Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

Магістерська кваліфікаційна робота на тему:
«Система виявлення інформаційних вкидів під час інформаційної війни»

Виконав: студент 2 курсу групи 2БС-22м
спеціальності 125 Кібербезпека
_____Богдан П'ЯТАК

Керівник: к. т. н., доцент каф. ЗІ
_____Олеся Войтович
«__13__»_____12_____2023 р.

Опонент: к. т. н., доцент каф. ПЗ
_____Олександр ХОШАБА
«__13__»_____12_____2023 р.

Допущено до захисту
Завідувач кафедри ЗІ
д. т. н., проф.
_____Володимир ЛУЖЕЦЬКИЙ
«__14__»_____12_____2023 р.

Вінниця 2023

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації
Рівень вищої освіти II (магістерський)
Галузь знань – 12 Інформаційні технології
Спеціальність – 125 Кібербезпека
Освітньо-професійна програма – Безпека інформаційних і комунікаційних систем

ЗАТВЕРДЖУЮ
Завідувач кафедри ЗІ,
д. т. н., проф.
_____Володимир ЛУЖЕЦЬКИЙ
«__19__» __09_____2023 року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

П'ятаку Богдану Олеговичу

1. Тема роботи: «Система виявлення інформаційних вкидів під час інформаційної війни»
керівник роботи: Войтович Олеся Петрівна, к. т. н., доцент кафедри ЗІ,
затверджені наказом ректора ВНТУ №247 від 18.09.2023р.

2. Строк подання студентом роботи: 13 грудня 2023р.

3. Вихідні дані до роботи:
- поняття інформаційних вкидів;
- мова програмування JavaScript;
- тип програми – телеграм-бот.

4. Зміст текстової частини: Вступ. 1. Аналіз методів та способів виявлення інформаційних вкидів. 2. Метод виявлення інформаційних вкидів під час інформаційної війни 3. Розробка системи виявлення інформаційних вкидів під час інформаційної війни 4. Тестування системи виявлення інформаційних вкидів. 5. Економічна частина. Висновки. Перелік використаних джерел. Додатки.

5. Перелік ілюстративного матеріалу: Узагальнена архітектура системи (плакат А4). Схема роботи модулю взаємодії з користувачем (плакат А4). Схема роботи модулю парсингу(плакат А4). Схема роботи модулю запису в базу даних (плакат А4). Схема роботи модулю аналізу (плакат А4). Схема роботи модулю факт-

чекінгу (плакат А4). Токенізація тексту (плакат А4). Правила визначення значущості тексту (плакат А4). Результати тестування (плакати А4).

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Олеся ВОЙТОВИЧ, к. т. н., доц. каф. ЗІ	19.09	26.09
2	Олеся ВОЙТОВИЧ, к. т. н., доц. каф. ЗІ	19.09	28.09
3	Олеся ВОЙТОВИЧ, к. т. н., доц. каф. ЗІ	19.09	21.10
4	Олеся ВОЙТОВИЧ, к. т. н., доц. каф. ЗІ	19.09	4.10
5	Ольга РАТУШНЯК, к.т.н., доц.каф ЕПВМ	19.09	

7. Дата видачі завдання 1вересня 2023р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз завдання. Вступ	01.09.2023 – 10.09.2023	
2	Аналіз літературних джерел за напрямком магістерської роботи	10.09.2023 – 15.09.2023	
3	Розробка рішень	16.09.2023 – 22.09.2023	
4	Розробка модуля програмного засобу	30.09.2023 – 12.10.2023	
5	Практична реалізація, моделювання, експериментування, результати	14.10.2023 – 10.11.2023	
6	Розробка розділу тестування і обґрунтування доцільності розробки	11.11.2023 – 17.11.2023	
7	Аналіз виконання, висновки	18.11.2023 – 24.11.2023	
8	Оформлення пояснювальної записки	25.11.2023 – 30.11.2023	
9	Попередній захист та доопрацювання МКР	28.11.2023 – 10.12.2023	
10	Представлення МКР до захисту	11.12.2023 – 14.12.2023	
11	Захист МКР	14.12.2023 – 21.12.2023	

Студент_____Богдан П’ЯТАК

Керівник роботи_____Олеся Войтович

АНОТАЦІЯ

Магістерська кваліфікаційна робота складається з 83 сторінок формату А4, на яких є 26 рисунків, 8 таблиць, 27 формул, список використаних джерел містить 26 найменувань.

Магістерська кваліфікаційна робота присвячена розробці системи виявлення інформаційних вкидів під час інформаційної війни. У роботі проаналізовано сучасні засоби захисту від впливу інформаційних вкидів, а також методи їх виявлення та аналізу. У межах роботи виконано теоретико-множинний опис методу виявлення та описано метод аналізу. Спроектовано архітектуру системи, а також описано основні алгоритми роботи. Після розробки схем функціонування програмного засобу в цілому і алгоритмів його окремих складових здійснено програмну реалізацію. Проведено тестування системи на коректність роботи в реальних умовах. Виконані економічні розрахунки показали доцільність розробки.

Ключові слова: інформаційний вкид, інформаційна війна, система виявлення, маніпуляція.

ABSTRACT

The master's qualification work consists of 83 pages of A4 format, on which there are 26 figures, 8 tables, 27 formulas, the list of used sources contains 26 titles.

The master's thesis is devoted to the development of a system for detecting information leaks during an information war. The work analyzes modern means of protection against the influence of information leaks, as well as methods of their detection and analysis. Within the framework of the work, a theoretical-multiple description of the detection method is performed and the analysis method is described. The system architecture is designed, and the main work algorithms are also described. After the development of schemes for the functioning of the software tool as a whole and the algorithms of its individual components, software implementation was carried out. The system was tested for correctness of operation in real conditions.

Key words: information attack, information war, detection system, manipulation.

ЗМІСТ

ВСТУП

В сучасному світі, де інформація є однією з ключових сировин для формування громадської думки, впливу на політичні рішення, та загалом для керування суспільством, інформаційні війни стають дедалі більш актуальними і загрожуючими явищами [1]. Відкритість та доступність інформації в інтернеті та суспільній мережі призвела до нового виду конфлікту, що зазвичай називається інформаційною війною. Сутність інформаційної війни полягає в тому, що не лише військові конфлікти, але й політичні, економічні та ідеологічні суперництва відбуваються на полях інформаційного простору[2]. Ця форма конфлікту визначається використанням інформаційних засобів та комунікаційних платформ для поширення дезінформації, маніпуляції громадською думкою та впливу на політичні та суспільні процеси.

Спроможність розрізняти правдиву інформацію від інформаційних вкидів та маніпулятивних повідомлень стала надзвичайно важливою в сучасному світі, де інформаційна війна може стати загрозою для демократії, стабільності та безпеки[3]. Системи виявлення інформаційних вкидів стають ефективним інструментом для виявлення та аналізу таких впливових спроб, спрямованих на дезорієнтацію суспільства та маніпуляцію інформацією.

На цьому етапі історії людства важливо розуміти, як інформаційні війни можуть впливати на політичні рішення, суспільні відносини та міжнародні конфлікти[4]. Однак наукове дослідження та розробка ефективних систем виявлення інформаційних вкидів можуть допомогти суспільству реагувати на ці загрози та зміцнювати свою інформаційну безпеку.

Дослідження, присвячене методам та засобам виявлення інформаційних вкидів під час інформаційної війни, представляє істотний прорив у сфері інформаційної безпеки та управління інформаційними ресурсами. Воно вирішує два ключові завдання. По-перше, це саме виявлення дезінформації та фейків. Це означає, що дослідження допомагає виявити і відфільтрувати дезінформацію,

фейки та маніпуляцію з метою збереження інформаційної чистоти та запобігання їхньому впливу на громадську думку. По-друге, допомагає вчасно виявляти нові тренди та методи ведення інформаційної війни. Це в свою чергу, дозволить адаптувати заходи безпеки та контрзаходи.

Магістерська кваліфікаційна робота присвячена темі виявлення інформаційних вкидів під час інформаційної війни та спрямована на розробку та аналіз інструментів та методологій для виявлення та аналізу інформаційних загроз. Вона має на меті вдосконалити здатність відстоювати інформаційну безпеку та допомогти суспільству розуміти та ефективно реагувати на ці загрози.

Об’єктом дослідження є забезпечення кібербезпеки під час інформаційної війни.

Предметом дослідження є методи виявлення інформаційних вкидів.

Метою цієї роботи є покращення кібербезпеки суспільства шляхом створення системи виявлення інформаційних вкидів у кіберпросторі.

Для досягнення мети потрібно розв’язати такі задачі:

– проаналізувати проблеми виявлення інформаційних вкидів;
– проаналізувати методи виявлення;
– розробити метод виявлення;
– розробити архітектуру модуля виявлення;
– розробити алгоритми роботи засобу.

Наукова новизна роботи: удосконалено метод виявлення інформаційних вкидів шляхом використання моделей обробки природньої мови, методів семантичного аналізу та аналізу контексту, яка працює в реальному часі, що дозволяє оперативно виявляти інформаційні вкиди.

Практична цінність: розроблено програмний засіб для виявлення інформаційних вкидів під час інформаційної війни;

Результати отримані у магістерській кваліфікаційній роботі були представлені на таких конференціях: Молодь у науці: дослідження, проблеми, перспективи 2024 року, Вінницького національного технічного університету [5].

1 АНАЛІЗ МЕТОДІВ ТА СПОСОБІВ ВИЯВЛЕННЯ ІНФОРМАЦІЙНИХ ВКИДІВ

1.1 Аналіз систем виявлення інформаційних вкидів

В сучасному інформаційному суспільстві інформаційна війна стала надзвичайно актуальною проблемою. Загрози, пов'язані з інформаційною війною, включають дезінформацію, фейкові новини, кібератаки та інші інструменти, що використовуються для впливу на громадську думку, політичний процес, економіку та загалом на стабільність суспільства.

Аналіз систем виявлення інформаційних вкидів є важливою складовою стратегій захисту від цих загроз. Впровадження ефективних систем виявлення інформаційних вкидів важливо не лише для захисту від впливу ззовні, але і для збереження інформаційної чистоти та дотримання принципів інформаційної безпеки.

NewsGuard - це розширення для браузера та мобільний додаток, розроблений компанією NewsGuard Technologies [6]. Його основна мета полягає в наданні користувачам інформації про вірогідність та довіру новинним веб-сайтам, які вони відвідують. NewsGuard має на меті допомогти користувачам відрізняти надійні джерела інформації від ненадійних.

Основні функції та інформація про NewsGuard включають в себе:

– Детальна Оцінка - розширення надає докладну інформацію про кожен веб-сайт, включаючи його історію, власність та редакційні практики. Відзначає будь-які випадки поширення неправдивого чи оманливого контенту та те, чи дотримується веб-сайт етичних стандартів журналістики.

– «Людський Нагляд» - Однією з важливих рис NewsGuard є використання аналітиків для оцінки та класифікації веб-сайтів, а не повне покладання на автоматизовані алгоритми. Цей підхід дозволяє нюансовано оцінювати контент та практики кожного сайту.

– Прозорість - NewsGuard ставить перед собою завдання бути прозорим щодо свого методу та критеріїв класифікації веб-сайтів. Користувачі можуть отримувати детальні звіти про класифікацію кожного сайту, включаючи пояснення до призначеного кольорового коду.

– Кольорова Класифікація - NewsGuard використовує систему класифікації за кольором для оцінки веб-сайтів за їхньою вірогідністю та прозорістю. Зелені значки вказують на надійні джерела, тоді як червоні значки вказують на менш надійні або можливо упереджені (рис. 1.1).

Рисунок 1.1 – Система класифікації за кольором на NewsGuard

NewsGuard отримав як позитивну, так і критичну оцінку. Хоча він пропонує корисний інструмент для швидкої оцінки надійності джерел новин, деякі висловлюють обурення можливими упередженнями у класифікаціях та впливом таких рейтингів на медіа.

Ноаху - це інструмент для візуалізації поширення інформації в соціальних мережах, який допомагає користувачам відстежувати поширення новинних статей та інших інформаційних матеріалів [7]. Ноаху дозволяє користувачам розуміти, як швидко та широко поширюються певні теми та інформація в соціальних мережах, і як це може впливати на громадську думку.

Основні функції та характеристики Ноаху включають в себе:

– Візуалізація поширення інформації - Ноаху надає графіки та графи, які демонструють, як новини та інші матеріали поширюються в соціальних мережах. Це дозволяє користувачам легко спостерігати за тим, як інформація поширюється в мережі.

– Співставлення з фактами - Hoaxy дозволяє користувачам порівнювати інформацію, що поширюється в соціальних мережах, з фактами та перевіреними джерелами. Це допомагає розрізняти правдиву інформацію від дезінформації та фейків.

– Виявлення вірусних тем - Hoaxy може допомогти виявити теми та інформацію, які стали вірусними в соціальних мережах. Це корисно для вивчення того, які теми отримують значний публічний інтерес та увагу (рис. 1.2).

