT-репорти, сигнали зі спеціалізованих індикаторних платформ (наприклад, MITRE ATT&CK, VirusTotal, MISP тощо) [4, 6]. Включення таких джерел у OSINT-аналітику дозволяє здійснювати мультиджерельну верифікацію подій, підвищувати точність класифікації та формувати комплексні сценарії загроз.

Крім автоматизованих інструментів, ефективність OSINT-практик значною мірою залежить від людино-орієнтованих інтерфейсів — аналітичних панелей, візуалізацій і динамічних фільтрів, які дозволяють користувачу не лише переглядати структуровані дані, а й взаємодіяти з ними, змінювати параметри кластеризації, вводити власні гіпотези та формувати альтернативні траєкторії аналізу. Подібна взаємодія реалізує підхід human-in-the-loop, який забезпечує високу гнучкість у випадках, коли автоматичні моделі стикаються з неоднозначністю.

З прикладної точки зору, можливості прогнозування та класифікації OSINT-систем можуть бути реалізовані не лише у сфері кібербезпеки. Вони мають також значний потенціал у контексті економічного моніторингу, кризового реагування, військової аналітики, антикорупційних розслідувань, а також у сфері інформаційної політики. Наприклад, автоматичне виявлення сплесків активності щодо певних економічних ключових слів у відкритих джерелах може сигналізувати про підготовку санкцій, ринкову маніпуляцію або сплановану кампанію впливу.

Таким чином, перспективи практичного застосування OSINT виходять далеко за межі вузькотехнічної безпеки, формуючи підґрунтя для побудови інформаційно-аналітичної надбудови, здатної не лише спостерігати, а й пояснювати та прогнозувати розвиток складних подій на стику інформаційних, соціальних і геополітичних чинників.

2.6 Проблеми та обмеження використання OSINT

Попри високу ефективність OSINT як аналітичного інструменту в сфері інформаційної безпеки, журналістики та корпоративного моніторингу, на практиці його застосування супроводжується рядом системних обмежень та викликів, які суттєво впливають як на точність результатів, так і на юридичну безпечність обробки даних.

Однією з головних проблем є надмірність відкритої інформації, що призводить до формування т. зв. «інформаційного шуму». З огляду на безперервне зростання обсягів цифрових публікацій — зокрема у соціальних мережах, блогах, онлайн-медіа — процес верифікації, фільтрації та структуризації контенту стає дедалі складнішим [11, 32]. Велика кількість дублювань, нерелевантних повідомлень, синтетично згенерованого тексту (зокрема ботами) не лише ускладнює аналітичний процес, а й знижує якість вхідного матеріалу для автоматичних моделей.

Не менш критичним є аспект достовірності даних. У відкритому інформаційному середовищі постійно циркулюють маніпулятивні повідомлення, фейки, дезінформація [9, 10], а також зміст, створений з метою свідомого викривлення фактів або провокаційного впливу. За відсутності обов’язкової редакційної модерації або верифікаційних механізмів, OSINT-аналітик або система автоматичного аналізу змушені застосовувати додаткові процедури перехресної перевірки (cross-validation), що значно ускладнює і сповільнює процес отримання висновків.

З технічної точки зору, діяльність у сфері OSINT нерідко стикається з обмеженнями доступу до джерел. Зокрема, масовий парсинг вебресурсів може супроводжуватись блокуванням IP-адрес, активацією CAPTCHA-захисту або обмеженням на кількість запитів через API [26, 27]. Багато цифрових платформ, особливо комерційні або геополітично чутливі, свідомо ускладнюють доступ до свого контенту, впроваджуючи антискрейпінгові алгоритми або обмеження на

перегляд архівів, що унеможливлює систематичний збір даних без спеціалізованих обхідних технологій.

Окрему категорію ризиків становлять правові та етичні аспекти, зокрема — можливість ненавмисного порушення норм захисту персональних даних чи комерційної таємниці. Навіть у випадках, коли інформація перебуває у відкритому доступі, її використання може суперечити принципам добросовісного збирання або перевищувати дозволені рамки вторинного аналізу, встановлені законом. Зокрема, Закон України «Про захист персональних даних» та Загальний регламент ЄС (GDPR) містять обмеження щодо повторного використання даних, які дозволяють ідентифікувати фізичних осіб або призводять до втручання в їхнє приватне життя [30].

Ще одним викликом є мовна специфіка OSINT-аналізу, особливо в україномовному сегменті інформаційного простору. Попри наявність окремих моделей і лінгвістичних інструментів для української мови, таких як lang-uk чи ukr-roberta, загальний рівень підтримки у популярних NLP-фреймворках істотно поступається англомовним аналогам. Це ускладнює реалізацію точних процедур токенізації, морфологічного розбору, лематизації та виявлення сутностей у текстах українською мовою. Крім того, український інформаційний простір часто містить змішане мовне середовище, що включає суржик, транслітерацію, помилки, регіональні діалекти, що ще більше ускладнює автоматизований аналіз.

Таким чином, хоча OSINT залишається потужним і гнучким інструментом, його ефективне застосування вимагає чіткого усвідомлення технологічних, правових і когнітивних обмежень, а також розробки комплексних підходів до їх подолання — шляхом удосконалення алгоритмів, підвищення юридичної обізнаності аналітиків і розвитком локалізованих мовних технологій.

2.7 Хибні спрацювання: причини, наслідки, шляхи зменшення

В процесі реалізації OSINT-аналізу, особливо в автоматизованих системах збору, класифікації та оцінки інформації, одним із найпоширеніших ускладнень є хибні спрацювання (false positives). Вони становлять собою ситуації, коли система або аналітик помилково ідентифікує безпечну інформацію як загрозливу або підозрілу. Попри технічну коректність таких спрацювань у межах заданих параметрів, їхня фактична значущість відсутня, що створює когнітивне навантаження, інформаційний шум та управлінські помилки.

До головних причин хибних спрацювань належать як технологічні, так і методологічні чинники. По-перше, недостатня точність або адаптація NLP-моделей — зокрема при роботі з полісемією, іронією, сарказмом або складними лінгвістичними конструкціями — призводить до помилкового виявлення загрозових сигналів. Особливо це характерно для україномовного контенту, де машинне розуміння контексту все ще обмежене через низьку представленість відповідних моделей у навчальних корпусах [13, 14].

По-друге, використання надто загальних або нечітких шаблонів класифікації ризиків, побудованих за принципом keyword matching, без урахування контексту або взаємозв’язків між сутностями, призводить до фальшивої тривоги. Наприклад, згадка слова «бомба» у контексті музичного альбому чи фільму може бути інтерпретована як індикатор терористичної загрози, якщо система не здатна враховувати семантичне поле вужчого контексту.

Ще одним джерелом хибних сигналів є аномалії у вхідних даних, що спричинені автоматично згенерованим контентом, зламами акаунтів, активністю ботів або спам-атаками, які навмисно створюють інформаційне перевантаження для систем моніторингу [12, 29]. У таких випадках система може розцінити нетипову частотність або синтаксичну схожість контенту як загрозу, хоча насправді це є результатом алгоритмічної атаки на саму OSINT-платформу.

Наслідки хибних спрацювань можуть бути як технічними, так і стратегічними. На технічному рівні — це зниження продуктивності систем, перевантаження обчислювальних ресурсів, збільшення часу на обробку та зниження точності моделей через "розмиття" релевантних патернів. На управлінському рівні — це формування хибних рішень, відволікання ресурсів на необґрунтовану загрозу, втрата довіри до автоматизованих систем і демотивація аналітиків, які змушені постійно проводити повторну перевірку.

Особливо небезпечними хибні спрацювання є у сфері державної безпеки та критичної інфраструктури, де вони можуть призвести до ескалації ситуацій, помилкового запуску процедур реагування або навіть порушення міжнародних комунікаційних протоколів у випадку міждержавних інформаційних конфліктів.

Для зниження частоти хибних спрацювань застосовуються методи багаторівневої перевірки (multi-layer validation), семантичного розпізнавання контексту, комбіновані моделі з людською експертизою (human-in-the-loop), а також періодичне оновлення лексиконів і тренування моделей з урахуванням нових типів загроз і мовних зрушень.

У відповідь на зазначені проблеми сучасні системи OSINT дедалі частіше орієнтуються на інтеграцію багаторівневих підходів, які поєднують алгоритмічну обробку, семантичний аналіз і людську експертизу. Одним з найбільш дієвих рішень є застосування гібридних моделей, що поєднують машинне навчання з логічними правилами та експертними фільтрами [29]. Такі моделі можуть адаптивно перемикатися між автоматизованими й ручними сценаріями в залежності від ступеня впевненості у вхідних даних.

Ключову роль у підвищенні точності відіграє розширення навчального корпусу, що дозволяє моделі краще розуміти специфіку мови, сленгу, контекстів і структури речень [13, 14]. Особливо це стосується україномовного сегменту, де впровадження глибших мовних моделей (наприклад, ukr-BERT, lang-uk XXL, монолінгвальні RoBERTa-моделі) забезпечує точнішу інтерпретацію інформаційного контексту та зниження кількості помилкових класифікацій.

Ще одним напрямом оптимізації є використання метаінформації (метаданих) — часових міток, геолокації, історії активності, типу пристрою, що дозволяє системі виводити додаткові критерії достовірності повідомлення [33, 35]. Наприклад, однотипний текст, що повторюється протягом короткого проміжку часу з однієї IP-адреси, має нижчу вагу, ніж унікальний контент із перевіреного акаунту.

Також доцільним є впровадження семантичних фільтрів другого рівня, які працюють після початкової класифікації повідомлення. Такі фільтри дозволяють враховувати ко-тексти (наприклад, суміжні фрази, інтонаційні структури) для виключення ситуацій, коли окремі слова виявляються поза загальним змістом. Це особливо важливо при роботі з емоційно зарядженими або сатиричними повідомленнями, де формальні індикатори можуть хибно сигналізувати про ризик.

У випадках високої невизначеності доцільно застосовувати механізми підтвердження з альтернативних джерел — наприклад, перехресний аналіз згадок у декількох незалежних інформаційних потоках. Такий підхід дозволяє уникнути помилкових висновків, заснованих на одиничному вкиді або фейковому повідомленні.

Нарешті, ключову роль відіграє реалізація зворотного зв’язку з боку аналітиків — тобто фіксація випадків хибного спрацювання, які використовуються для донавчання моделі (feedback loop). Це дозволяє не лише зменшити кількість помилок у майбутньому, а й підвищити чутливість системи до змінних патернів поведінки та мовних трансформацій.

Таким чином, зменшення кількості хибних спрацювань є не лише технічним викликом, а стратегічним завданням для стабільного функціонування OSINT-систем. Його вирішення вимагає не одномоментного втручання, а постійного процесу адаптації, міждисциплінарного удосконалення й еволюції інструментів на основі зворотного досвіду та контекстуальної обізнаності.

Висновки за розділом 2

У межах даного розділу було здійснено багатовимірний аналіз теоретичних засад OSINT — як із позицій технологічної структури, так і з урахуванням нормативно-правових, мовних та методологічних особливостей його застосування. Представлені концептуальні положення формують основу для формалізації розвідки з відкритих джерел як системного, цілісного інструменту, що інтегрує лінгвістичні, аналітичні, правові та технічні складові.

Зокрема, були окреслені основи збору даних з використанням як ручних, так і автоматизованих методів, включно з API, парсингом і скрейпінгом. Розглянуто структуру обробки — від первинної фільтрації до нормалізації та агрегації інформації. Окрему увагу було приділено NLP-компонентам: виявленню ключових фраз, аналізу тональності, класифікації сутностей і формуванню ризикових профілів повідомлень.

Застосування методів обробки природної мови (NLP) відіграє центральну роль у вивченні текстових джерел, дозволяючи трансформувати неструктурований контент у формалізовані ознаки, що можуть бути використані для аналітики. Саме за допомогою NLP можливо автоматично ідентифікувати індикатори загроз, виявляти повторювані шаблони поведінки зловмисників, аналізувати часову динаміку ризиків і виводити семантичні зв’язки між об’єктами. У контексті кібербезпеки це дозволяє вчасно виявляти потенційно небезпечну інформацію навіть у складному багатомовному середовищі, що є особливо важливим для державного сектору.

Описано методи оцінки ризику та візуалізації результатів, а також проблеми достовірності, перевантаження даними, технічні та правові обмеження. У результаті розділ створює цілісну теоретичну основу для подальшої реалізації програмного рішення, дозволяючи обґрунтувати вибір технологій, інструментів та архітектури веб-інструменту, який буде детально розглянуто у наступному розділі.

РОЗДІЛ 3
РОЗРОБКА ВЕБ-ІНСТРУМЕНТУ ДЛЯ OSINT-АНАЛІЗУ ТЕКСТОВИХ ДЖЕРЕЛ

3.1 Архітектура системи та технології

Для реалізації веб-інструменту OSINT-аналізу необхідно обрати ефективну архітектурну модель та відповідний стек технологій, які дозволили б забезпечити швидкість обробки, масштабованість, гнучкість у роботі з різними джерелами даних і зручність для користувача. У цьому підрозділі описується загальна структура системи, використані технології та логіка взаємодії між компонентами.

Програмне забезпечення для проведення OSINT-аналізу буде реалізоване у вигляді веб-орієнтованого застосунку, побудованого на базі мікрофреймворку Flask [15, 17], що забезпечує мінімальну інфраструктуру для створення серверної логіки з урахуванням вимог до гнучкості та масштабованості. Загальна архітектура системи відповідає класичній парадигмі клієнт-серверної моделі, у межах якої фронтенд-компонент виконує функцію інтерфейсу користувача, тоді як бекенд відповідає за обробку запитів, аналітичну обробку даних, зберігання інформації та виконання алгоритмічно складних обчислень. Структурну взаємодію між основними компонентами системи подано на рисунку 3.1.

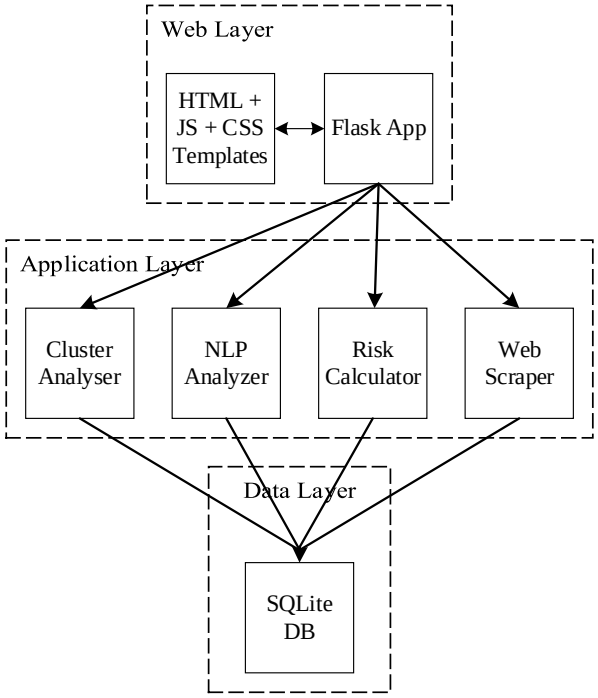

Рисунок 3.1 – Архітектура веб-інструменту для OSINT-аналізу текстових джерел

З метою забезпечення ефективної обробки даних було інтегровано такі технологічні компоненти:

- Flask — мінімалістичний веб-фреймворк мови Python, який забезпечує маршрутизацію запитів та гнучке управління станом застосунку;
- SQLAlchemy — об'єктно-реляційний мапер (ORM), що забезпечує абстрагований доступ до бази даних на основі SQLite, дозволяючи зменшити кількість SQL-запитів вручну;
- HTML/CSS/JavaScript — стандартні технології веб-розмітки, що застосовуються для побудови інтерфейсу користувача із залученням шаблонізатора Jinja2 [22, 24, 25], реалізованого у директоріях templates/ і static/;
- NLTK, spaCy, scikit-learn — програмні бібліотеки для обробки природної мови, зокрема для виявлення мовних конструкцій, аналізу настроїв, виявлення загроз [18, 20] на основі ключових слів та кластеризації текстових фрагментів;

- ThreadPoolExecutor — високорівневий інтерфейс до багатопотокового виконання, що забезпечує асинхронну обробку веб-ресурсів під час скрапінгу;
- BeautifulSoup — інструмент парсингу HTML-контенту, який використовується для вилучення релевантної інформації зі сторінок веб-сайтів.

У межах проєкту створено відповідну ORM-модель для зберігання результатів скрапінгу у реляційній базі даних. Кожен запис містить URL-адресу ресурсу, заголовок сторінки, повний текстовий вміст, кількість абзаців, тривалість обробки запиту та мітку часу, що відповідає моменту збереження інформації.

Окрім базових функціональних модулів, реалізовано підсистему для семантичного аналізу тексту, яка здійснює виявлення ймовірних загроз на основі ключових слів як англійською, так і українською мовами. Механізм автоматичного визначення мови, заснований на бібліотеці langdetect, дозволяє адаптувати перелік релевантних термінів відповідно до контексту.

Аналіз небезпеки базується на метриках співвідношення кількості "позначених" речень до загальної кількості речень у документі, а також на оцінці емоційного забарвлення тексту із залученням аналізатора настроїв VADER. У випадках, коли текст подано англійською мовою, додатково виконується виокремлення іменованих сутностей за допомогою мовної моделі spaCy.

Для оцінювання рівня ризику застосовано комбіновану метрику, яка враховує кількість виявлених загроз, ймовірність їхнього виникнення, а також полярність емоційної оцінки тексту. Одержане значення служить кількісним індикатором рівня кіберзагрози.

З метою забезпечення масштабованості системи для аналізу великої кількості джерел реалізовано функціонал кластеризації документів із використанням алгоритму K-середніх (KMeans) на базі TF-IDF-векторизації [19]. Це дозволяє групувати схожі за тематикою документи, визначати домінуючі лексичні одиниці в межах кожного кластера та будувати евристичну модель інформаційного простору.

Водночас реалізовано побудову мережевої моделі співзгадуваних сутностей, у якій вузли представляють унікальні іменовані об'єкти, а ребра — кількість спільних згадувань у межах окремих документів. Це дозволяє ідентифікувати ключові зв'язки між суб'єктами інформаційного простору.

Таким чином, реалізоване рішення забезпечує повноцінну архітектурну, інструментальну та функціональну базу для здійснення багаторівневого OSINT-аналізу [1, 29, 33], спрямованого на виявлення, категоризацію та оцінювання ризиків інформаційних загроз у відкритих джерелах.

3.2 База даних та модулі збору інформації

У процесі реалізації прикладного програмного забезпечення для автоматизованого збору текстових даних з відкритих джерел, доцільно застосування вбудованої реляційної системи керування базами даних SQLite [21]. Основною причиною вибору даного рішення слугувала його повна сумісність із мовою програмування Python, відсутність необхідності в налаштуванні окремого серверного середовища, а також здатність забезпечувати задовільну продуктивність у випадках локального зберігання структурованих даних середнього обсягу.

Базова схема зберігання представлена єдиною, але функціонально завершеною таблицею ScrapedData, структура якої сформована з урахуванням вимог щодо збереження як основної текстової інформації, так і супровідних метаданих. Таблиця містить такі атрибути:

- url (Text) — символьне поле, призначене для фіксації повної адреси джерела інформації;
- title (Text) — назва HTML-документа, вилучена з відповідного тега;
- content (Text) — агрегований текстовий контент, сформований у результаті обробки текстових абзаців сторінки;
- scrape_duration (Float) — часовий інтервал у секундах, протягом якого здійснювався процес вилучення даних;

- num_paragraphs (Integer) — кількісна оцінка абзаців, що були ідентифіковані й збережені;
- timestamp (DateTime) — автоматично згенерована позначка часу, яка відповідає моменту запису відповідного екземпляра даних у базу.

Ініціалізація структури таблиці, як і подальші операції створення або оновлення записів, реалізовані за допомогою бібліотеки SQLAlchemy, що функціонує як об'єктно-реляційний міст між Python-кодом та відповідним SQL-представленням. Такий підхід, хоча й дещо збільшує накладні витрати на обробку, значно підвищує гнучкість та читабельність коду.

Модуль збору інформації реалізовано у вигляді окремої функції scrape_single, функціональне призначення якої полягає у здійсненні повного циклу отримання, аналізу та фрагментації веб-документів. З технічної точки зору, ця функція виконує послідовність наступних операцій:

- ініціювання HTTP-запиту до цільового ресурсу з використанням бібліотеки requests;
- парсинг HTML-контенту у форматі DOM-дерева з застосуванням засобів бібліотеки BeautifulSoup [26];
- вилучення значення тега , що відповідає за заголовок сторінки;
- ідентифікація та фільтрація абзаців, представлених тегами , з подальшим об'єднанням текстових фрагментів у суцільний масив;
- фіксація часу виконання операції скрапінгу з метою подальшого аналізу ефективності;
- генерація об'єкта таблиці ScrapedData та передача його ORM-движку для коміту в локальну базу даних.