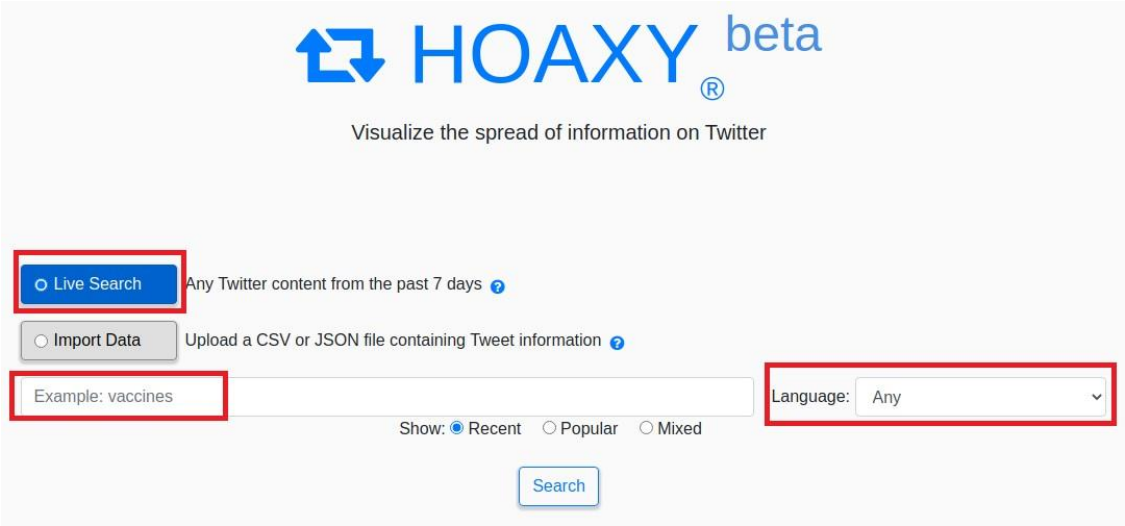

Рисунок 1.2 – Вигляд функції виявлення трендів в соціальній мережі Твіттер

– Моніторинг дезінформації та фейків - Hoaxy може бути використаний для виявлення та відстеження поширення дезінформації та фейків в соціальних мережах. Він допомагає користувачам бути обережними щодо недостовірної інформації.

– Дослідження впливу інформації - допомагає дослідникам та журналістам розуміти вплив інформації на громадську думку та суспільство.

Hoaxy допомагає користувачам бути більш обізнаними щодо того, як інформація поширюється в мережі та як це може впливати на суспільство та думку громадськості.

Snopes (також відомий як Snopes.com) - це популярний веб-сайт та факт-чекінгова платформа, яка спеціалізується на перевірці правдивості інформації та розкритті міфів, чуток, фейків і недостовірних стверджень. Заснований Девідом та Барбарою Міккельсон у 1994 році, Snopes став одним із найбільш впливових та надійних джерел факт-чекінгу в онлайн-просторі [8].

Основні функції :

– Факт-чекінг - Snopes працює над перевіркою різноманітних тверджень, легенд, чуток і новин з метою встановлення їхньої правдивості чи неправдивості. Вони ретельно досліджують інформацію, перевіряють факти, аналізують джерела та надають об'єктивну оцінку.

– Архів - Snopes зберігає великий архів факт-чекінгу, який можна використовувати для пошуку конкретних перевірених фактів або тем.

– Розбір сучасних новин - Сайт Snopes також розбирає сучасні новини і суспільно-політичні події з метою виявлення недостовірної інформації і роз'яснення складних тем (рис 1.3).

Рисунок 1.3 – Розбір сучасних новин від експертів Snopes.com

– Дослідження чуток та міфів - Snopes спеціалізується на розкритті міфів, які можуть поширювати недостовірну інформацію. Вони надають чіткі відповіді на питання про те, що правда і що ні.

– Об'єктивність та незалежність - сайт славиться своєю об'єктивністю та незалежністю. Вони стежать за стандартами журналістики та дотримуються високих професійних стандартів у роботі з фактами.

– Визнання впливу - Snopes впливовий факт-чекер і надійний джерело для журналістів, активістів, політиків та громадян, які шукають правдиву інформацію в цифровій епоці.

Snopes є важливим інструментом для боротьби з дезінформацією та фейками в інтернеті. Вони надають об'єктивний та документований аналіз інформації, що допомагає громадськості робити осмислений вибір та визначати правдивість новин та стверджень.

Програма перевірки фактів на Facebook - це ініціатива, запущена соціальною мережею для вирішення проблеми дезінформації та фейків на її платформі. Програма передбачає співпрацю з незалежними факт-чекінговими організаціями з метою оцінки достовірності вмісту, що розміщується на Facebook. Основна мета - зменшити поширення неправдивої або вводячої в оману інформації та надати користувачам більше точної інформації [9].

Основні функції та компоненти Програми перевірки фактів на Facebook включають в себе:

– Співпраця з факт-чекінговими організаціями - Facebook співпрацює з довіреними факт-чекінговими організаціями по всьому світу. Ці організації обираються на підставі їхньої репутації за точність та безпристрасність.

– Виявлення потенційної дезінформації - використовує автоматизовані інструменти та алгоритми для виявлення потенційно неправдивого або вводячого в оману вмісту на платформі. Ці інструменти можуть виявляти підозрілі патерни, такі як швидке поширення вмісту чи позначені ключові слова.

– Процес перевірки фактів - Коли виявляється потенційно неправдива інформація, вона відправляється на перевірку партнерським факт-чекінговим організаціям. Ці факт-чекери проводять глибокий аналіз тверджень, що містяться в контенті, та надають оцінку їхньої достовірності.

– Оцінка і позначення - Якщо факт-чекінгова організація встановлює, що вміст є неправдивим або вводячим в оману, він позначається як такий на Facebook. Користувачі, які зустрічають такий вміст, бачать попереджувальний прапор, що вказує на те, що вміст був перевірений і є неправдивим або вводячим в оману (рис.1.4)

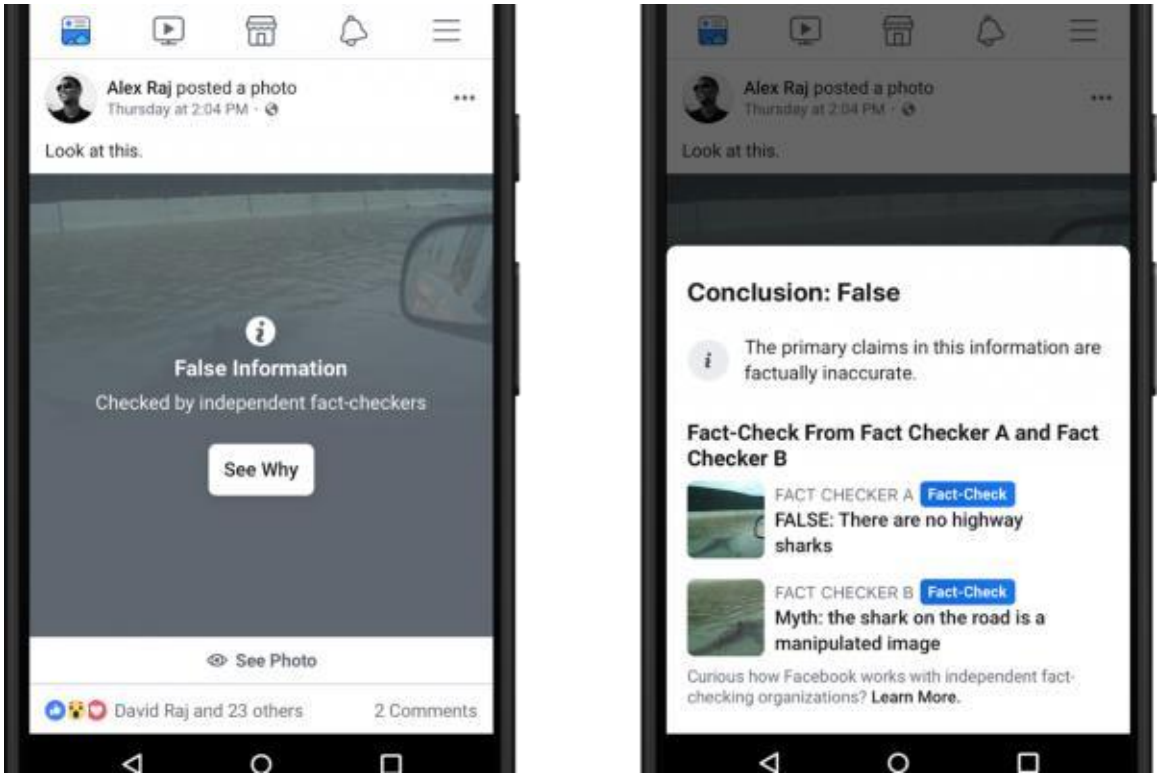

Рисунок 1.4 – Приклад позначення неправдивого вмісту в Фейсбук

– Зменшення поширення - Вміст, який був позначений як неправдивий або вводячий в оману та перевірений факт-чеками, може бачити зменшене поширення в стрічці новин користувачів. Це робиться з метою обмеження поширення та впливу такого вмісту.

– Вимкнення можливості монетизації реклами - У деяких випадках вміст, який позначено як неправдивий або вводячий в оману, може мати вимкнену можливість монетизації реклами, що не дає створювачам вмісту заробити на дезінформації.

– Прозорість - Facebook прагне бути прозорим у своїх зусиллях з перевірки фактів. Вони публікують регулярні звіти, в яких надають інформацію про

ефективність програми, вплив перевірки фактів на зменшення поширення дезінформації та інші відповідні метрики.

– Зворотний зв'язок та апеляції - Facebook дозволяє користувачам та створювачам вмісту надавати зворотний зв'язок та апелювати щодо рішень щодо перевірки фактів, якщо вони вважають, що рейтинг є неправильним.

B.S. Detector - це розширення для веб-браузерів, яке розроблено для виявлення неправдивої або маніпульованої інформації на веб-сайтах[10]. Цей інструмент призначений для допомоги користувачам бути більш обізнаними і критичними стосовно інформації, яку вони зустрічають у мережі.

Основні характеристики B.S. Detector :

– Піктограми Попередження - Розширення додає піктограми попередження біля заголовків статей або іншого контенту на веб-сайтах. Ці піктограми можуть вказувати на те, що інформація може бути неправдивою або маніпульованою.

– Джерела Факт-чекінгу - B.S. Detector використовує факт-чекінгові ресурси та бази даних для перевірки надійності інформації. Якщо вміст опиняється під сумнівом, розширення може вказати на це і надати користувачеві доступ до перевіреного джерела.

– Користувацькі Налаштування - Користувачі можуть налаштувати B.S. Detector згідно своїх власних вимог і визначити, які джерела факт-чекінгу вони хочуть використовувати для перевірки інформації.

– Сповіщення для Користувача - Якщо розширення виявляє неправдиву або сумнівну інформацію, воно може надсилати сповіщення користувачеві, щоб попередити його про можливу недостовірність.

Це розширення може бути корисним для тих, хто прагне зменшити вплив дезінформації та фейків у своєму онлайн-досвіді.

Bot Sentinel - це сервіс, який використовується для виявлення та ідентифікації підозрілих ботів та штучних інтелектів (AI) в соціальних мережах, зокрема на платформі Twitter. Цей сервіс розроблений з метою боротьби з недостовірними

акаунтами та автоматизованими системами, які можуть поширювати дезінформацію та маніпулювати громадською думкою.

Основні характеристики та функції Bot Sentinel включають в себе:

– Виявлення Підозрілих Акаунтів - Bot Sentinel використовує ряд алгоритмів та параметрів для визначення, наскільки підозрілим є конкретний акаунт на Twitter. Це включає аналіз активності, аватарів, поведінки та інших ознак.

– Рейтинг Акаунтів - Кожен акаунт отримує рейтинг від Bot Sentinel, що вказує на його рівень підозрілості. Акаунти можуть отримувати рейтинги від "найбільш підозрілих" до "найменш підозрілих" (рис. 1.5).

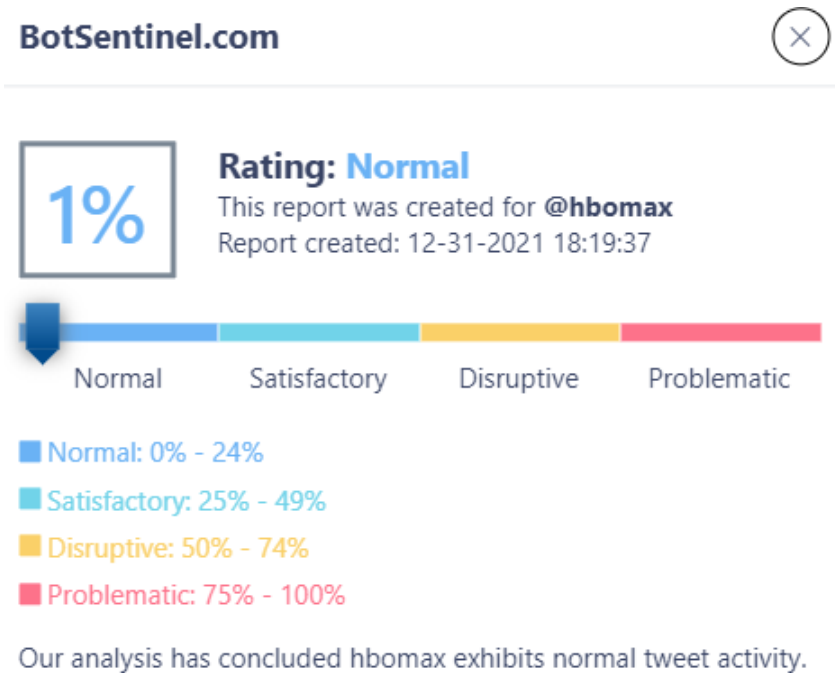

Рисунок 1.5 – Рейтинг аккаунта в Bot Sentinel

– Сповіщення та звіти - Користувачі можуть отримувати сповіщення про підозрілих акаунтах у своїй стрічці та отримувати звіти щодо активності підозрілих акаунтів.

– Аналіз Твітів - Bot Sentinel також аналізує твіти та ретвіти підозрілих акаунтів, щоб виявити можливі спроби поширення дезінформації чи маніпулювання громадською думкою.

– Спільнота Користувачів - Bot Sentinel розвиває спільноту користувачів, які спільно працюють над виявленням та забороною підозрілих акаунтів на Twitter.

Цей сервіс корисний для тих, хто бажає покращити якість дискусій та інформаційних обмінів на соціальних мережах, запобігти поширенню фейків та дезінформації, а також захистити від спроб маніпулювання громадською думкою шляхом застосування автоматизованих активностей. Bot Sentinel допомагає зробити соціальні мережі більш надійними для користувачів.

ClaimBuster - це інструмент, який використовує штучний інтелект для аналізу та ідентифікації фактичних тверджень у текстових матеріалах та новинах. Цей інструмент призначений для виявлення можливих неправдивих чи недостовірних заяв у текстах і допомагає журналістам, факт-чекерам і дослідникам визначити рівень достовірності текстового контенту.

Основні функції ClaimBuster :

– Аналіз Текстів - ClaimBuster використовує алгоритми обробки природної мови (NLP) для аналізу текстових матеріалів, таких як статті, новини, блоги та інші джерела інформації.

– Виявлення Фактичних Тверджень - Інструмент виявляє фактичні твердження та заяви, які містяться в тексті, і виділяє їх для подальшого аналізу.

– Оцінка Достовірності - ClaimBuster використовує різні параметри та методи для оцінки рівня достовірності фактичних тверджень. Це може включати перевірку фактів через сторонні джерела та контекстуальний аналіз.

– Позначення Підозрілих Тверджень - Якщо інструмент виявляє підозрілі або неправдиві твердження, він може позначити їх для подальшого розгляду або факт-чекінгу.

– Підтримка Журналістів і Дослідників - ClaimBuster призначений для підтримки журналістів, факт-чекерів і дослідників у роботі над виявленням та перевіркою недостовірної інформації в новинах та інших джерелах.

– Широкий Обсяг Застосувань - Цей інструмент може бути використаний у різних галузях, включаючи журналістику, факт-чекінг, наукові дослідження та контент-аналіз.

ClaimBuster допомагає підвищити якість інформації, яку споживачі отримують з різних джерел та зменшити поширення неправдивої або маніпульованої інформації.

Таблиця 1.1 – Порівняння інструментів для виявлення інформаційних вкидів

Характе-ристика	NewsGuard	Hoaxy	Snopes	B.S. Detector	Bot Sentinel	ClaimBuster
Основна функція	Оцінка надійності джерел	Виявлення та відстеження поширення	Факт-чекінг	Позначення недостовірного контенту	Виявлення підозрілих акаунтів	Виявлення недостовірних тверджень
Джерела факт-чекінгу	Незалежні журналістські організації, журналісти	Спільнота користувачів дослідники	Власні дослідження спільнота користувачів	Спільнота користувачів	Незалежні дослідники, спільнота користувачів	Незалежні джерела факт-чекінгу
Тип інформації	Веб-сайти, новини, інформаційні портали	Соціальні мережі, новини, твіти	Різний контент	Веб-сайти	Соціальні мережі	Текстовий контент
Спільнота користувачів	Так	Так	Так	Ні	Так	Ні
Наявність API	Так	Ні	Так	Ні	Ні	Так
Оцінка достовірності	Так	Ні	Так	Ні	Ні	Ні

У результаті аналізу різних інструментів для виявлення інформаційних вкидів, створено таблицю з порівнянням усіх цих засобів (таб. 1.1) та визначено, що на ринку існує багато різноманітних рішень, які відповідають різним потребам суспільства. NewsGuard підходить для оцінки надійності джерел, Hoaxy - для виявлення поширення інформації, FFCP і B.S.Detector – для перевірки та позначеннння недостовірного контенту, Snopes і ClaimBuster – безпосередньо факт-чекінг, а Bot Sentinel - для виявлення підозрілих акаунтів. Вибір інструменту залежить від конкретних потреб та завдань.

1.2 Аналіз методів та способів виявлення інформаційних вкидів

В сучасному інформаційному середовищі інформаційні вкиди, фейкові новини та інші форми дезінформації стали серйозною загрозою для суспільства та демократії [11]. Для боротьби з цими загрозами необхідно розробляти та вдосконалювати методи та засоби виявлення інформаційних вкидів.

Один із найпоширеніших методів виявлення інформаційних вкладів полягає в застосуванні методів машинного навчання та аналізу тексту[12]. Ці методи дозволяють аналізувати великі обсяги текстового контенту, ідентифікувати ключові слова та фрази, виявляти аномалії у структурі тексту та визначати ймовірність дезінформації.

Ці методи базуються на використанні різних алгоритмів машинного навчання для аналізу текстового контенту з метою виявлення дезінформації та фейків та включає в себе наступні ключові етапи:

– Аналіз текстового контенту. Алгоритми машинного навчання аналізують текстовий контент, включаючи новини, статті, соціальні мережі, блоги та інші джерела інформації. Аналіз може включати в себе виявлення ключових слів, тем та структури тексту.

– Створення навчальної моделі: Для навчання алгоритмів машинного навчання потрібна велика кількість навчальних даних, включаючи як достовірну, так і недостовірну інформацію. Навчальна модель використовується для визначення характерних ознак дезінформації та підказує алгоритму, як виявляти недостовірну інформацію.

– Визначення ознак дезінформації. Для виявлення дезінформації алгоритми шукають певні ознаки, які можуть свідчити про недостовірну інформацію. Це можуть бути несуттєві або суперечливі дані, відсутність джерел та посилань, використання заголовків-приманок, аномалії в структурі тексту тощо.

– Класифікація тексту. Після визначення ознак дезінформації алгоритм класифікує текст на дві категорії: достовірний і недостовірний. Для класифікації

використовуються різні алгоритми, такі як метод опорних векторів (SVM), нейронні мережі, дерева рішень тощо.

– Позначення недостовірної інформації. Якщо текст визнається як недостовірний, його можуть позначити або відмітити для подальшого аналізу та перевірки. Це може бути використано журналістами, факт-чекерами або іншими експертами для додаткового розслідування та підтвердження недостовірності інформації.

Методи машинного навчання для виявлення інформаційних вкладів є досить ефективними і можуть використовуватися для автоматизованого виявлення недостовірної інформації в реальному часі. Однак він вимагає наявності великої кількості навчальних даних та впровадження складних алгоритмів машинного навчання.

Аналіз джерел та поширення інформації. Цей метод зосереджений на аналізі джерел, які поширюють інформацію, та способів, якими ця інформація поширюється в інтернеті. Основні аспекти цього методу включають наступне:

Аналіз джерел інформації:

– Ідентифікація джерел, які поширюють інформацію, включаючи веб-сайти, соціальні мережі, блоги, форуми тощо.

– Оцінка репутації джерел та джерелаїв новин.

– Виявлення ступеня авторитетності та об'єктивності джерела.

– Моніторинг інформації, яка походить від певних джерел.

Виявлення ботів і автоматизованих активностей:

– Виявлення автоматизованих аккаунтів, ботів та інших штучних інтелектів, які можуть поширювати дезінформацію.

– Моніторинг активності цих аккаунтів та виявлення надмірної активності, яка може свідчити про спроби маніпуляції.

Аналіз мережевої топології:

– Вивчення структури мережі поширення інформації, включаючи посилання, репости та зв'язки між джерелами.

– Виявлення груп або індивідів, які спільно сприяють поширенню дезінформації.

– Виявлення паттернів в поширенні інформації, таких як швидкість поширення, географічні обмеження тощо.

Використання джерел та факт-чекінгу:

– Співпраця з факт-чекерами та незалежними джерелами перевірки фактів для підтвердження чи спростування інформації.

– Використання публічних баз даних та архівів для перевірки інформації та виявлення суперечливих даних.

– Моніторинг та виявлення аномалій:

–Систематичний моніторинг поширення інформації та виявлення аномалій, які можуть свідчити про дезінформацію.

–Виявлення надмірної активності, яка може бути показником маніпуляцій.

Цей метод допомагає виявляти дезінформацію та фейки, аналізуючи джерела та способи поширення інформації. Він може виявляти недостовірні джерела, аномалії в поширенні інформації та автоматизовані спроби маніпуляції. Метод особливо корисний для виявлення дезінформації в соціальних мережах та інтернет-просторі, де поширення інформації швидке і велике.

Колаборативні підходи до виявлення інформаційних вкладів базуються на залученні громадськості та фахівців. Вони включають в себе:

- Створення спільнот користувачів та дослідників, які спільно працюють над виявленням та аналізом дезінформації.

- Залучення факт-чекерів і журналістів для перевірки інформації та позначення недостовірних джерел.

- Використання публічних баз даних та архівів для порівняння інформації та виявлення суперечливих даних.

Цей метод виявлення інформаційних вкладів використовує сучасні технології та інструменти для підвищення достовірності та автентичності інформації. Він включає наступні ключові аспекти:

- Блокчейн. Використання технології блокчейн для створення децентралізованих систем, які гарантують недоторканність та цілісність інформації. Блокчейн дозволяє створити публічний реєстр, який неможливо підробити або змінити без відповідної авторизації.

- Цифрові підписи. Використання цифрових підписів для підтвердження автентичності інформації. Кожен документ чи повідомлення може бути підписано електронним підписом, що дозволяє перевірити, що він не був змінений після підписання.

- Криптографія. Використання методів криптографії для захисту інформації від несанкціонованого доступу та змін. Криптографічні методи дозволяють шифрувати та розшифровувати інформацію, забезпечуючи конфіденційність та цілісність даних.

- Децентралізовані медіа. Створення платформ та медіа, які базуються на децентралізованих принципах. Це означає, що інформація розміщується на різних серверах та джерелах, що ускладнює можливість маніпуляції та цензури.

- Цифрова ідентифікація. Використання цифрових методів ідентифікації для перевірки автентичності користувачів та джерел інформації. Це допомагає визначати, хто стоїть за конкретними повідомленнями чи новинами.

- Автоматизовані системи перевірки. Використання різних інструментів та програм для автоматизованої перевірки інформації на достовірність та недостовірність. Ці системи можуть виявляти суперечливу інформацію, підозрілі посилання та інші ознаки дезінформації.

Цей метод надає технологічні рішення для забезпечення недоторканості та автентичності інформації, ускладнюючи завдання впливу та маніпуляції над нею. Він дозволяє створювати більш захищені та надійні джерела інформації.

Метод розвитку антидезінформаційних стратегій спрямований на створення та вдосконалення підходів для запобігання і поширенню дезінформації. Цей метод включає в себе ряд дій та інструментів, які сприяють формуванню освітніх

програм, зміцненню медіаграмотності та посиленню зусиль для протидії дезінформації.

Основні аспекти цього методу включають:

- Освітні програми. Розробка та запровадження освітніх програм для громадян з метою навчання навичкам критичного мислення та аналізу інформації.

- Медіаграмотність. Підвищення рівня медіаграмотності серед населення, що допомагає людям розуміти, як працюють ЗМІ, як визнавати достовірні джерела та розпізнавати дезінформацію.

- Факт-чекінг. Підтримка та розвиток факт-чекінгових організацій та ініціатив, які перевіряють інформацію на достовірність та публікують результати своїх перевірок. Факт-чекінг допомагає виявляти та спростовувати дезінформацію.

- Сприяння відкритості та прозорості. Залучення урядових організацій, ЗМІ та інших стейкхолдерів до публікації достовірної інформації та відкритості в прийнятті рішень. Це сприяє зменшенню простору для поширення дезінформації.

- Інформаційні кампанії. Здійснення інформаційних кампаній, які надають громадянам доступ до достовірних джерел інформації та надихають до критичного мислення. Інформаційні кампанії можуть включати в себе відеоматеріали, плакати, веб-ресурси тощо.

- Міжнародне співробітництво. Розвиток співробітництва між країнами та міжнародними організаціями для обміну найкращими практиками та інформацією щодо протидії дезінформації.

Цей метод спрямований на запобігання поширенню дезінформації та формування відповідального ставлення до інформації серед громадян. Розвиток антидезінформаційних стратегій допомагає зміцнити інформаційну стійкість суспільства.

Аналіз методів та способів виявлення інформаційних вкидів демонструє широкий перелік доступних методів для вирішення проблеми з дезінформацією в інформаційних ресурсах.

1.3 Постановка задачі

Розробка нового засобу для виявлення дезінформації та фейків, у вигляді телеграм-боту, є вельми актуальним завданням в контексті боротьби з інформаційними вкидами під час інформаційної війни. Мета проекту полягає в створенні інтерактивної платформи, де користувачі зможуть перевіряти інформацію на наявність інформаційних вкидів, та разом з тим долучатися до перевірки.

Серед основних завдань цього проекту включається розробка інтерфейсу боту, можливість автоматичного та ручного аналізу інформації, інтеграція спільноти до перевірки інформації, створення механізмів для маркування невідповідного контенту.

Підсумовуючи розділ, присвячений систем виявлення інформаційних вкидів, методам їхньої роботи та постановці завдання для розробки нової системи виявлення інформаційних вкидів, відзначається важливість ретеального аналізу та перевірки інформації, що поширюється, оскільки це є важливим завданням для забезпечення інформаційної безпеки та інформованості громадян.

Виявлення інформаційних вкладів є складним завданням, і вимагає комплексного підходу, який включає в себе аналіз, технологічні рішення та спільнотні ініціативи.

Слід відзначити роль спільнот користувачів та експертів у виявленні та аналізі дезінформації, і спільні зусилля, які можуть бути надзвичайно ефективними. Використання сучасних технологій, таких як штучний інтелект та блокчейн, може підвищити надійність та автентичність інформації.

Вирішення проблеми інформаційної війни вимагає спільних зусиль суспільства, уряду та глобальної спільноти, і розвиток ефективних методів та засобів для виявлення дезінформації є ключовим елементом цього процесу.