Задля підвищення загальної продуктивності системи було впроваджено механізм паралельної обробки запитів, що дозволяє значно зменшити час, необхідний для обробки великої кількості джерел. Конкретна реалізація цього механізму базується на використанні класу ThreadPoolExecutor з модуля concurrent.futures, за допомогою якого формуються окремі потоки виконання для

кожного скрапінгового завдання. Такий підхід, хоча й не є повністю асинхронним з погляду GIL-архітектури Python, усе ж демонструє прийнятну масштабованість та ефективність у межах задач, де затримки, пов'язані з мережею, є головним обмежувальним фактором. Загальну послідовність обробки даних у межах реалізованого веб-інструменту подано у вигляді блок-схеми на рисунку 3.2. Вона відображає основні етапи роботи системи — від ініціалізації запиту до формування структурованого результату в базі даних. Відповідний програмний код, що реалізує логіку цієї схеми, подано у додатку Б.

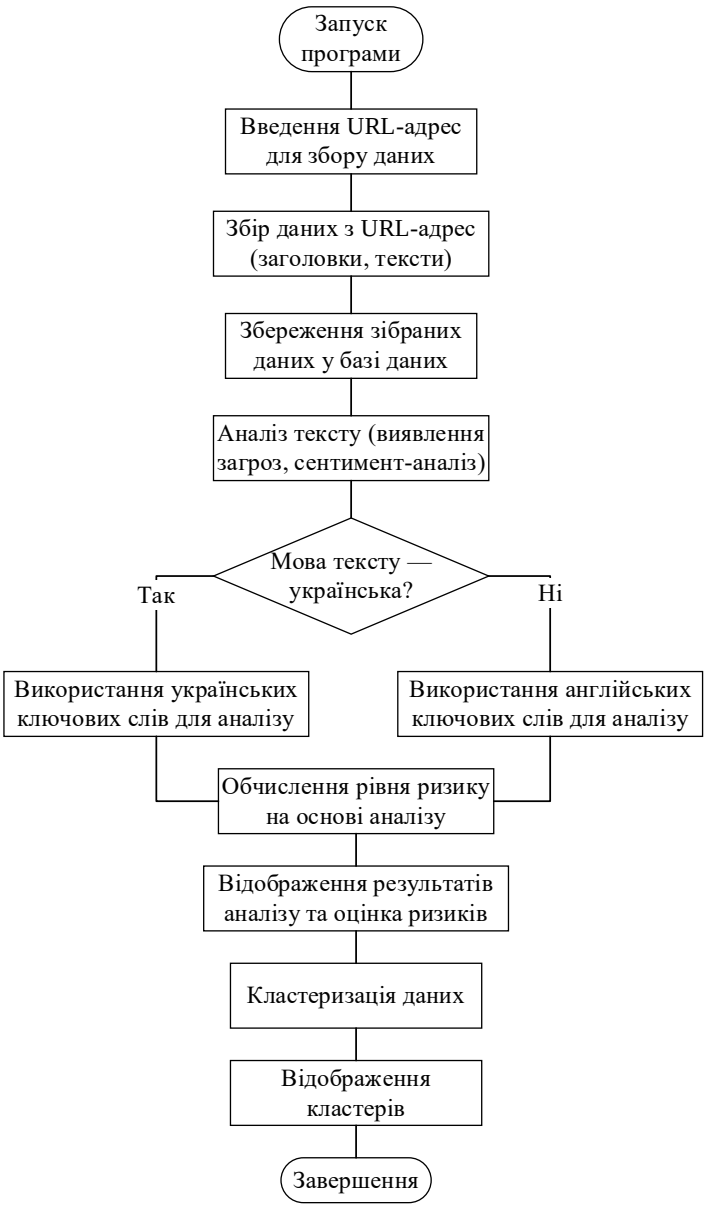

Рисунок 3.2 – Блок-схема роботи програмного забезпечення для OSINT-аналізу

У результаті впровадженого архітектурного рішення забезпечується не лише коректне та структуроване збереження первинних текстових даних, але й оптимізація процесу їхнього отримання, що критично важливо в контексті побудови систем OSINT-аналітики з потенційною можливістю подальшої семантичної обробки, тематичної класифікації чи візуального представлення інформації.

3.3 Обробка текстів і визначення загроз

Процес ідентифікації потенційних загроз у текстовій інформації, отриманій із відкритих джерел, реалізовано через спеціалізований програмний модуль, функціональність якого зосереджено у функції analyze_text. Ця функція виступає як центральна логічна одиниця обробки, що послідовно здійснює операції лінгвістичної та семантичної інтерпретації вхідних даних з метою виявлення інформаційних одиниць, які можуть вказувати на наявність загроз кібербезпеці [1, 11, 29, 33, 34].

Початковим кроком є автоматизоване визначення мови тексту з використанням засобів бібліотеки langdetect, що забезпечує можливість коректного подальшого застосування мовно-специфічних моделей аналізу. Визначення мови відбувається на основі статистичних характеристик текстового корпусу й дозволяє обирати відповідні словники загроз.

Після ідентифікації мовної приналежності виконується розбиття тексту на речення з наступною перевіркою кожного сегмента на наявність лексем, що містяться у сформованих словниках ключових слів, пов'язаних із потенційними загрозами. Зазначені словники формуються окремо для кожної мови (української та англійської) та містять типові терміни, які мають відношення до шкідливої активності, інформаційного впливу або кіберінцидентів.

Наступна фаза включає класифікацію рівня загрози, що реалізується через побудовану евристичну модель, яка оперує трирівневою шкалою (Low, Medium, High). При цьому враховуються як кількість виявлених ключових лексем, так і контекстуальні чинники, що визначають ступінь критичності повідомлення.

Алгоритм класифікації враховує не лише лексичну насиченість, а й структурну складність синтаксичних конструкцій, у яких згадані лексеми з’являються.

Для підвищення точності оцінки залучається аналіз емоційного тону повідомлення з використанням алгоритму VADER, що входить до складу бібліотеки nltk [18, 19]. Цей метод, оптимізований під короткі тексти неформального характеру, дозволяє виявляти як позитивну, так і негативну конотацію висловлювань. Наявність негативного емоційного тону у текстах, що вже класифіковані як потенційно загрозливі, може слугувати додатковим аргументом на користь підвищення рівня ризику.

У випадку, якщо автоматично виявлено, що мова тексту є англійською, ініціюється процедура розпізнавання іменованих сутностей (Named Entity Recognition) із використанням бібліотеки spaCy [20]. Застосування цієї моделі дозволяє виділяти згадки про суб’єкти, об’єкти або організації, що можуть бути причетними до описаних подій чи становити окрему загрозу. Результати NER-аналізу є особливо цінними для встановлення зв'язків між фрагментами інформації в подальших етапах обробки.

Підсумковим етапом виступає виклик функції compute_risk, у якій здійснюється об’єднання всіх попередньо отриманих параметрів: кількість речень, що містять ознаки загрози, результати аналізу тональності та класифікаційні ознаки загрозливих фрагментів. Результуючий рівень ризику фіксується у базі даних з метою подальшого аналізу, кореляції з іншими подіями, а також візуалізації в межах інтерфейсу моніторингу ситуаційної обстановки.

3.4 Користувацький інтерфейс

Інтерфейс користувача розроблений як веб-додаток з інтуїтивно зрозумілим дизайном, що дозволяє взаємодіяти з усіма основними функціональними модулями системи. Завдяки чіткому поділу на логічні вкладки — «Головна», «NLP Аналіз», «Entity Network», «Кластеризація», «Cyber Risk» — користувач може швидко орієнтуватися в можливостях інструменту, не заглиблюючись у технічні деталі реалізації.

Після переходу на головну сторінку системи користувач бачить вхідне вікно (див. рис. 3.3), яке запрошує ввести посилання на новинний ресурс. Це може бути одне або кілька посилань, розділених комами. Натиснувши кнопку «Почати парсинг», користувач ініціює процес автоматизованого збору текстового контенту зі вказаного джерела.

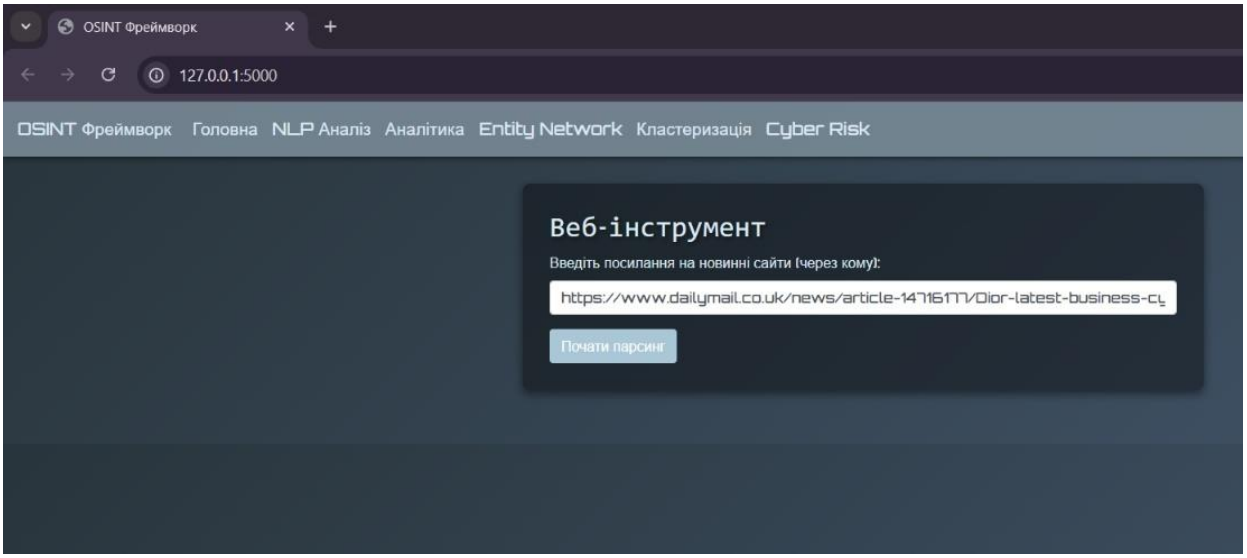

Рисунок 3.3 – Головна сторінка інтерфейсу з формою введення URL для парсингу

Як наслідок, після завершення обробки користувач потрапляє на екран результатів парсингу (рис. 3.4). Тут у зручному табличному форматі відображено зібрані метадані: URL джерела, заголовок статті, час, витрачений на парсинг, кількість абзаців у тексті, а також його загальна довжина. Крім цього, система відразу пропонує перейти до наступного кроку аналізу — NLP Аналізу, натиснувши відповідну кнопку.

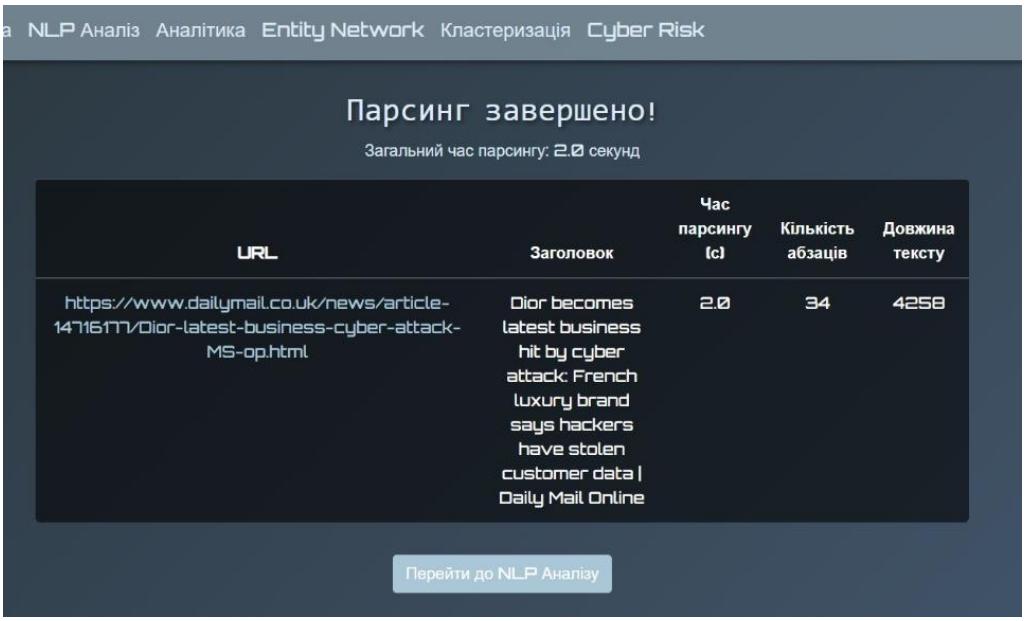

Рисунок 3.4 – Сторінка з результатами парсингу новинного ресурсу

Далі, після переходу на вкладку NLP Аналіз, відкривається сторінка з результатами лінгвістичної обробки зібраних статей (рис. 3.5). Тут, у розширеній таблиці, відображено ID документа, URL, заголовок, кількість абзаців, довжину тексту, час парсингу, а також кількість виявлених загроз або ключових індикаторів. Таким чином, кор.истувач одразу отримує стислу, але інформативну аналітику щодо оброблених джерел.

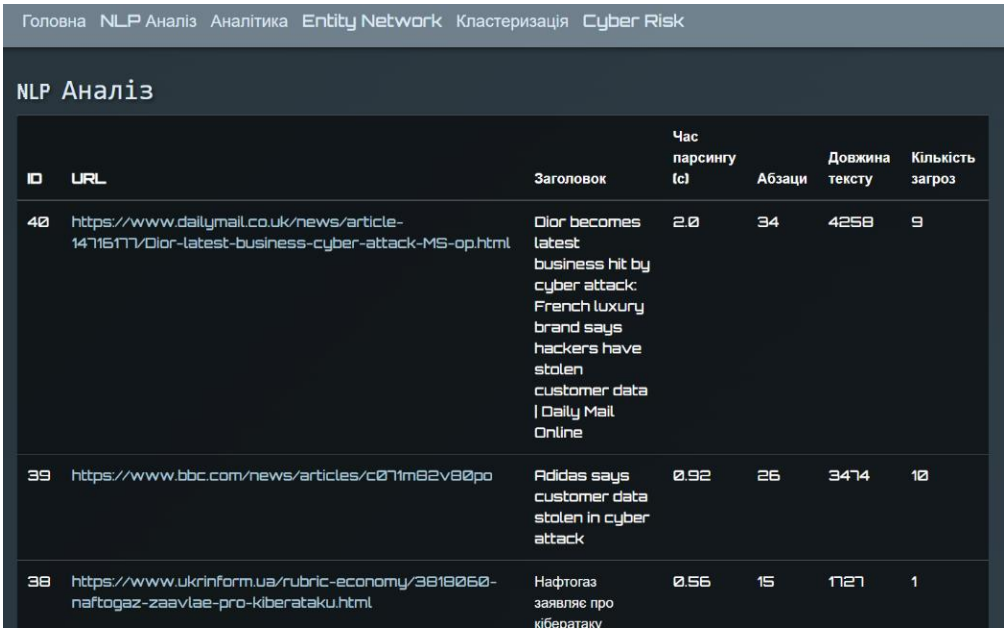

Рисунок 3.5 – Вкладка NLP Аналіз з таблицею оброблених новинних джерел

Після аналізу сутностей користувач може перейти до розділу «Аналітика», що забезпечує розширене представлення зібраних даних у вигляді діаграм, таблиць та графіків (рис. 3.6).. Це дозволяє не лише переглядати окремі результати парсингу та NLP-обробки, а й оцінювати загальні тенденції та закономірності.

Розділ «Аналітика» відкривається інтерфейсом, який дозволяє застосовувати гнучкі фільтри за датою та рівнем загрози. Завдяки цій функції користувач може обмежити вибірку джерел за певним часовим інтервалом або проаналізувати лише повідомлення з визначеним рівнем ризику (наприклад, лише з високим рівнем загрози).

Нижче розміщена таблиця з усіма зібраними параметрами, серед яких — ID запису, URL, заголовок, тривалість парсингу, кількість абзаців, довжина тексту тощо. Це значно полегшує ручний перегляд великої кількості даних та сприяє детальному аналізу джерел, зокрема при потребі виявити записи з найбільшою кількістю виявлених загроз чи аномалій.

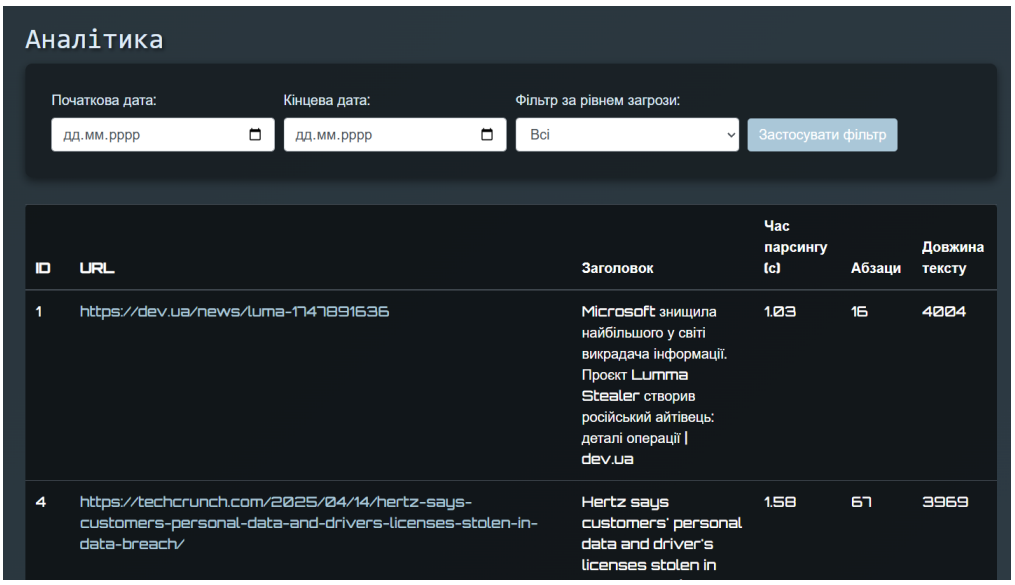

Рисунок 3.6 – Панель фільтрації та зведена таблиця джерел у модулі аналітики

Наступним компонентом інтерфейсу є гістограма, яка ілюструє розподіл довжин текстів (рис. 3.7). Завдяки ній можна легко виявити аномальні значення — наприклад, надмірно довгі чи короткі статті, які можуть впливати на точність

подальшого аналізу. Зокрема, на графіку видно наявність декількох записів із суттєво вищою довжиною тексту, що одразу привертає увагу аналітика.

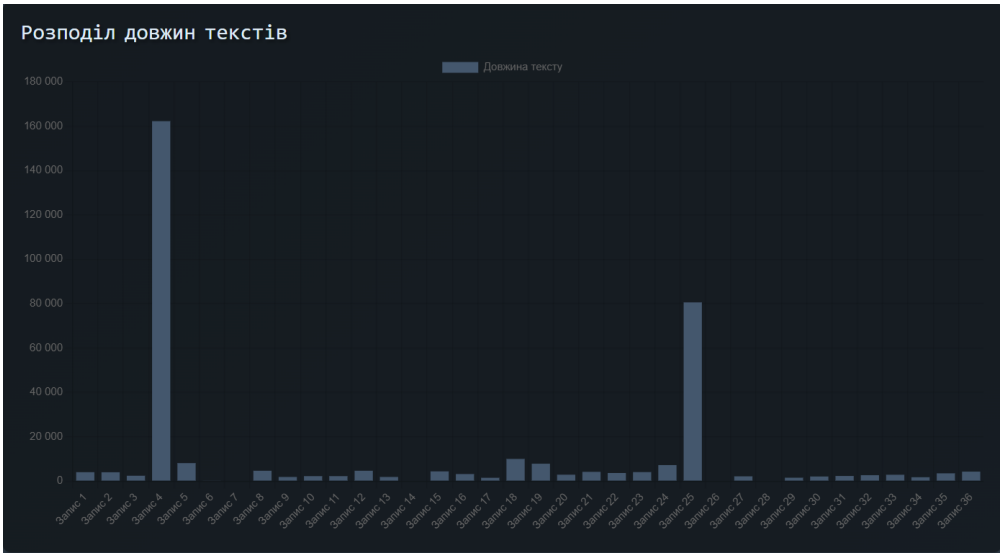

Рисунок 3.7 – Гістограма розподілу довжин оброблених текстів

Крім того, система дозволяє переглядати розподіл довжин заголовків (рис. 3.8), що відображено на окремому лінійному графіку. Завдяки цьому графіку користувач може оцінити, наскільки змістовними є заголовки проаналізованих статей. Здебільшого довжина заголовків є сталою, проте спостерігаються і певні відхилення, які можуть вказувати на технічні помилки або специфіку джерела.