2. ОБГРУНТУВАННЯ ВИБОРУ МЕТОДІВ ДЛЯ ВИЯВЛЕННЯ ІНФОРМАЦІЙНИХ ВКИДІВ

2.1 Математична модель виявлення інформаційних вкидів

В процесі виявлення інформаційних вкладів, математика грає ключову роль у створенні моделей та алгоритмів, які дозволяють визначити, яка інформація є недостовірною чи потенційно шкідливою. Моделі мови, статистичні методи та аналіз тексту - це лише декілька прикладів математичних інструментів, які використовуються для розуміння та ідентифікації інформаційних вкладів.

Однією з ключових задач виявлення інформаційних вкладів є виявлення аномалій у текстовому або числовому матеріалі[13]. Це може включати в себе виявлення незвичайних лексичних конструкцій, синтаксичних аномалій, використання надмірної кількості специфічних слів або фраз, а також виявлення неспівмірностей у розподілі даних.

Виявлення аномалій у тексті - це процес аналізу текстових даних з метою виявлення незвичайних або підозрілих особливостей, які можуть свідчити про наявність інформаційних вкидів чи дезінформації[14]. Однією з основних математичних технік для виявлення аномалій є використання статистики та ймовірності.

Важливою складовою є виявлення незвичайних лексичних конструкцій та фраз. Це може включати в себе пошук слів або фраз, які рідко зустрічаються у звичайних текстах або є незвичайними для даного контексту. Один із підходів до цього - використання статистики входження слів у текст та порівняння їх зі звичайними лексичними конструкціями.

З іншого боку, аналіз синтаксичних аномалій включає в себе перевірку синтаксичної структури речень та тексту загалом. Наприклад, деякі аномалії можуть включати надмірну вкладеність речень, незвичайні залежності між словами, або використання синтаксичних структур, які не типові для даної мови.

Ще однією важливою математичною технікою є аналіз статистичних розподілів у тексті[15]. Наприклад, можна визначити, які слова або фрази мають надмірно високу частоту вживання, яка відрізняється від типових розподілів. Це може вказувати на спробу маніпуляції текстом або вставку недостовірної інформації.

Формально, аномалії можуть бути виявлені шляхом визначення відстані між спостережуваними даними та звичайними чи очікуваними значеннями. Один із популярних методів виявлення аномалій - це використання методів зіставлення розподілів та порівняння ймовірностей входження слів чи фраз у текст зі стандартними розподілами.

У практиці виявлення аномалій в тексті може використовувати багато різних методів, включаючи машинне навчання, статистичні та лінгвістичні підходи.

Ймовірність грає важливу роль у виявленні інформаційних вкидів, де визначення ймовірностей входження певних слів, фраз, чи виразів у текст може вказати на ймовірність дезінформації. Статистичні методи, такі як аналіз розподілів та використання статистичних моделей, дозволяють виявляти аномалії в даних та вказувати на можливі інформаційні вклади.

Визначення ймовірностей грає важливу роль у виявленні інформаційних вкладів та дезінформації в текстових даних[16]. Використання ймовірностей допомагає визначити, наскільки незвичайними є певні аспекти тексту порівняно з очікуваними чи стандартними розподілами. В основі багатьох методів виявлення аномалій у тексті лежить порівняння логарифмічних ймовірностей.

Однією з популярних технік є використання логарифмічної ймовірності (log probability), яка визначає, наскільки ймовірним є входження слова чи фрази у текст порівняно зі стандартним розподілом слів. Ця ймовірність може бути розрахована за допомогою наступної формули:

$$P(x) = \log\left(\frac{N_x + 1}{N + V}\right)$$

де

- \(P(x)\) - логарифмічна ймовірність входження слова чи фрази \(x\) у текст.
- \(N_x\) - кількість разів, коли слово \(x\) зустрілося у тексті.
- \(N\) - загальна кількість слів у тексті.
- \(V\) - загальна кількість унікальних слів у тексті.

Ця формула використовується для обчислення логарифмічної ймовірності входження слова чи фрази у текст на основі статистики, отриманої з тексту. Чим менше значення логарифмічної ймовірності, тим менш ймовірно входження слова чи фрази у текст порівняно з очікуваним розподілом слів. Отже, низькі значення можуть свідчити про аномалії або підозрілість.

Під час аналізу тексту для виявлення інформаційних вкладів, дослідники можуть порівнювати логарифмічні ймовірності різних слів чи фраз, визначати, які з них мають найнижчі значення, і вважати їх підозрілими або аномальними. Порівнання логарифмічних ймовірностей дозволяє виділити слова чи фрази, які можуть свідчити про недостовірну інформацію або інформаційні вклади у тексті.

Однак важливо враховувати, що виявлення аномалій в тексті - це складний процес, який може вимагати додаткового аналізу та контекстуального розуміння тексту. Точність результатів може покращити за допомогою інших методів та алгоритмів машинного навчання.

Загалом, математичний опис процесу виявлення інформаційних вкладів допомагає створити раціональні та точні методи для захисту інформації в умовах інформаційної війни, і розвивається надзвичайно швидко відповідно до постійно змінних загроз та технологій.

2.2 Метод машинного навчання та штучний інтелект

Використання машинного навчання та штучного інтелекту є критично важливим для виявлення інформаційних вкладів під час інформаційної війни. Ці технології надають можливість автоматизувати процес аналізу текстових даних та виявлення аномалій в них.

Машинне навчання дозволяє створювати класифікатори, які можуть розпізнавати текст, що містить ознаки інформаційних вкладів, або визначати емоційний тон тексту, що може вказувати на спроби маніпуляції аудиторією. Використання глибокого навчання та нейромереж дозволяє створювати складні моделі для аналізу тексту, зокрема розпізнавати семантичні зв'язки та виявляти надмірну емоційну забарвленість тексту. Машинне навчання також може використовуватися для виявлення аномалій у тексті, включаючи незвичайні лексичні конструкції та синтаксичні аномалії.

Додатково, використання мовних моделей та машинного перекладу може допомогти виявити спроби маніпулювання перекладами для поширення дезінформації. Машинне навчання також використовується для кластеризації текстових документів, що дозволяє групувати схожі текстові матеріали і виявляти групи документів, що мають спільні теми або структури. Загалом, використання машинного навчання та штучного інтелекту робить процес виявлення інформаційних вкладів більш ефективним та точним, особливо в умовах інформаційної війни, де обсяг та складність текстових даних можуть бути значними. Важливо зазначити, що виявлення інформаційних вкладів вимагає поєднання різних методів та технологій, включаючи машинне навчання та ШІ, а також експертні знання і людську експертизу. Інтеграція цих підходів дозволяє підвищити надійність та точність виявлення інформаційних вкладів та дезінформації.

Більше того, розробка та навчання моделей машинного навчання вимагає постійного оновлення і адаптації до нових видів дезінформації та інформаційних вкладів. Всесвітні події та стратегії дезінформації постійно еволюціонують, і системи виявлення повинні бути гнучкими для адаптації до цих змін.

Загалом, поєднання машинного навчання, штучного інтелекту та людської експертизи допомагає створити комплексні системи для виявлення інформаційних вкладів та захисту від дезінформації в цифровому світі, де поширення неправдивої інформації стає все більш важливим та шкідливим явищем.

2.3 Метод графів та мереж

Метод графів та мереж є потужним інструментом для аналізу та виявлення інформаційних вкладів під час інформаційної війни. Цей підхід базується на ідеї представлення інформації у вигляді графа, де вузли представляють сутності, а ребра - взаємозв'язки між ними.

Аналіз соціальних мереж є однією з ключових сфер використання цього методу. В інформаційній війні, де соціальні мережі важливий канал поширення дезінформації, аналіз графової структури може виявити ненормальні взаємодії та активність, такі як автоматизовані боти, які поширюють неправдиву інформацію. Аналіз графів дозволяє виявити групи користувачів, які спільно сприяють поширенню дезінформації.

Поширення дезінформації на веб-сайтах і через посилання також може бути виявлене за допомогою графового аналізу. Аналіз графової структури веб-сайтів допомагає визначити, які сайти мають найбільший вплив та як вони пов'язані між собою через гіпертекстові посилання. Це може бути важливим для виявлення джерел дезінформації.

Мережі пошуку та рейтингу, такі як Google, можуть також використовувати графовий аналіз для виявлення та рейтингування джерел інформації. Це допомагає визначати авторитетні джерела та виявляти потенційно недостовірні джерела, що поширюють дезінформацію.

Аналіз текстових графів та семантичний аналіз тексту допомагають виявити аномалії та незвичайні зв'язки в тексті, що можуть свідчити про спроби маніпуляції інформацією.

Моніторинг медіа-мереж дозволяє виявляти дезінформацію та інформаційні вклади у новинах та інших медіа-ресурсах.

Усі ці аспекти методу графів та мереж спільно допомагають створити систему виявлення інформаційних вкладів та дезінформації, засновану на аналізі взаємозв'язків та структури інформації в цифровому просторі.

2.4 Використання статистичниих та аналітичних методів

Використання статистичних та аналітичних методів грає важливу роль у виявленні інформаційних вкладів під час інформаційної війни[17]. Ці методи дозволяють аналізувати великий обсяг даних та виявляти аномалії та закономірності, які можуть свідчити про дезінформацію.

Детальніше про використання статистичних та аналітичних методів:

– Аналіз текстової інформації. Статистичні методи дозволяють аналізувати текстову інформацію, включаючи визначення частоти вживання слів, фраз, тематичний аналіз текстів та виявлення незвичайних мовних конструкцій. Наприклад, використання певних слів або фраз може свідчити про спроби маніпуляції аудиторією.

– Аналіз соціальних мереж. Статистичні методи можуть бути використані для аналізу активності користувачів у соціальних мережах. Це включає в себе визначення часових шаблонів публікацій, кількості лайків, репостів та коментарів. Аномалії в цих показниках можуть вказувати на штучну активність або спроби поширення дезінформації.

– Аналіз медіа-мереж та новин. Статистичні методи можуть бути використані для аналізу медіа-мереж та новин. Це включає в себе визначення публікаційних патернів, виявлення однотипних заголовків та текстів у різних джерелах та визначення кількості публікацій на певну тему. Аномалії у цих показниках можуть свідчити про розповсюдження дезінформації.

– Аналіз рейтингів та репутації джерел. Статистичні методи можуть використовуватися для визначення рейтингів та репутації джерел інформації. Вони дозволяють виявити джерела, які мають низьку авторитетність та можуть поширювати недостовірну інформацію.

– Виявлення аномалій та відхилень. Статистичні методи дозволяють виявляти аномалії та відхилення від звичайних патернів в розповсюдженні інформації. Це

може включати в себе виявлення надмірної активності на певних ресурсах, зміни у фокусі тем та інші незвичайні явища.

– Прогностичні моделі. Статистичні методи дозволяють створювати прогностичні моделі для виявлення інформаційних вкидів. Вони можуть базуватися на історичних даних та аналізі трендів для визначення можливих спроб дезінформації.

Статистичні та аналітичні методи допомагають виявити аномалії та закономірності в інформаційному просторі, що може свідчити про дезінформацію та інформаційні вкиди. Ці методи сприяють створенню систем виявлення інформаційних вкидів, які допомагають захищати від дезінформації та підвищують інформаційну безпеку.

2.5 Вирішення проблеми доступу бота до інформації.

При створенні телеграм-бота можуть виникати деякі проблеми, особливо коли йдеться про взаємодію з певними обмеженнями, наприклад, коли бот стикається з обмеженнями та потребує певної імітації реального користувача або використання одноразових телефонних номерів. Нижче наведені деякі можливі проблеми та шляхи їх вирішення:

– Блокування чи обмеження використання бота. Телеграм може обмежити доступ бота через підозру у спамі чи ненормативній активності. Для вирішення даної проблеми необхідно здійснити використання бота з певною активністю, аналогічною активності реального користувача, може допомогти уникнути блокування. Це включає затримку між повідомленнями, реальне використання ботом та відповідне оброблення помилок.

– Необхідність підтвердження номера телефону. Для створення акаунта бота в телеграмі може знадобитися номер телефону. Вирішення виступає використання одноразових телефонних номерів (наприклад, через сервіси, які надають тимчасові

номери) дозволяє створити акаунт для бота без прив'язки реального особистого номера.

– Перешкоди у взаємодії з API. Деякі обмеження можуть бути в API телеграму, наприклад, ліміт на кількість запитів або доступ до певних функцій. Спроба імітувати поведінку реального користувача, затримки між запитами та розумне управління потоком запитів можуть допомогти уникнути блокування або лімітів API.

– Автоматична перевірка безпеки. В деяких випадках можуть застосовувати системи автоматичної перевірки безпеки, що може сприймати бота як потенційно шкідливого. Використання поведінки, схожої на реального користувача, з можливістю виконання різноманітних дій, а не створенням великої кількості подій одночасно, може зменшити ризик сприйняття бота як потенційно шкідливого.

Важливо збалансувати автоматизацію та імітацію реальної активності користувача, щоб уникнути проблем з блокуванням або обмеженнями. Імітація реального користувача та використання одноразових номерів телефону будуть використані під час програмної реалізації бота та допоможуть

2.6 Висновки з розділу

У даному розділі було розглянуто різні методи та підходи, що використовуються для виявлення інформаційних вкидів під час інформаційної війни. Відмічено, що ця проблема стає все більш актуальною в контексті сучасного інформаційного середовища, де поширення дезінформації та неправдивої інформації стає загрозою для суспільства та політичного процесу. Методи виявлення інформаційних вкидів розглядалися в контексті застосування штучного інтелекту, машинного навчання, аналітики тексту та графового аналізу. Важливим є розуміння, що жоден з цих методів не є універсальним, і їх успішність залежить від контексту та специфіки інформаційної війни.