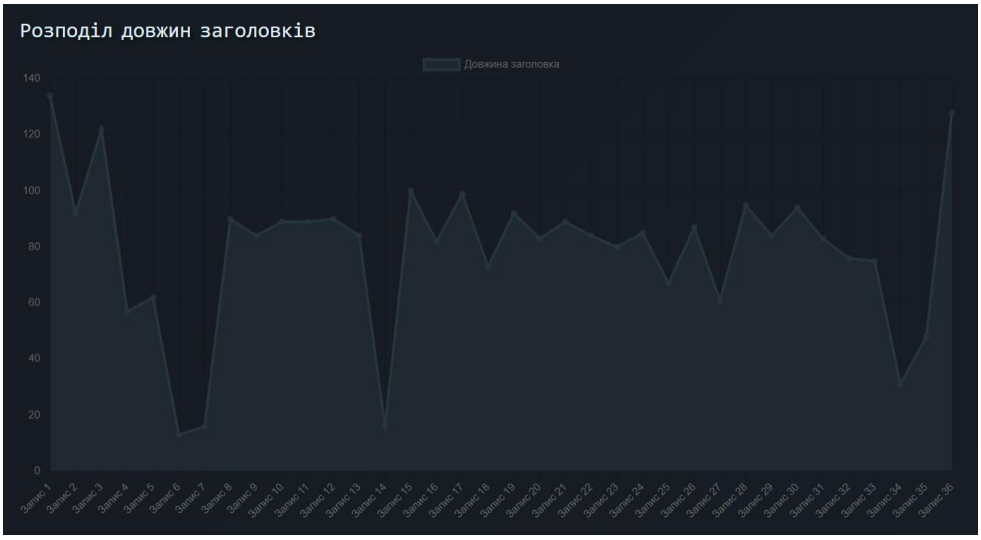

Рисунок 3.8 – Графік розподілу довжин заголовків новинних повідомлень

Ще одним важливим аспектом є аналіз настроїв (sentiment analysis), результати якого представлені у вигляді стовпчикового графіка. На цьому графіку відображено зведене значення Compound Score (див. рис. 3.9). для кожного документа — інтегральний індикатор, що показує загальну емоційну тональність тексту (від -1 до 1). Позитивні значення свідчать про позитивний настрій повідомлення, негативні — про негативну забарвленість, а значення, близькі до нуля, вказують на нейтральність.

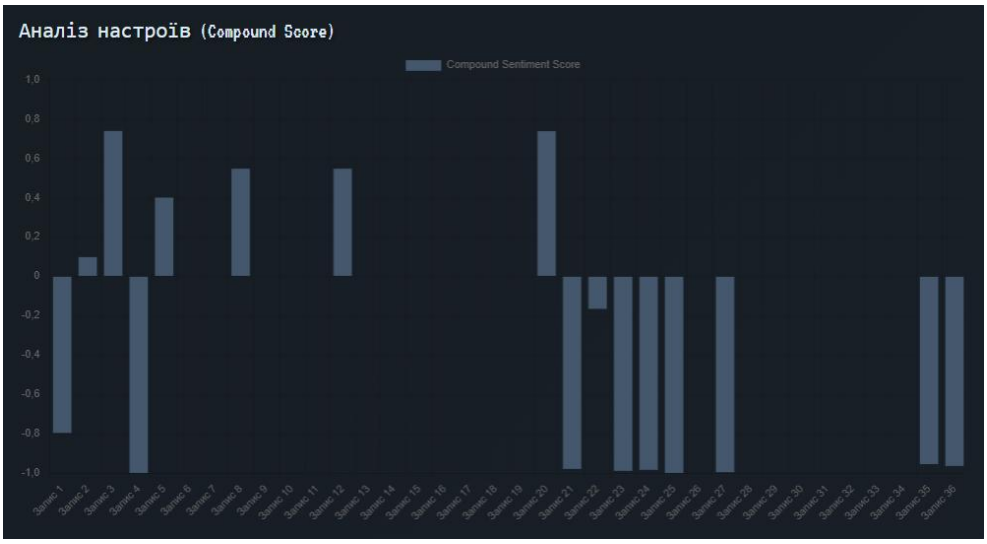

Рисунок 3.9 – Графік аналізу настроїв (Compound Sentiment Score) для оброблених статей

Окремо варто виділити блок з агрегацією ключових слів, який поєднує тегову хмару та кругову діаграму. Цей компонент демонструє, які терміни найчастіше зустрічалися серед виявлених загроз (див. рис. 3.10).. Візуалізація дозволяє не лише побачити переважаючі поняття (наприклад, “phishing”, “malware”, “data breach”), а й зробити висновки щодо тематики інформаційного поля в аналізований період.

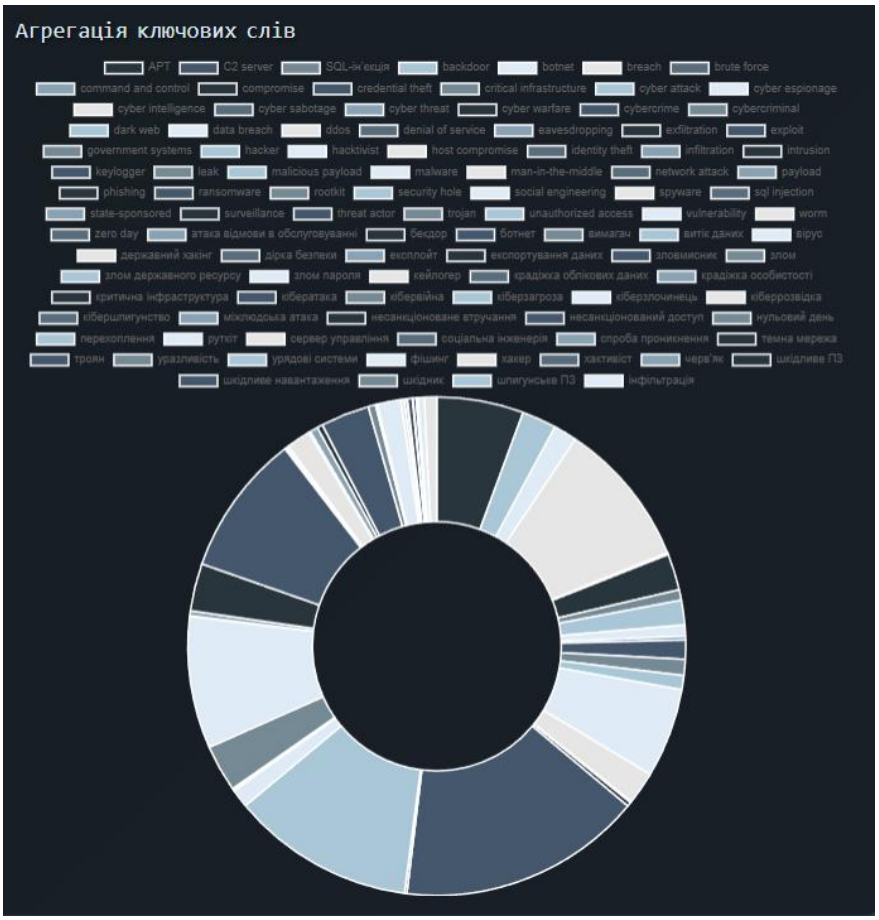

Рисунок 3.10 – Агрегація ключових слів, згрупованих за частотністю у вигляді хмари та діаграми

Крім індивідуального аналізу, система також агрегує дані для оцінки загального емоційного фону. У цьому випадку використано радарну діаграму (рис. 3.11), що ілюструє середні значення показників позитивності, негативності, нейтральності та загального Compound-індексу. Подібне зображення дозволяє швидко оцінити, який тип емоцій переважає у загальному наборі новин.

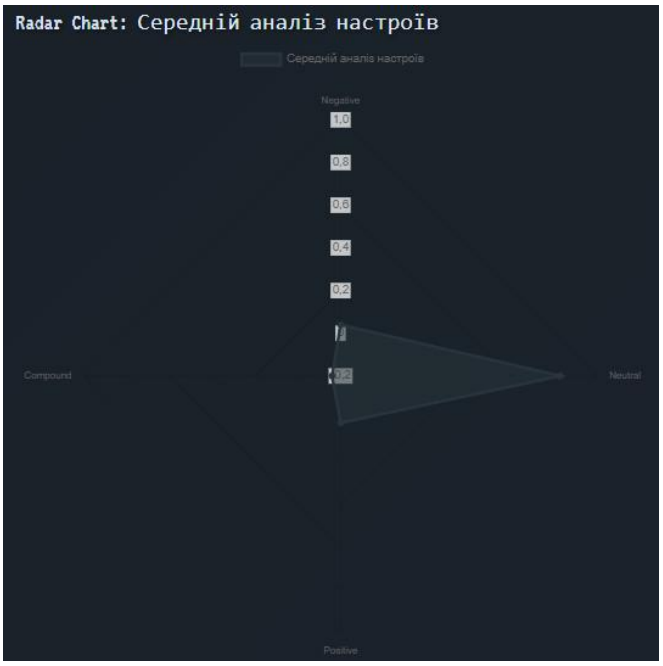

Рисунок 3.11 – Радарна діаграма середнього аналізу емоційних тональностей текстів

Окрему увагу варто звернути на розподіл рівнів загроз, який виводиться у вигляді кругової діаграми (рис. 3.12). Вона дозволяє визначити, яка частка джерел має високий, середній або низький рівень загрози. Така візуалізація є ключовою для формування загального уявлення про ризиковість інформаційного простору на основі аналізованих даних.

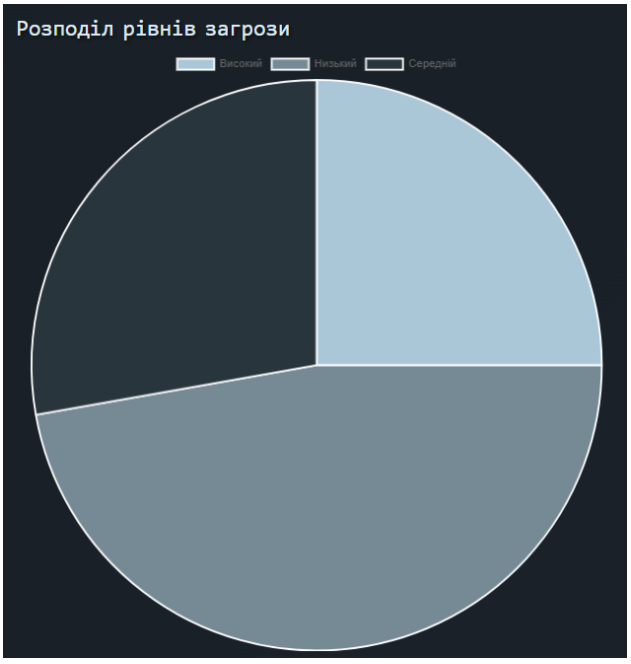

Рисунок 3.12 – Кругова діаграма розподілу рівнів загрози

З метою дослідження взаємозв’язку між кількісними характеристиками, інтерфейс також пропонує точкову діаграму (scatter plot) (рис. 3.13), яка ілюструє залежність між часом, витраченим на парсинг статті, та її довжиною. Завдяки цьому можна виявити аномальні записи, для яких парсинг тривав незвично довго або, навпаки, був надто швидким для великого обсягу даних.

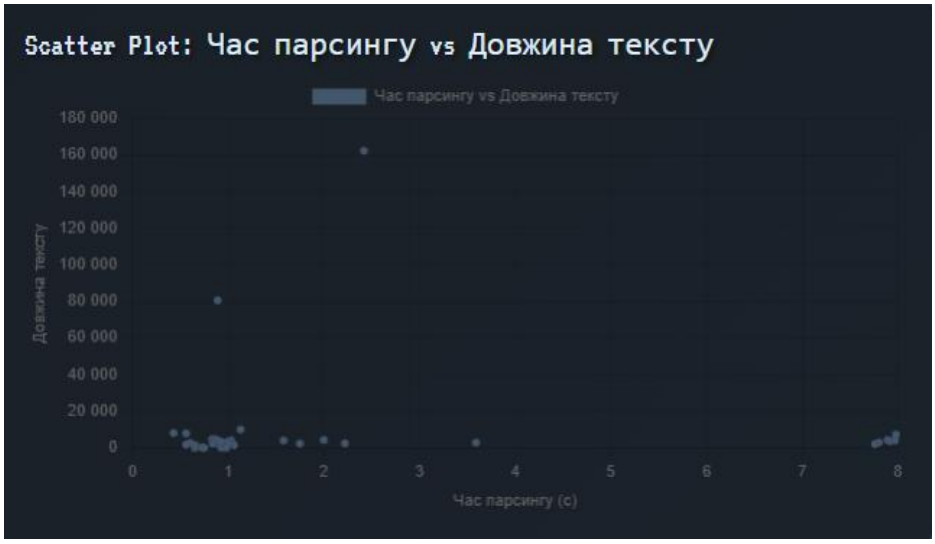

Рисунок 3.13 – Діаграма розсіювання: Час парсингу vs Довжина тексту

Таким чином, аналітична панель об’єднує як загальні метрики, так і деталізовану інформацію, що дозволяє ефективно інтерпретувати результати обробки та оцінювати потенційні кіберзагрози як у відокремлених текстах, так і у загальному інформаційному потоці.

Після завершення NLP-аналізу та візуалізації аналітичних результатів користувач може скористатися функціоналом кластеризації новинних повідомлень, що доступний на відповідній вкладці. Цей модуль автоматично виконує групування статей за семантичною подібністю на основі TF-IDF-представлення тексту та алгоритму K-Means [19]. Як наслідок, кожному документу присвоюється кластер з урахуванням схожості термінів.

На екрані відображається список ключових слів для кожного кластеру, що дозволяє швидко ідентифікувати тематику груп. Наприклад, у Кластері 1 фігурують такі слова, як hackers, ransomware, data, cyber, company, що свідчить

про кіберінциденти у корпоративному секторі. Нижче наведено гістограму (рис. 3.14), яка демонструє кількісний розподіл документів між кластерами, даючи можливість оцінити домінуючі теми.

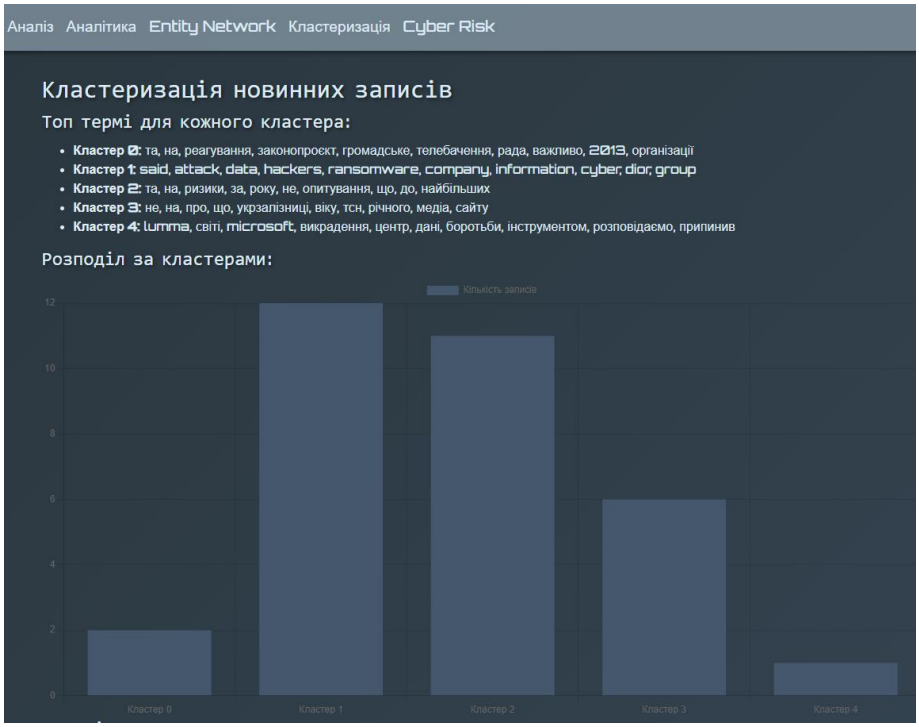

Рисунок 3.14 – Розподіл новинних повідомлень за кластерами з візуалізацією провідних термінів

Наступним логічним кроком є аналіз кіберризиків, що реалізується на вкладці «Cyber Risk». Цей розділ формує зведену таблицю (рис. 3.15), де кожен запис містить: URL джерела, заголовок, оцінку ризику (Risk Score), рівень загрози, а також емоційний тон тексту (Sentiment Compound).

Оцінка ризику розраховується на основі кількості виявлених загроз, їх імовірності та тональності тексту, що дозволяє пріоритезувати статті за ступенем потенційної небезпеки [3, 5]. Таким чином, аналітик може оперативно реагувати на критичні повідомлення з високим рівнем тривожності та високим ризик-індексом.

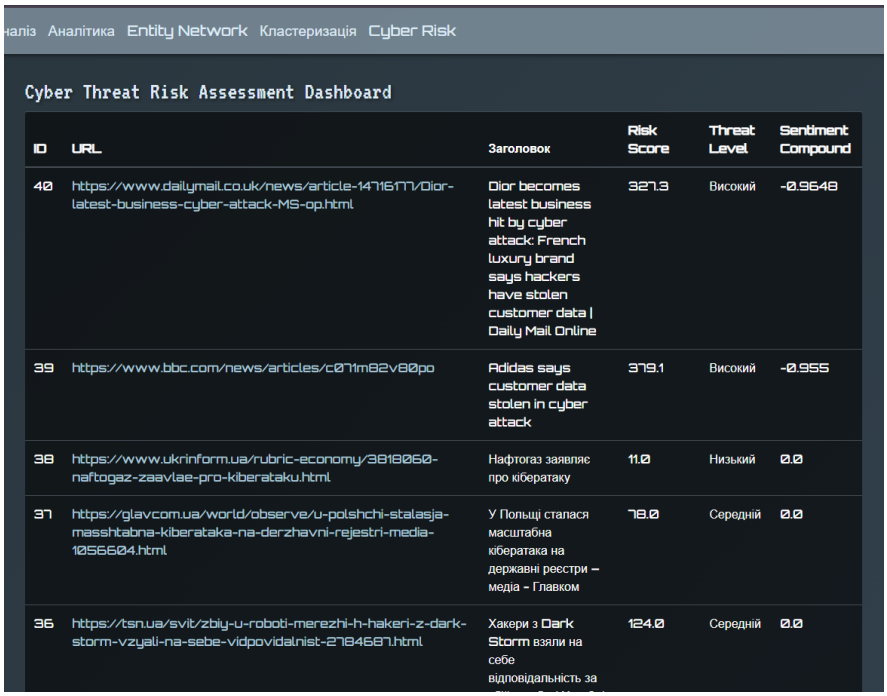

Рисунок 3.15 – Інтерфейс дашборду оцінки ризику на основі кіберзагроз

Висновки за розділом 3

У цьому розділі було представлено реалізацію веб-застосунку для OSINT-аналізу текстових джерел, що охоплює повний цикл — від збору відкритої інформації до її аналізу та візуалізації результатів.

Система побудована на основі фреймворку Flask із використанням бібліотек для NLP, класифікації та кластеризації даних. Збір текстів реалізовано за допомогою requests і BeautifulSoup, що забезпечує ефективне витягування інформації з новинних джерел. Зібрані дані зберігаються в SQLite, що спрощує подальший аналіз. Особливу увагу приділено модулям семантичної обробки: визначенню загроз, аналізу тональності, класифікації ризику та побудові кластерів. Результати виводяться у зручному веб-інтерфейсі, що доступний навіть користувачам без технічної підготовки.

Таким чином, реалізоване програмне забезпечення поєднує технічну ефективність із аналітичною гнучкістю та може використовуватись як інструмент для виявлення кіберзагроз у державному секторі.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було послідовно реалізовано комплекс завдань, спрямованих на розробку ефективного веб-інструменту для OSINT-аналізу текстових джерел з метою виявлення кіберзагроз у державному секторі.

По-перше, було проаналізовано потенціал використання технологій відкритої розвідки (OSINT) для проактивного виявлення кіберзагроз у публічному управлінні. Оскільки сучасні загрози мають складну природу та часто проявляються через відкриту інформацію задовго до їх реалізації, така форма моніторингу дозволяє своєчасно виявляти потенційно небезпечні події.

По-друге, на основі огляду доступних джерел було визначено релевантні типи текстової інформації, що містять індикатори загроз. Внаслідок цього реалізовано механізми автоматизованого збору таких даних, зокрема за допомогою парсингу та API-запитів, що забезпечило масштабованість і регулярність моніторингу.

По-третє, було вивчено методи обробки природної мови (NLP) і машинного навчання, які дозволяють здійснювати аналіз та класифікацію отриманих текстів. Ці методи інтегровано в інструмент для підвищення точності визначення контексту та оцінки ризиків, пов’язаних із виявленими повідомленнями.

Далі, на основі отриманих результатів, було розроблено функціональний веб-інструмент, що поєднує модулі збору, аналізу й візуалізації даних. Архітектура рішення включає компоненти Web Scraper, NLP Analyzer, Risk Calculator, Cluster Analyzer, а також інтуїтивно зрозумілий інтерфейс користувача, реалізований засобами Flask і підтриманий базою даних SQLite. Завдяки цьому забезпечено зручність роботи з інструментом як для технічних фахівців, так і для аналітиків.