Загальною тенденцією є поєднання різних методів та підходів для досягнення найкращих результатів. Важливо також враховувати постійну еволюцію стратегій дезінформації, що вимагає постійного вдосконалення систем виявлення.

Розділ надає базове уявлення про різноманітні методи та підходи до виявлення інформаційних вкладів, але варто пам'ятати, що виявлення інформаційних вкладів - це складна і багатоаспектна задача, яка вимагає поєднання технічних засобів, експертної експертизи та моніторингу інформаційного середовища.

3. РОЗРОБКА СТРУКТУРНОЇ СХЕМИ ЗАСОБУ ВИЯВЛЕННЯ ІНФОРМАЦІЙНИХ ВКИДІВ

3.1 Формалізація задачі

Система призначена для виявлення інформаційних вкидів. Дана система містить такі блоки:

- модуль взаємодії з користувачем, у вигляді телеграм боту, за допомогою бібліотеки TelegramAPI[18];
- модуль парсингу на основі gram-js;
- модуль аналізу відпарсеного вмісту;
- модуль факт-чекінгу;
- база даних для зберігання проаналізованого вмісту.

Узагальнена архітектура взаємодії модулів представлена нижче (рис. 3.1).

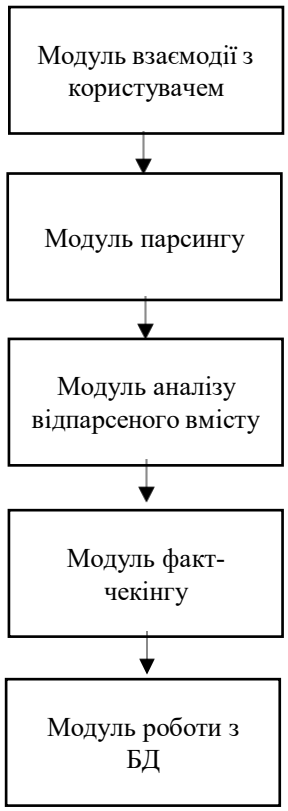

Рисунок 3.1 – Узагальнена архітектура системи

Потрібно розробити інтерфейс на основі телеграм-боту за допомогою

NodeTelegramAPI. Для зручності взаємодії додати кілька навігаційних кнопок.

Парсинг розробляється за допомогою gram-js. Після розробки модулю, парсинг буде імітувати реального користувача завдяки бібліотеці TelegramSession.

У модулі аналізу відпарсеного контунту буде відбуватись семантичний та аналіз методом графів, потенційного вкиду.

Модуль факт-чекінгу представлятиме собою останню перевірку імовірного вкиду в незалежних журналістів, на основі оцінки яких, буде прприйматись остаточний результат перевірки інформації і виноситись рішення чи це інформаційний вкид.

Результати перевірки записуються в базу даних, а користувачі можуть отримати результати.

3.2 Модуль взаємодії з користувачем

Користувацький інтерфейс розробляється на основі телеграм-боту. Вибір такого рішення пов’язаний з тим, що це є дуже зручно і ефективно. Крім того, за допомогою цього рішення вирішується питання кросплатформенності для цього засобу, так як Telegram має застосунку для різних операційних систем. Для зручності користувачів потрібно додати навігаційні кнопки і відповіді боту для кожного повідомлення користувача.

Після кожної взаємодії бот не потребує перезапуску так як має функції асинхронності.

Для роботи модулю з Telegram використано бібліотеку node telegram-bot-api.

Блок-схема роботи модуля представлена на рис. 3.2.

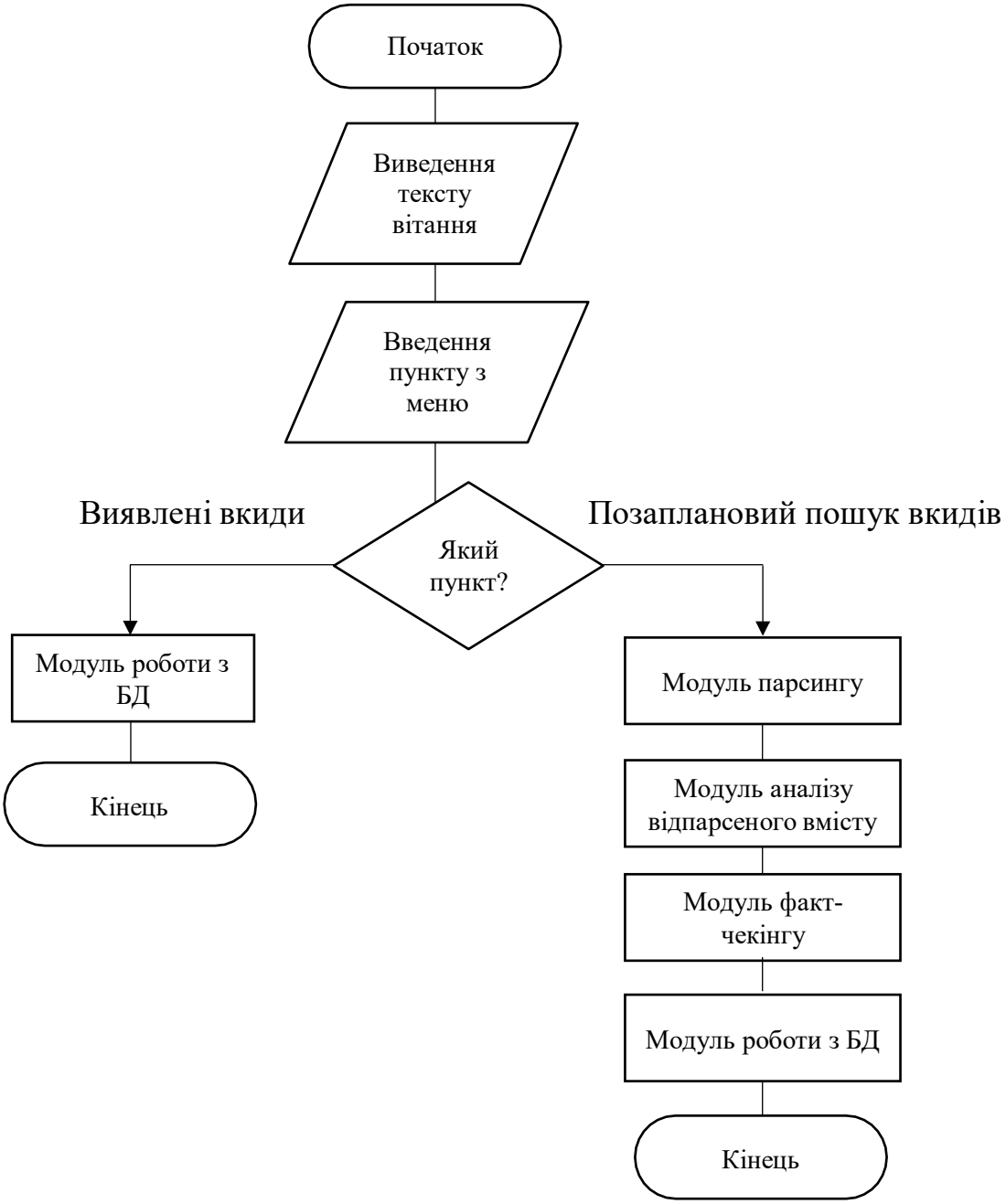

Рисунок 3.2 – Схема роботи модулю взаємодії з користувачем

Цей модуль пов’язує між собою інші модулі застосунку для повноцінної роботи.

3.3 Модуль парсингу

Модуль парсингу в першу чергу буде парсити новинні телеграм канали, та офіційні телеграм канали української влади. Це пов’язано з тим, що з початком

повномаштабного вторгнення, саме телеграм канали найбільше використовуються для поширення новин, є одним з самих швидких методів поширення інформації та мають найбільшу довіру в користувачів[19]. Проте, крім того, для порівняння та статистики також будуть парситись новинні ресурси та агрегатори новин.

Для роботи модулю парсингу створено нового користувача Telegram. Тому щоTelegramBotAPI не дозволяє парсити публічні телеграм-канали. З цією метою, потрібно розробити процеси, які створюють файл сесії і імітують реального користувача. Це питання вирішується за допомогою TelegramSession

Для подальшого парсингу модуль отримує від боту переслане повідомлення потрібного поста. Після цього модуль читає це повідомлення і переводить його в json. Далі модуль отримує текст даного повідомлення і надсидає його в модуль аналізу.

Схема роботи модулю парсингу зображена на рисунку 3.3.

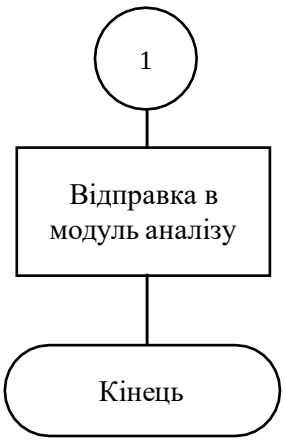

Рисунок 3.3 – Схема роботи модуля парсингу

Схема представляє роботи модуля парсингу, а саме частини де відбувається парсинг інформації з новинних телеграм каналів.

3.4 Модуль запису в базу даних

Для реалізації всіх функції засобу потрібно мати доступ до баз даних, для вирішення даної задачі обрано mySQL. Через переваги у простоті роботи та швидкості підключення і запису в базу даних.

Щоб не перевіряти один інформаційний вкид багато разів, бот після перевірки буде інформація. і значення перевірки в базу даних. Перед наступними перевірками бот перевірятиме чи є в базі даних відомості про даний інформаційний вкид і якщо вони є, не проводитиме перевірку, а виведе користувачеві повідомлення про те, що новина вже перевірялася і результат перевірки.

Для реалізації цієї задумки потрібно написати модуль запису результатів перевірки в базу даних, додати токен бази даних в конфігураційний файл, а також відкрити доступ до IP-адреси в налаштуваннях самої бази даних.

Після перевірки модуль отримуватиме новину і значення лічильника і вноситиме їх в базу даних з заміною значення лічильника на результат перевірки.

Загальна схема роботи модулю запису в базу даних представлена на рисунку 3.4.

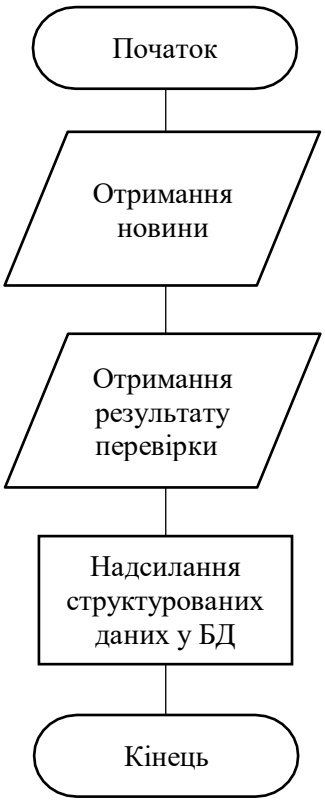

Рисунок 3.4 – Схема роботи модулю запису в базу даних

Робота з базою даних допоможе забезпечити запис виявлених інформаційних вкидів, результати перевірок, а також дасть змогу зменшити кількість перевірок одного вкиду велику кількість разів.

3.5 Модуль аналізу відпарсеного вмісту

Модуль аналізу базується на використанні семантичного аналізу та методу графів для ефективного виявлення інформаційних вкидів.

Архітектура модулю спрямована на поєднання семантичного аналізу тексту та використання графів для ефективного виявлення інформаційних вкидів. Це

дозволить системі точно визначати та виділяти ключові аспекти тексту, що вказують на можливі відхилення від звичайних моделей або інформаційних шаблонів.

Першим етапом обробки є предпроцесинг тексту для очищення від зайвих символів, розбиття на слова та фрази, зведення слів до базової форми та зведення слів до базової форми з урахуванням контексту.

Використання методів семантичного аналізу для розуміння сутностей, зв'язків та значень у тексті. Це включає в себе використання векторних моделей, наприклад Word2Vec, для перетворення слів у числовий векторний простір та визначення семантичних відстаней між ними[20].

На основі семантичного аналізу відбувається побудова графів, де вузли представляють слова чи фрази, а ребра – зв’язки та відносини між ними. Граф допомагає візуалізувати та аналізувати структуру тексту, виявляти ключові паттерни та зв’язки між елементами.

Застосування алгоритмів аналізу графів, таких як алгоритм виявлення вузлів-центрів та алгоритм пошуку аномалій, в кінцевому результаті допоможуть знайти аномальні зв’язки, несподівані взаємозв’язки чи інших ознак, які можуть вказувати на інформаційні вкиди[21].

Результат перевірки даного модулю записується в базу даних або надсилається користувачеві, проте сам текст новини направляється в модуль факт-чекінгу на остаточну перевірку. Результати перевірки на даному етапі можуть бути змінені в базі даних на основі перевірки факт-чекінгу.

Схема роботи модулю аналізу зображена на рисунку 3.5.

Рисунок 3.5 – Схема роботи модулю аналізу

Результат перевірки в даному модулі проводиться на основі семантичного аналізу та методу графів, а тому може бути не достатньо повним.

Для впевненості в абсолютно точній роботі системи, відпарсений текст новини відправляється також в модуль факт-чекінгу.

3.6 Модуль факт-чекінгу

Семантичний аналіз, як і будь-який інший аналітичний процес, має свої обмеження та потенційні помилки, тому незалежні журналістські факт-чекінгові ресурси грають важливу роль у виявленні інформаційних вкидів.

Оскільки вони можуть допомогти в перевірці істинності чи точності інформації, виявленні потенційних помилок чи неточностей у тексті, а також уточненні контексту та перевірці джерела інформації. Вони вносять додатковий шар перевірки, що може бути важливим, особливо у випадках, коли автоматизовані методи можуть не бути достатньо ефективними у виявленні неточностей або маніпуляційної інформації.

З метою реалізації додаткової перевірки можливого інформаційного вкиду чи маніпуляції модуль факт-чекінгу отримує відпарсений текст новини з попередніх модулів.

Далі з отриманого тексту формується повідомлення на основі імітації реального користувача і надсилається в телеграм-бот «Перевірка». Бот є проектом «Gwara Media» - незалежних українських журналістів та факт-чекерів, які на добровільних засадах перевіряють отриману від користувачів інформацію.

Це зроблено з метою додаткової перевірки новини, тому що семантичний аналіз може бути неефективним в одному конкретному контексті, інформація може бути неоднозначною або потрібно перевірити саме джерело інформації.

На основі результату перевірки факт-чекерами, формується остаточний звіт про те, чи є новина інформаційним вкидом, якщо результат факт-чекінгу відрізняється від результату перевірки семантичним аналізом та методом графів, значення перевірки змінюється і в базі даних.

Схема роботи модулю факт-чекінгу представлена на рисунку 3.6.

Рисунок 3.6 – Схема роботи модулю факт-чекінгу

Крім того, для можливого звернення до проаналізованих вкидів, інформація в базі даних буде зберігатися тривалий період часу і може бути передана органам влади за потреби.

Отже розроблені модулі та алгоритми засобу для виявлення інформаційних вкидів, що дозволить розробити програмний засіб.

4 РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ

4.1 Формування вимог до програмного засобу

Метою є розробка системи виявлення інформаційних вкидів. Розроблений програмний засіб повинен бути у вигляді боту в додатку Telegram.

Додаток повинен відповідати наступним вимогам:

- Наявність графічного інтерфейсу – на основі телеграм-боту;
- Можливість перевіряти інформацію як з телеграм-каналів, так і з Інших новинних агрегаторів;
- Наявність роботи з файлами, для зчитування вхідної інформації;
- Наявність роботи з базами даних, для зчитування та запису даних про перевірену інформацію;
- Можливість перевірки інформації за вимогою користувача, а також систематичний аналіз новин з обраним інтервалом.
- Розробка супроводжувальної документації – детальний опис додатку, процесів та інструкції з використання;
- Застосування об’єктно-орієнтованого програмування.

Телеграм-бот може посилатись спеціалізовані боти чи сервіси для перевірки інформації на достовірність.

4.2 Обґрунтування засобів для реалізації.

Для реалізації програми необхідно використати одну зі сучасних об’єктно-орієнтованих мов програмування, яка працює на сервері і може відповідати на запити через HTTPS. Найпопулярнішими мовами програмування які використовуються для створення телеграм-ботів є:

- C#;

- Javascript (Node.js);
- Python.

C# - проста, сучасна мова програмування, забезпечує принципи ООП. На синтаксис мови C# має найбільший вплив C++. Дана мова є ввихідним кодом під .NET, який працює незалежно від Microsoft. C# має ряд переваг таких як:

- швидкість, порівняно з іншими мовами високого рівня;
- велика підтримка – завдяки розробникам і спільноті користувачів. Найбільша спільнота експертів знаходиться на StackOverflow, не в останню чергу через те що сайт побудований на C#;
- крос-платформа - є мовою на різних платформах. Дає змогу створювати програми .NET, які можна розгорнути не тільки на Windows, а й на Linux, Mac чи у хмарі та контейнерах;
- повністю об'єктно-орієнтований, що є рідкісною характеристикою. Багато найпоширеніших мов до певної міри включають об'єктну орієнтацію, але дуже мало хто досягнув такої величини, не втрачаючи прихильності людей. Така концепція має багато різних переваг, таких як гнучкість та ефективність;
- безпечний та ефективний - не дозволяє перетворювати типи, які призводять до точних даних або чогось іншого. Що дозволяє розробляти більш безпечний код;
- універсальний – ця мова програмування була розроблена всебічно і використовується для великої кількості різних потреб: для створення сучасних програмних додатків, використовується для розробки клієнтських програм Windows, мобільних програми, сервісів, бібліотек, веб-програм та відеоігр;
- швидко розвиваючий - 8.0 остання версія C#. Якщо поглянути на історію, то ця мова розвивається швидше, ніж будь-які інші. Завдяки Microsoft та потужній підтримці спільноти. Спочатку вона була розроблена для написання клієнтських програм Windows, але сьогодні можна робити майже все, що завгодно.

Javascript – мова сценаріїв, одна з основних мов для створення веб-сайтів.

Зазвичай вона є мовою на стороні клієнта, але існує платформа Node.js, яка дозволяє виконувати код на сервері.

Зі свого трохи крихкого початку JavaScript став найпопулярнішою мовою програмування у світі. Відповідно до звіту GitHub за 2018 рік, сховищ коду на даній мові є більше, ніж будь-якої іншої, - і ця кількість постійно зростає.

JavaScript - це «безпечна» мова програмування. Вона не має низькорівневого доступ до пам'яті або центрального процесора, оскільки спочатку була реалізована тільки для браузерів, які цього не потребували. Можливості мови сильно залежать від середовища, в якому вона працює

Плюсами розробки на цій мові програмування є:

– статичний тип змінних - після оголошення вона не змінює свого типу і може приймати лише певні значення. Компілятор попереджає розробників про помилки, пов'язані з типом, тому вони не мають можливості потрапити на фазу виробництва. Це призводить до меншої кількості помилок та кращої продуктивності під час виконання;

– знаходження ранніх помилок - виявлено, що JavaScript виявляє 15 відсотків поширених помилок на етапі компіляції. Ця сума все ще є досить значною, щоб заощадити час і дозволити зосередитись на виправленні помилок в логіці;

– передбачуваність - у JavaScript все залишається так, як визначено спочатку. Якщо змінну оголошено як число, вона завжди буде цим типом і не перетвориться на щось інше. Це підвищує ймовірність функцій, що працюють так, як передбачалося спочатку;

– читабельність - завдяки строгим типам, які роблять код більш самовиразним, можна побачити задум розробників, які спочатку написали код. Це особливо важливо для команд з великою кількістю розробників, що працюють над одним проектом. Код, який говорить сам за себе, може компенсувати відсутність прямого спілкування між членами команди.

– багата підтримка IDE - інформація про типи робить редактори та

інтегровані середовища розробки набагато кориснішими. Вони можуть запропонувати такі функції, як навігація кодом та автозаповнення, надаючи точні пропозиції. Ви також отримуєте зворотній зв'язок під час набору тексту: редактор позначає помилки, включаючи типи, щойно вони виникають. Все це допомагає написати код, який можна підтримувати, і призводить до значного підвищення продуктивності;

– швидкий рефакторинг - JavaScript робить цей процес менш болючим. Оскільки середовища розробки знають багато про даний код. Крім того, багато помилок виявляються автоматично. Це спрощує та прискорює рефакторинг, що особливо корисно, коли є велика частина кодової бази;

– велика підтримка розробників - кількість розробників JavaScript сягає 11,4 мільйона - майже вдвічі більше, ніж у такий популярних мов Python або Java.

Python - це популярна мова програмування, яка має загальне призначення, що може бути використане для найрізноманітніших програм. Вона включає динамічне введення тексту та прив'язування, високорівневі структури даних та багато інших функцій, які роблять його настільки ж корисним для складної розробки додатків, як і для створення сценаріїв. Крім цього, може бути розширений для здійснення системних викликів майже до всіх операційних систем і для запуску коду, написаного на C ++. Тому Python - це універсальна мова, що зустрічається в безлічі різних застосувань.

Основними перевагами Python є:

– простий і зрозумілий синтаксис;

– відсутність дужок;

– автоматичний розподіл пам'яті;

– динамічний набір тексту;

– велика підтримка в інтернеті;

– багато підтримуваних бібліотек.

Програми чудово працюють для машинного навчання. Але Python поганий вибір для мобільних додатків.

Можливі мінуси Python:

– повільний і може стати громіздким для великих і складних програм;

– мова високого рівня, яка не підходить для написання програмних програм;

– для деяких завдань неявне виділення пам’яті може бути недоліком.

Для реалізації програмного засобу обрано мову програмування Javascript, оскільки ця мова має такі переваги:

- незалежність від платформи, на якій виконується програма;
- мова є повністю об’єктно-орієнтованою;
- повна інкапсуляція;
- відсутність глобальних функцій;
- зручні бібліотеки для виконання специфічних завдань бота.

Порівняння мов програмування і платформ між собою показало, що Javascript та платформа Node.js мають сучасні переваги перед своїми конкурентами в рамках розробки даного програмного застосунку.

Крім того, Javascript ідеально підходить для роботи з TelegramBotAPI та TelegramSession – ключовими модулями для роботи телеграм-боту.

Серед таких середовищ, як Atom, Webstorm, Visual Studio Code, в межах робити обрано Visual Studio Code. Він є безкоштовним, має інтеграцію з Git, і режим налагодження коду. Підтримує мову програмування Javascript, автоматично виділяє синтаксичні конструкції, має підказки та довідку.

Реалізація графічного інтерфейсу застосунку відбулась за допомогою наявного в Telegram графічного інтерфейсу бота.

4.3 Реєстрація та попереднє налаштування бота

Для створення телеграм-боту спершу потрібно зареєструвати його в Telegram, щоб отримати доступ до його API-токену.

Головним інструментом створення та реєстрації бота в Telegram є

@BotFather. Реєстрацію боту зображено на рисунку 4.1.

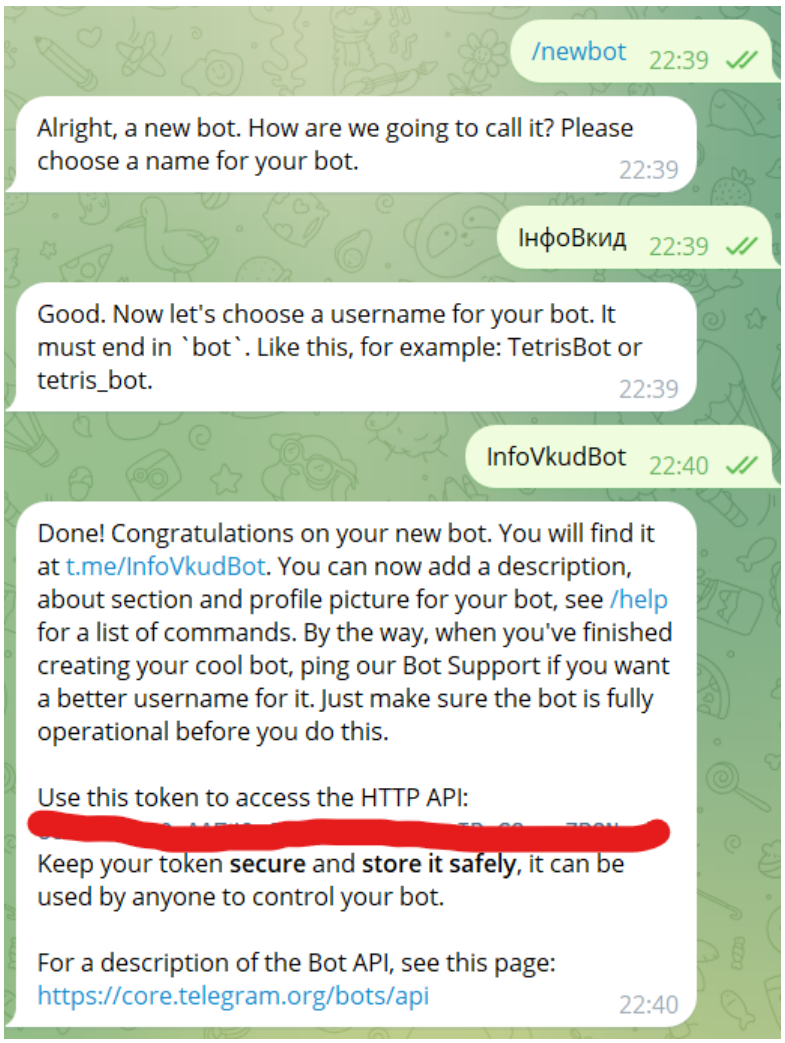

Рисунок 4.1 – Скріншот реєстрації бота

Після запуску, боту дано назву та ім’я, у відповідь отримано API-токен, який буде використовуватися для подальшої реалізації застосунку.

Крім того, через даний інструмент можна додати список команд для майбутнього модулю взаємодії з користувачем (рис. 4.2).