Нарешті, було проведено тестування веб-застосунку на прикладах реальних OSINT-джерел. Результати тестування підтвердили стабільність роботи системи, її функціональність та здатність ефективно виконувати поставлені завдання в умовах реального інформаційного середовища.

Таким чином, усі поставлені в роботі завдання було виконано, мету дослідження досягнуто, а сама робота є завершеним практичним і теоретичним внеском у сферу моніторингу кіберзагроз у державному секторі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Д.В. Ланде. OSINT у кібербезпеці : навч. пос. / Ланде Д.В. – Київ: ТОВ «Інжиніринг», 2024. – 522 с. ISBN 978-966-2344-97-4

2. ISO/IEC 27032:2012. Information technology — Security techniques — Guidelines for cybersecurity, 2012. – 76 p.

3. National Institute National Institute of Standards & Technology. Guide for Conducting Risk Assessments: NIST SP 800-30 Rev 1. Independently Published, 2019.

4. CERT-UA. Загальні характеристики кіберзагроз у публічному секторі : аналітичний огляд. URL: https://cert.gov.ua/.

5. Mansfield-Devine S. Verizon: Data Breach Investigations Report. 2024. 100 p. URL: https://www.verizon.com/business/resources/reports/dbir/.

6. APT1: Exposing one of China's cyber espionage units. Mandiant, 2013. 76 p. URL: https://nsarchive.gwu.edu/sites/default/files/documents/2700170/Document-83.pdf.

7. Рада національної безпеки і оборони України: правові основи становлення та розвитку. Київ : РНБО України, 2023. 34 с.

8. Cheng R. An Analysis of the SolarWinds Supply Chain Breach via Attack Graphs. Pennsylvania State University, 2024. 53 p. URL: https://honors.libraries.psu.edu/files/final_submissions/9959.

9. Gill M., Hansen P. Strategic Communications Hybrid Threats Toolkit / ed. by B. Heap. NATO StratCom COE, 2021. 48 p. URL: https://stratcomcoe.org/publications/strategic-communications-hybrid-threats-toolkit/213.

10. Kavanagh J., Rich M. Truth Decay: An Initial Exploration of the Diminishing Role of Facts and Analysis in American Public Life. RAND Corporation, 2018. 326 p. URL: https://doi.org/10.7249/rr2314.

11. . Скрипнік О. Використання технології OSINT у науковій та аналітичній діяльності. Cybersecurity: Education, Science, Technique. 2023. № 2. С. 75–83.

12. Prokipchuk O., Vysotska V. UKRAINIAN LANGUAGE TWEETS ANALYSIS TECHNOLOGY FOR PUBLIC OPINION DYNAMICS CHANGE PREDICTION BASED ON MACHINE LEARNING. Radio Electronics, Computer Science, Control. 2023. No. 2. P. 103. URL: https://doi.org/10.15588/1607-3274-2023-2-11.

13. Haltiuk M., Smywiński-Pohl A. LiBERTa: advancing Ukrainian language modeling through pre-training from Scratch. Proceedings of the Third Ukrainian Natural Language Processing Workshop (UNLP) @ LREC-COLING 2024. 2024. P. 120–128. URL: https://aclanthology.org/2024.unlp-1.14.pdf.

14. Шевчук В., Бондаренко В., Марченко А. SED-UA-Small: українськомовний синтетичний набір даних для моделей текстових ембеддінгів. Вісник Національного університету «Львівська політехніка» «Інформаційні системи та мережі». 2025. № 17. С. 403–410. URL: https://science.lpnu.ua/uk/sisn/vsi-vypusky/vypusk-17-2025/sed-ua-small-ukrayinomovnyy-syntetychnyy-nabir-danyh-dlya-modeley.

15. Розробка веб застосунку звикористанням фреймворку Flask та графічної бібліотеки Folium. Києво-Могилянська академія. URL: https://ekmair.ukma.edu.ua/server/api/core/bitstreams/8a2cc4c0-01e6-4052-856e-0376c0083966/content.

16. Конструктор сторінок HTML на основі деревоподібної структури даних. Києво-Могилянська академія. URL: https://ekmair.ukma.edu.ua/server/api/core/bitstreams/db1a21f3-649c-4210-8b4d-ba5dcd877763/content.

17. Flask Documentation. Flask. URL: https://flask.palletsprojects.com/en/stable/.

18. Natural Language Toolkit. NLTK. URL: https://www.nltk.org/.

19. Перші кроки в NLP: розглядаємо Python-бібліотеку NLTK в реальному завданні. URL: https://dou.ua/lenta/articles/first-steps-in-nlp-nltk/.

20. Industrial-strength Natural Language Processing in Python. spaCy. URL: https://spacy.io/.

21. SQLite Documentation. SQLite. URL: https://www.sqlite.org/docs.html.

22. HTML: HyperText Markup Language. MDN Web Docs. URL: https://developer.mozilla.org/en-US/docs/Web/HTML.

23. JavaScript documentation. DevDocs. URL: https://devdocs.io/javascript/.

24. JavaScript Guide. MDN Web Docs. URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide.

25. CSS documentation. DevDocs. URL: https://devdocs.io/css/.

26. Парсинг Python та його використання у різних бізнес-задачах. FoxmindEd. URL: https://foxminded.ua/parsynh-python/.

27. Що таке парсинг сайтів і для чого це потрібно?. WEDEX. URL: https://wedex.com.ua/blog/shho-take-parsyng-sajtiv-i-dlya-chogo-cze-potribno/.

28. Пономаренко О. Як вебскрейпінг допомагає збирати дані. Основні технології, бібліотеки та інструменти для вебскрейпінгу. ProIT. URL: https://proit.ua/iak-viebskrieipingh-dopomaghaie-zbirati-dani-osnovni-tiekhnologhiyi-bibliotieki-ta-instrumienti-dlia-viebskrieipinghu/.

29. Dakova L., Levytska M., Havenko K. Usage of open-source intelligence for security of critical infrastructure. Scientific journal Tras Shevchenko National university of Kyiv. Information systems and technologies security. 2024. Т. 2, вип. 8. С. 49–55. ISSN 2707-1758. URL: https://doi.org/10.17721/ists.2024.8.49-55.

30. Open Source Intelligence (OSINT): Issues for Congress. 27 p. URL: https://digital.library.unt.edu/ark:/67531/metadc819267/.

31. Harding T. Open Source Intelligence Techniques: Resources for Searching and Analyzing Online Information. Independently Published, 2019. 534 p.

32. Bazzell M. Open Source Intelligence Techniques: Resources for Searching and Analyzing Online Information. Independently Published, 2022. 526 p. URL: https://anyflip.com/mhnd/przi/basic.

33. Lande D. V. OSINT as a part of cyber defense system. Theoretical and Applied Cybersecurity. 2019. P. 103–108. URL: https://doi.org/10.20535/tacs.2664-29132019.1.169091.

34. Nonum E. O., Avwokuruaye O., Ezemonye T. M. Role of Open Source Intelligence (OSINT) in Cybersecurity and Threat Analysis. International Journal of Latest Technology in Engineering Management & Applied Science. 2025. Т. 14, вип. 2. С. 189–200. URL: https://doi.org/10.51583/IJLTEMAS.2025.140300023.

ДОДАТКИ

Додаток А
Апробація результатів дослідження

Dakova L., Levytska M., Havenko K. Usage of open-source intelligence for security of critical infrastructure. Scientific journal Tras Shevchenko National university of Kyiv. Information systems and technologies security. 2024. Т. 2, вип. 8. С. 49–55. ISSN 2707-1758. URL: https://doi.org/10.17721/ists.2024.8.49-55.

Додаток Б

Програмний код веб-застосунку для OSINT-аналізу текстових джерел

```
import os
import time
from datetime import datetime
from concurrent.futures import ThreadPoolExecutor, as_completed
from flask import Flask, render_template, request, redirect, url_for
from flask_sqlalchemy import SQLAlchemy
import requests
from bs4 import BeautifulSoup
import nltk
from nltk.tokenize import sent_tokenize
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from langdetect import detect, LangDetectException
import spacy
nltk.download('punkt', quiet=True)
nltk.download('vader_lexicon', quiet=True)
nltk.download('punkt_tab')
nlp_model_spacy = spacy.load("en_core_web_sm")
sia = SentimentIntensityAnalyzer()

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.cluster import KMeans

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///osint.db'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
db = SQLAlchemy(app)

class ScrapedData(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    url = db.Column(db.Text, nullable=False)
    title = db.Column(db.Text)
    content = db.Column(db.Text)
    scrape_duration = db.Column(db.Float)
    num_paragraphs = db.Column(db.Integer)
    timestamp = db.Column(db.DateTime, default=datetime.utcnow)

with app.app_context():
    db.create_all()
```

Продовження додатку Б

```
THREAT_KEYWORDS_EN = [
    "cyber attack", "data breach", "leak", "malware", "ransomware", "spyware",
"phishing", "ddos", "denial of service",
    "brute force", "sql injection", "zero day", "exploit", "backdoor", "trojan", "worm",
"keylogger", "rootkit",
    "APT", "hacker", "cybercriminal", "hacktivist", "surveillance", "eavesdropping",
"man-in-the-middle", "unauthorized access",
    "compromise", "intrusion", "cyber threat", "cyber espionage", "cyber warfare",
"state-sponsored", "infiltration",
    "critical infrastructure", "government systems", "cyber sabotage", "social
engineering", "credential theft", "identity theft",
    "exfiltration", "vulnerability", "security hole", "breach", "network attack", "host
compromise", "malicious payload",
    "threat actor", "cyber intelligence", "dark web", "botnet", "command and control",
"C2 server", "payload", "cybercrime"
]
THREAT_KEYWORDS_UK = [
    "кібератака", "злом", "витік даних", "шкідливе ПЗ", "вірус", "троян",
"черв’як", "кейлогер", "руткіт", "бекдор",
    "фішинг", "шпигунське ПЗ", "вимагач", "шкідник", "SQL-ін’єкція", "атака
відмови в обслуговуванні", "ddos", "уразливість",
    "експлойт", "нульовий день", "злом пароля", "несанкціонований доступ",
"перехоплення", "міжлюдська атака", "зловмисник",
    "кіберзлочинець", "хакер", "кіберзагроза", "кібершпигунство", "кібервійна",
"державний хакінг", "інфільтрація",
    "урядові системи", "критична інфраструктура", "соціальна інженерія",
"крадіжка облікових даних", "крадіжка особистості",
    "експортування даних", "дірка безпеки", "ботнет", "сервер управління",
"шкідливе навантаження", "темна мережа",
    "кіберрозвідка", "злом державного ресурсу", "хактивіст", "спроба
проникнення", "несанкціоноване втручання"
]
def scrape_single(url):
    start_time = time.time()
    try:
        response = requests.get(url.strip(), timeout=10)
        soup = BeautifulSoup(response.content, 'html.parser')
        title = soup.title.string.strip() if soup.title and soup.title.string else 'No Title'
        paragraphs = soup.find_all('p')
        content = "\n".join([p.get_text().strip() for p in paragraphs if p.get_text().strip()])
        num_paragraphs = len(paragraphs)
    except Exception as e:
```