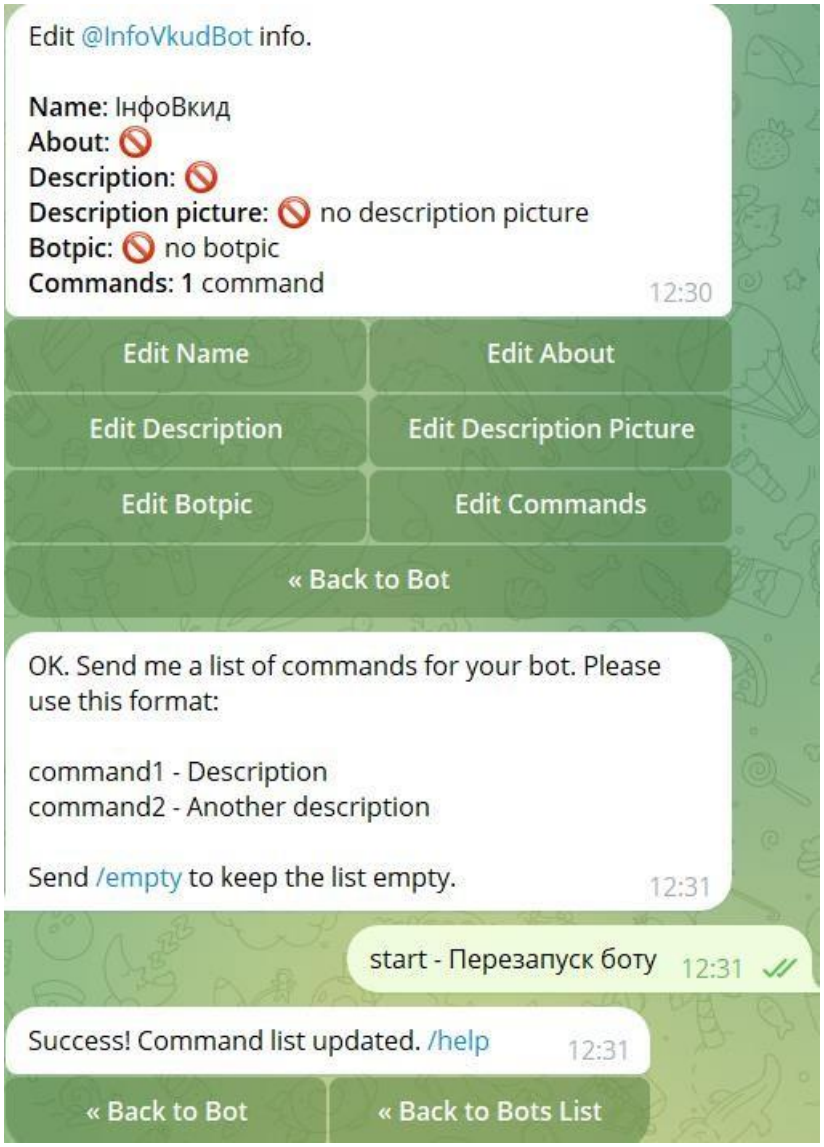

Рисунок 4.2 – Скріншот при додаванні команд

Після виконання реєстрації і попереднього налаштування боту можна переходити в редактор коду для подальшої програмної реалізації.

4.4 Налаштування роботи з базою даних.

Для роботи засобу і усунення зайвих перевірок однієї новини на наявність інформаційного вкиду по багато разів, застосунок повинен працювати з базою даних, куди записувати виявлені інформаційні вкиди і читати її перед перевіркою, якщо результат вже є – виводити його.

Для програмного застосунку обрано систему керування базами даних MySql. MySQL відома своєю високою продуктивністю та швидкодією[22]. Вона працює ефективно навіть з великими обсягами даних, що робить її ідеальним вибором для додатків з вимогами до швидкодії та відповідності.

Можливості MySQL:

- висока швидкість;
- підтримка реплікації;
- гнучкість використання;
- використання індексів;
- підтримка транзакцій та замків;
- використання сховищ та процедур.

Для роботи з телеграм-ботом, спочатку потрібно створити базу даних, це можна зробити через веб-інтерфейс phpMyAdmin[23] або через програму MySQL Workbench.

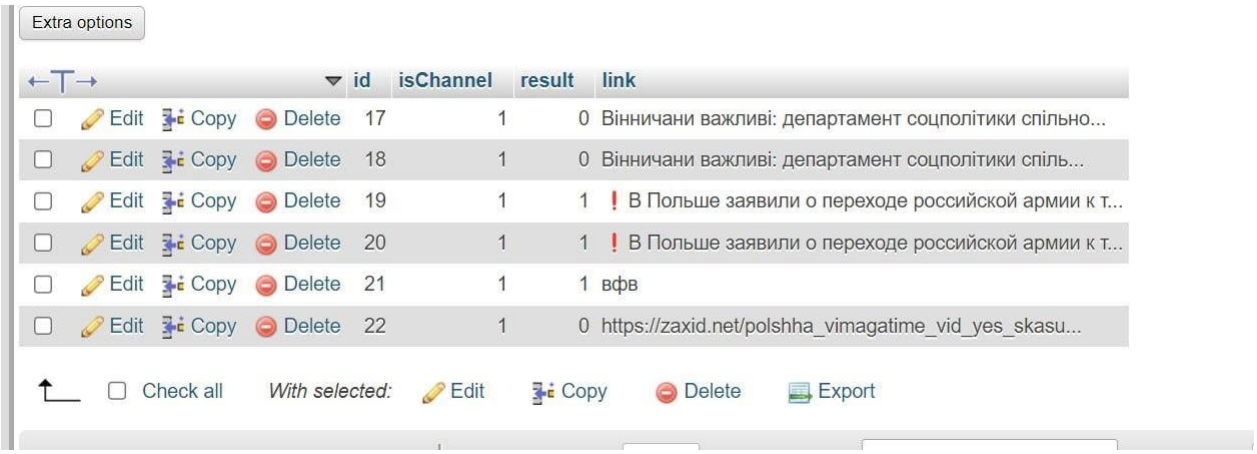

Рисунок 4.3 – База даних для роботи системи

На рисунку вище зображено готову до використання базу даних MySQL, далі збережене посилання для підключення і пароль потрібно вписати в файлі config.js, де вже знаходяться токени для налаштування бота отримані з Telegram.

На рисунку 4.4 зображено вміст файлу config.js, в якому містяться потрібні для роботи засобу токени.

```
require('dotenv').config()

const databaseConfig = {
  host: process.env.DATABASE_HOST,
  user: process.env.DATABASE_USER,
  database: process.env.DATABASE_NAME,
  password: process.env.DATABASE_PASSWORD,
};

module.exports = databaseConfig;
```

Рисунок 4.4 – Вміст файлу config.js

Для створення з'єднання і можливості читання/запису в базу даних в застосунку використовується об'єктно орієнтована бібліотека dotenv [24]. Під час запуску телеграм-боту через термінал VS Code, вкінці додано вивід в термінал "DB has connected" що означає, що все функціонує правильно і база даних підключена і готова до запису чи читання.

4.5 Реалізація модулю парсингу

Парсинг телеграм-боту виконується через метод імітації реального користувача за допомогою TelegramSession тому, що TelegramBotAPI забороняє робити це напряму через бота.

Для початку необхідно створити нового користувача Telegram, для захисту персонального аккаунту від блокування, але можна використовувати і власний.

Потрібно пройти автентифікацію облікового запису Telegram через TelegramSession і зберегти отриманий в результаті автентифікації токен, для файлу config.js.

В бот потрібно переслати бажане повідомлення з телеграм каналу з новиною, яка потребує перевірки або скинути посилання на новину в залежності від режиму роботи бота.

Парсинг тексту новинного ресурсу за адресою зображено на рисунку 4.5.

```
const axios = require('axios');
const cheerio = require('cheerio');

const url = user.send.url;

axios.get(url)
  .then(response => {
    const html = response.data;
    const $ = cheerio.load(html);

    // Отримання тексту
    const text = $('body').text();
    console.log('Текст сторінки:', text);
```

Рисунок 4.5 – Парсинг тексту за адресою новини

У випадку перевірки новини з телеграм каналу, користувач персилає в бота конкретне повідомлення з новиною або пише його вручну. Текст парситься і відправляється на аналіз.

4.6 Реалізація модулю аналізу

Для реалізації модулю аналізу новин на наявність інформаційних вкидів використовуються моделі обробки живої мови (nlp) [25].

Клас tokenizator приймає на вхід текст і розбиває на токени. Наприклад, кожне слово, послідовність прогалин, пунктуація вважаються окремими токенами (рис. 4.6).

```
tokenizator(text, options) {

  this.words = Array.isArray(text) ? text : this.RiTa.tokenize(text);
  this.ignoreCase = options && options.ignoreCase || false;
  this.ignoreStopWords = options && options.ignoreStopWords || false;
  this.ignorePunctuation = options && options.ignorePunctuation || false;
  this.wordsToIgnore = options && options.wordsToIgnore || [];

  this._buildModel();

  let result = {};
  for (let name in this.model) {
    result[name] = this.model[name].indexes.length;
  }
  return result;
}
```

Рисунок 4.6 – Токенізація тексту

Жодного додаткового аналізу над токенами не проводиться.

При цьому розбивка проводиться відносно «розумним» способом, який дозволяє виділяти окремі токени посилання, хештеги та адреси електронної пошти.

Морфологічний аналізатор приймає на вхід єдине слово (наприклад, отримане після токенізації) та повертає масив можливих варіантів його розбору: частину мови, відмінок, рід тощо. Варіанти сортуються за спаданням «правдоподібності», тому передбачається, що перший варіант має бути найближчим до істини.

Після цього відбувається виявлення слів або фраз, які несуть нову інформацію або змінюють хід розповіді. Це можуть бути наголошені факти, висловлювання зі значущою вагою, які виділяються у контексті тексту (рис. 4.7).

```
let rules = type === SING ? SING_RULES : PLUR_RULES;
for (let i = 0; i < rules.length; i++) {
  let rule = rules[i];
  if (rule.applies(check)) {
    dbug && console.log(word + ' (' + (type === SING ? 'singularize' : 'pluralize')
      + ') hit ' + (type === SING ? 'singular' : 'plural')
      + (i < rules.length - 1 ? ' rule #' + i : ' DEFAULT rule'), rule);
    return rules[i].fire(word);
  }
}
```

Рисунок 4.7 – Використання правила для визначення значущості висловлювання

Врахування способу, яким вживаються ключові слова або фрази в тексті допомагає визначити, чи є ці слова дійсно важливими в контексті, чи ж вони випадково потрапили у текст.

Коли ми говоримо про аналіз тексту за допомогою графів, можна уявити собі карту - карту, на якій слова та поняття зв'язані лініями. Ці лінії показують, як вони пов'язані між собою. Коли ми розглядаємо текст у вигляді графів, ми можемо побачити, які слова або ідеї взаємодіють найбільше.

Графічне представлення тексту у вигляді графів також допомагає визначити

важливі шляхи або зв'язки між словами. Наприклад, якщо ми малюємо графік зі словами "розвиток", "технологія" та "майбутнє", ми можемо побачити, як ці концепції пов'язані між собою і як вони створюють ключову тему в тексті.

Такий підхід допомагає не тільки візуалізувати текст, але і краще розуміти його структуру та зв'язки між окремими елементами.

4.7 Реалізація модулю факт-чекінгу

Для вирішення проблеми додаткової перевірки інформації на виявлення в ній інформаційних вкидів бот використовує парсинг, а також готові засоби перевірки у вигляді журналістів-фактчекерів України, а саме телеграм-бот «Перевірка | Perevirka».

В пропонованому готовому телеграм-боті проводиться перевірка по базі даних, яку постійно поповнюють журналісти, а разі відсутності інформації по шуканій новині проходить перевірка інформації вручну. Це забезпечує високий рівень правдивості отриманої відповіді.

Для реалізації додаткової перевірки бот парсить текст відправлений користувачем та відправляє його в телеграм-бот «Перевірка | Perevirka», відправляє користувачеві повідомлення про те що, інформацію взято в обробіток і очікує відповіді.

Після перевірки по базі даних журналістів або вручну телеграм-бот «Перевірка | Perevirka» відправляє відповідь, яку бот миттєво парсить і проводить аналіз відповіді.

В разі, якщо результат факт-чекінгу відрізняється від результату перевірки в модулі аналізу бот оновлює результат в базі даних та відправляє користувачеві.

4.8 Тестування програмного застосунку по виявленню інформаційних вкидів

З метою перевірки на правильність роботи телеграм бота та виявлення інформаційних вкидів було вирішено перевірити кілька новин з телеграм-каналів та новинних агрегаторів.

Першою буде будь-яка новина з телеграм каналу Вінницької міської ради, приклад зображено на рисунку 4.8.

Рисунок 4.8 – Приклад новини для тестування

Проводиться запуск бота через VS Code і вибір параметру «перевірка каналу на шкідливий вміст». Потім потрібно переслати повідомлення з каналу, який потрібно перевірити, в даному випадку «Вінницька міська рада» і почати перевірку.

Результат перевірки телеграм-каналу «Вінницька міська рада» в рамках

тестування зображено на рисунку 4.9.

Рисунок 4.9 – Результат тестування бота

Переслана в бота новина, пройшла перевірку правильно, тому що не містить ніяких ознак інформаційного вкиду, як і визначив бот.

Інша новина відібрана для тестування повинна бути інформаційним вкидом.

З цією метою була взята на тестування новина з телеграм-каналу російського пропагандиста.

Даний телеграм канал регулярно постить інформаційні вкиди, фейки, контент для розпалювання міжнаціональної ворожнечі та рашистські лозунги.

Скріншот вмісту телеграм-каналу «соловйов» зображено на рисунку 4.10.