Продовження додатку Б

```
        title = 'Error'
        content = ''
        num_paragraphs = 0
    duration = round(time.time() - start_time, 2)
    return {
        'url': url.strip(),
        'title': title,
        'content': content,
        'scrape_duration': duration,
        'num_paragraphs': num_paragraphs
    }

def analyze_text(text):
    try:
        lang = detect(text)
    except LangDetectException:
        lang = "en"
    if lang.startswith("uk"):
        threat_keywords = THREAT_KEYWORDS_UK
    else:
        threat_keywords = THREAT_KEYWORDS_EN
    flagged_sentences = []
    keyword_counts = {kw: 0 for kw in threat_keywords}
    sentences = sent_tokenize(text)
    total_sentences = len(sentences) if sentences else 1
    for sentence in sentences:
        sentence_flagged = False
        for keyword in threat_keywords:
            count = sentence.lower().count(keyword.lower())
            if count > 0:
                keyword_counts[keyword] += count
                sentence_flagged = True
        if sentence_flagged:
            flagged_sentences.append(sentence)
    threat_count = len(flagged_sentences)
    probability = round(threat_count / total_sentences, 2)
    if probability >= 0.3:
        threat_level = "Високий"
    elif probability >= 0.1:
        threat_level = "Середній"
    else:
        threat_level = "Низький"
```

Продовження додатку Б

```
    sentiment_scores = sia.polarity_scores(text)
    entities = []
    if not lang.startswith("uk"):
        doc = nlp_model_spacy(text)
        entities = [(ent.text, ent.label_) for ent in doc.ents]
    return {
        'flagged_sentences': flagged_sentences,
        'threat_count': threat_count,
        'probability': probability,
        'threat_level': threat_level,
        'sentiment': sentiment_scores,
        'keyword_counts': keyword_counts,
        'entities': entities,
        'language': lang
    }

def compute_risk(analysis):
    risk = analysis['threat_count'] * analysis['probability'] * 100
    if analysis['threat_level'] == "Високий":
        risk += 20
    elif analysis['threat_level'] == "Середній":
        risk += 10
    else:
        risk += 5
    if analysis['sentiment']['compound'] < 0:
        risk += abs(analysis['sentiment']['compound']) * 20
    return round(risk, 2)

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/scrape', methods=['POST'])
def scrape():
    urls_input = request.form.get('urls')
    if not urls_input:
        return redirect(url_for('index'))
    urls = [url.strip() for url in urls_input.split(',') if url.strip()]
    scraped_results = []
    with ThreadPoolExecutor(max_workers=5) as executor:
        future_to_url = {executor.submit(scrape_single, url): url for url in urls}
```

Продовження додатку Б

```
        for future in as_completed(future_to_url):
            result = future.result()
            scraped_results.append(result)
            record = ScrapedData(
                url=result['url'],
                title=result['title'],
                content=result['content'],
                scrape_duration=result['scrape_duration'],
                num_paragraphs=result['num_paragraphs']
            )
            db.session.add(record)
        db.session.commit()
    total_time = round(sum([res['scrape_duration'] for res in scraped_results]), 2)
    return render_template('scrape_progress.html', results=scraped_results,
total_time=total_time)

@app.route('/nlp')
def nlp():
    records = ScrapedData.query.order_by(ScrapedData.timestamp.desc()).all()
    analyses = []
    for record in records:
        analysis = analyze_text(record.content)
        analyses.append({
            'id': record.id,
            'url': record.url,
            'title': record.title,
            'scrape_duration': record.scrape_duration,
            'num_paragraphs': record.num_paragraphs,
            'content_length': len(record.content),
            'threat_count': analysis['threat_count'],
            'probability': analysis['probability'],
            'threat_level': analysis['threat_level'],
            'sentiment_compound': analysis['sentiment']['compound'],
            'flagged_sentences': analysis['flagged_sentences'],
            'entities': analysis['entities'],
            'language': analysis['language']
        })
    return render_template('nlp.html', analyses=analyses)

@app.route('/nlp/<int:record_id>')
def nlp_detail(record_id):
    record = ScrapedData.query.get_or_404(record_id)
```

Продовження додатку Б

```
    analysis = analyze_text(record.content)
    return render_template('nlp_detail.html', record=record, analysis=analysis)

@app.route('/delete/<int:record_id>', methods=['GET', 'POST'])
def delete_record(record_id):
    record = ScrapedData.query.get_or_404(record_id)
    db.session.delete(record)
    db.session.commit()
    return redirect(url_for('nlp'))

@app.route('/analytics', methods=['GET', 'POST'])
def analytics():
    start_date = request.values.get('start_date')
    end_date = request.values.get('end_date')
    threat_filter = request.values.get('threat_filter', 'All')
    query = ScrapedData.query
    if start_date:
        try:
            start_dt = datetime.strptime(start_date, '%Y-%m-%d')
            query = query.filter(ScrapedData.timestamp >= start_dt)
        except ValueError:
            pass
    if end_date:
        try:
            end_dt = datetime.strptime(end_date, '%Y-%m-%d')
            query = query.filter(ScrapedData.timestamp <= end_dt)
        except ValueError:
            pass
    records = query.all()
    filtered_records = []
    for record in records:
        analysis = analyze_text(record.content)
        if threat_filter == "All" or analysis['threat_level'] == threat_filter:
            filtered_records.append((record, analysis))
    text_lengths = [len(rec.content) for rec, _ in filtered_records]
    title_lengths = [len(rec.title) for rec, _ in filtered_records if rec.title]
    aggregated_keyword_counts = {}
    threat_levels = {'Низький': 0, 'Середній': 0, 'Високий': 0}
    scatter_data = []
    total_sentiments = {'neg': 0, 'neu': 0, 'pos': 0, 'compound': 0}
    count_sentiments = 0
    sentiment_compounds = []
```

Продовження додатку Б

```
    for rec, analysis in filtered_records:
        for kw, count in analysis['keyword_counts'].items():
            aggregated_keyword_counts[kw] = aggregated_keyword_counts.get(kw, 0) +
count
        threat_levels[analysis['threat_level']] += 1
        scatter_data.append({
            'duration': rec.scrape_duration,
            'content_length': len(rec.content)
        })
        s = analysis['sentiment']
        total_sentiments['neg'] += s['neg']
        total_sentiments['neu'] += s['neu']
        total_sentiments['pos'] += s['pos']
        total_sentiments['compound'] += s['compound']
        count_sentiments += 1
        sentiment_compounds.append(s['compound'])
    if count_sentiments > 0:
        avg_sentiment = {
            'neg': round(total_sentiments['neg'] / count_sentiments, 2),
            'neu': round(total_sentiments['neu'] / count_sentiments, 2),
            'pos': round(total_sentiments['pos'] / count_sentiments, 2),
            'compound': round(total_sentiments['compound'] / count_sentiments, 2)
        }
    else:
        avg_sentiment = {'neg': 0, 'neu': 0, 'pos': 0, 'compound': 0}
    return render_template('analytics.html',
                           records=[rec for rec, analysis in filtered_records],
                           text_lengths=text_lengths,
                           title_lengths=title_lengths,
                           keyword_counts=aggregated_keyword_counts,
                           threat_levels=threat_levels,
                           scatter_data=scatter_data,
                           avg_sentiment=avg_sentiment,
                           sentiment_compounds=sentiment_compounds,
                           start_date=start_date if start_date else "",
                           end_date=end_date if end_date else "",
                           threat_filter=threat_filter)

@app.route('/cyber_risk')
def cyber_risk():
    records = ScrapedData.query.order_by(ScrapedData.timestamp.desc()).all()
    risk_assessments = []
```

Продовження додатку Б

```
    for record in records:
        analysis = analyze_text(record.content)
        risk = compute_risk(analysis)
        risk_assessments.append({
            'id': record.id,
            'url': record.url,
            'title': record.title,
            'risk': risk,
            'threat_level': analysis['threat_level'],
            'sentiment_compound': analysis['sentiment']['compound']
        })
    return render_template('cyber_risk.html', risk_assessments=risk_assessments)

@app.route('/network')
def network():
    records = ScrapedData.query.all()
    entity_occurrence = {}
    for record in records:
        analysis = analyze_text(record.content)
        if analysis['language'].startswith("en"):
            entities = [ent[0] for ent in analysis['entities']]
            unique_entities = list(set(entities))
            for i in range(len(unique_entities)):
                for j in range(i+1, len(unique_entities)):
                    pair = tuple(sorted([unique_entities[i], unique_entities[j]]))
                    entity_occurrence[pair] = entity_occurrence.get(pair, 0) + 1
    nodes_set = set()
    for pair in entity_occurrence:
        nodes_set.add(pair[0])
        nodes_set.add(pair[1])
    nodes = [{'data': {'id': n, 'label': n}} for n in nodes_set]
    edges = [{'data': {'source': pair[0], 'target': pair[1], 'weight':
entity_occurrence[pair]}}
             for pair in entity_occurrence]
    return render_template('network.html', nodes=nodes, edges=edges)

@app.route('/clustering', endpoint='clustering')
def clustering_view():
    records = ScrapedData.query.all()
    texts = [record.content for record in records if record.content.strip() != ""]
    if not texts:
        return "Немає даних для кластеризації."
```

Продовження додатку Б

```
    vectorizer = TfidfVectorizer(stop_words='english')
    X = vectorizer.fit_transform(texts)
    k = 5
    model = KMeans(n_clusters=k, random_state=42)
    model.fit(X)
    labels = model.labels_
    order_centroids = model.cluster_centers_.argsort()[:, ::-1]
    terms = vectorizer.get_feature_names_out()
    cluster_top_terms = {}
    for i in range(k):
        top_terms = [terms[ind] for ind in order_centroids[i, :10]]
        cluster_top_terms[i] = top_terms
    clustering_results = []
    j = 0
    for record in records:
        if record.content.strip() != "":
            clustering_results.append({
                'id': record.id,
                'url': record.url,
                'title': record.title,
                'cluster': int(labels[j])
            })
            j += 1
    cluster_counts = {i: 0 for i in range(k)}
    for label in labels:
        cluster_counts[int(label)] += 1
    return render_template('clustering.html',
                           clustering_results=clustering_results,
                           cluster_top_terms=cluster_top_terms,
                           cluster_counts=cluster_counts)
if __name__ == '__main__':
    app.run(debug=True)
```