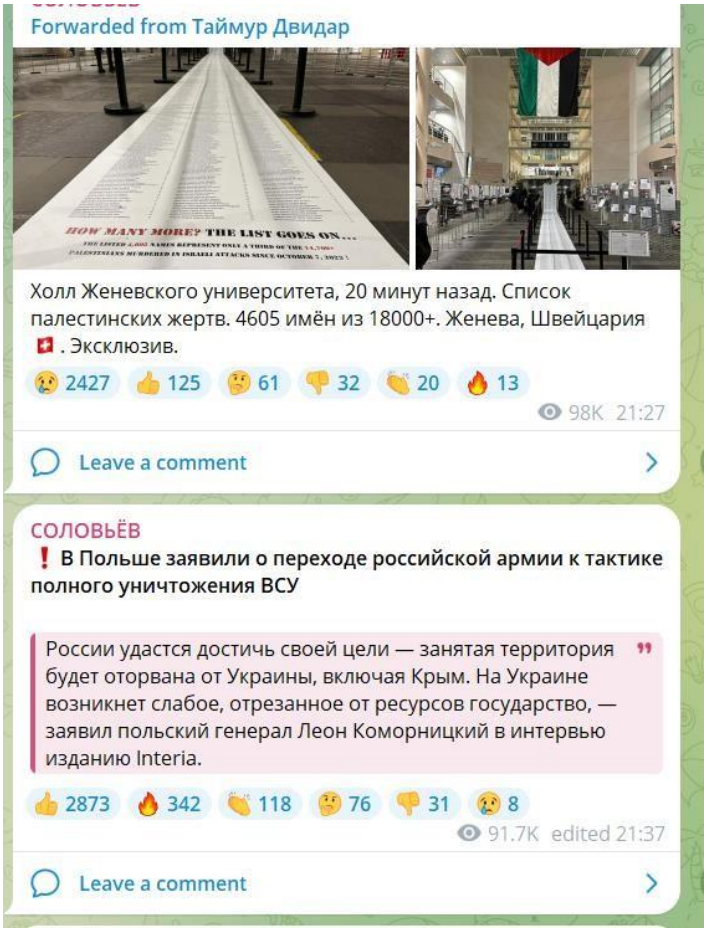

Рисунок 4.10 – Вміст телеграм-каналу «соловйов»

Для тестування коректної роботи бота по виявленню інформаційних, було вирішено, переслати йому на аналіз другу новину, оскільки неозброєним оком видно, що новина містить маніпуляцію, може бути повністю фейковою і використовується для погіршення стану українського суспільства, тобто скоріше за все є інформаційним вкидом.

Результат перевірки даної новини зображено на рисунку 4.11.

Бот визначив дану новину як інформаційний вкид, тому користувач отримав дане повідомлення, а також було здійснено запис в базу даних. Однак, після цього новина відправлена на додаткову обробку в модуль факт-чекінгу. По результатам роботи якого, запис в базі даних, а також результат, що отримав користувач можуть бути змінені.

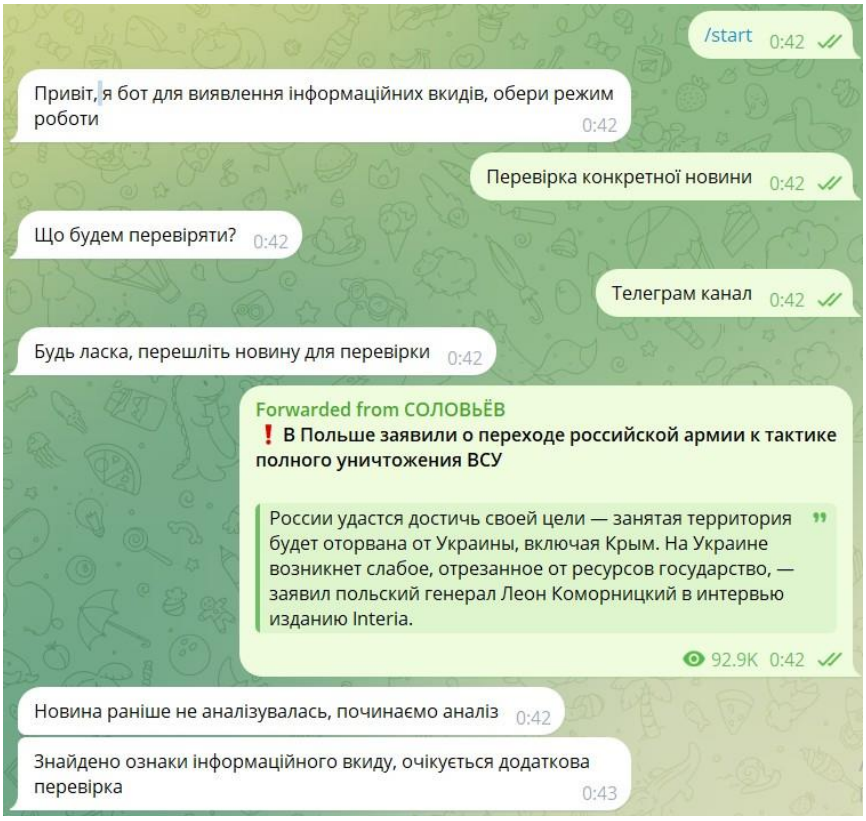

Рисунок 4.11 – Результат перевірки новини

По отриманню результату факт-чекінгу бот надіслав результат, що зображено на рисунку 4.12.

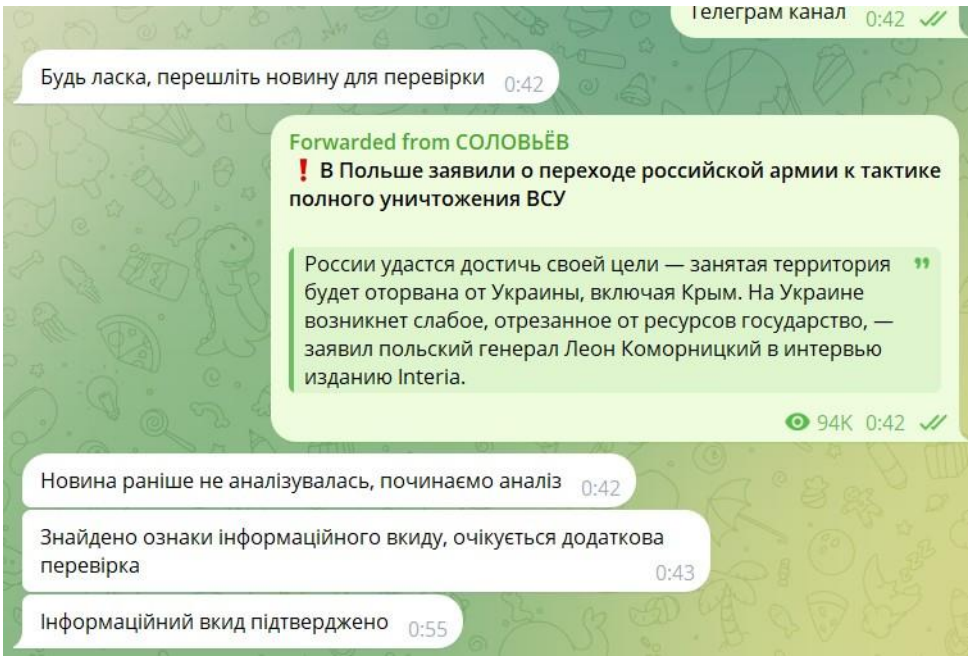

Рисунок 4.12 – Результат факт-чекінга

Отже, бот правильно виявив інформаційний вкид, і отримав підтвердження від факт-чекерів.

З метою тестування виявлення інформаційних сайтах на новинних агрегаторах та ресурсах було вирішено перевірити випадкову новину на ресурсі zaxid.net (рис. 4.13).

Рисунок 4.13 – Новина для тестування з ресурсу zaxid.net

Тестування перевірки на правильність виявлення інформаційних вкидів на сайтах відбувається так само, але такий поділ необхідний для того, щоб телеграм канали формували окремий список в базі даних, для функціонування функції моніторингу.

Результат перевірки на наявність вкидів зображено на рис. 4.14.

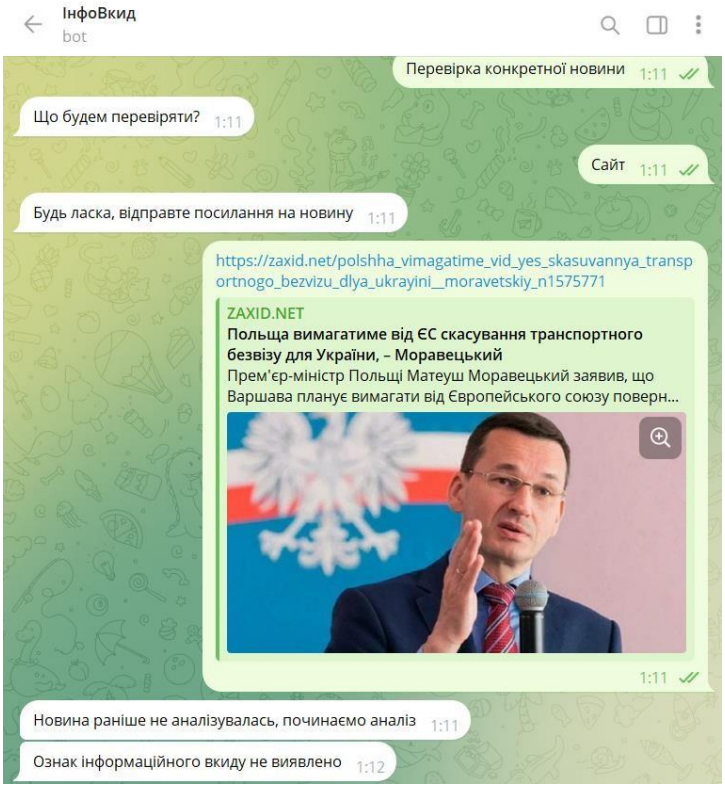

Рисунок 4.14 – Результат перевірки новини

Функція моніторингу являє собою перевірки останніх новин з телеграм каналів, новини з яких раніше проходили перевірку в ботові.

Результат тестування функції моніторингу телеграм каналів представлено на рисунку 4.15.

Рисунок 4.15 – Результат тестування функції моніторингу

Зважаючи на кількість телеграм каналів в базі даних, яка буде збільшуватися разом зі збільшенням користування ботом, для забезпечення швидкодії моніторингу каналів можливе зменшення кількості постів, що проходять перевірку в рамках моніторингу.

Більше детальна статистика тестування системи на реальних прикладах представлена в таблиці 4.1.

Таблиця 4.1 – Статистика тестування системи

Перевірені ресурси :		Визначення системи		Коректність роботи
		Виявлено інформаційний вкид	Не виявлено інформаційного вкиду	
Всього:	50			
З інформаційними вкидами	15	13	2	86,66%
Без інформаційних вкидів	35	1	34	97,15

Для проведення масштабного тестування роботи системи було перевірено 50 веб-ресурсів та телаграм-каналів.

Система показала гарні результати з правильності визначення інформаційних вкидів та порівняно низькі показники хибних спрацювань.

Загальний процент коректності роботи системи склав 91,9%, що вважається чудовим результатом для систем такого рівня.

Отже, розроблено програмний засіб, а саме телеграм-бот для виявлення інформаційних вкидів. За результатами тестування бот показав відносно невеликі затрати по часу та високий рівень виявлення інформаційних вкидів.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Система виявлення інформаційних викидів під час інформаційної війни» є покращення кібербезпеки суспільства шляхом створення системи виявлення інформаційних вкидів у кіберпросторі.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 4.1 [26].

Таблиця 5.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
	Технічні та споживчі властивості продукту значно гірші, ніж в	Технічні та споживчі влас-тивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в	Технічні та споживчі властивості продукту значно кращі, ніж в

Продовження табл. 5.1

	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
	Активна конкуренція великих компаній на	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
0	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
1	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років

Продовження табл. 5.1

2	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь- які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці. Для оцінки науково-технічного рівня і комерційного потенціалу розробки експертами було запрошено трьох незалежних експертів: Олеся Петрівна Войтович к.т.н. доцент кафедри захисту інформації Вінницького національного технічного університету, Олексій Палій Middle Software Engineer ТОВ «СМІСС», Дмитро Поворозник спеціаліст з налаштування інформаційних систем захисту ТОВ «АТК»

Таблиця 5.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	Олеся Петрівна Войтович К.т.н. доц.каф. ЗІ	Олексій Палій Middle Software Engineer ТОВ «СМІСС»	Дмитро Поворозник спеціаліст з налаштування інформаційних систем захисту ТОВ «АТК»
	Бали, виставлені експертами:		
1. Технічна здійсненність концепції	4	3	4
2. Ринкові переваги (наявність аналогів)	3	4	4
3. Ринкові переваги (ціна продукту)	3	3	3
4. Ринкові переваги (технічні властивості)	2	3	3
5. Ринкові переваги (експлуатаційні витрати)	4	3	4
6. Ринкові перспективи (розмір ринку)	3	3	3

Продовження табл. 5.2

7. Ринкові перспективи (конкуренція)	1	2	2
8. Практична здійсненність (наявність фахівців)	4	3	4
9. Практична здійсненність (наявність фінансів)	3	4	3
10. Практична здійсненність (необхідність нових матеріалів)	4	4	4
11. Практична здійсненність (термін реалізації)	4	4	4
12. Практична здійсненність (розробка документів)	4	4	4
Сума балів	$СБ_1$=39	$СБ_2$=40	$СБ_3$=42
Середньоарифметична сума балів $СБ_c$	$\overline{СБ} = \frac{\sum_1^3 СБ_i}{3} = \frac{39+40+42}{3} = 40.3$		

За результатами розрахунків, наведених в таблиці 4.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 4.3 [26].

Таблиця 5.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ , розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41…48	Високий
31…40	Вище середнього
21…30	Середній
11…20	Нижче середнього
0…10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою "Система виявлення інформаційних викидів під час інформаційної війни" становить 40 балів, що, відповідно до таблиці 4.3 рівень комерційного потенціалу розробки вище середнього, що свідчить про комерційну важливість проведення даних досліджень.

Магістерська кваліфікаційна робота "Система виявлення інформаційних викидів під час інформаційної війни" відноситься до науково-технічних робіт, які

орієнтовані на виведення на ринок, тобто при цьому відбувається комерціалізація науково-технічної розробки. Цей напрямок є для нас пріоритетним, оскільки результатами розробки можуть користуватися не тільки самі розробники, а й інші споживачі, отримуючи при цьому суттєвий економічний ефект.

Результатом магістерської роботи є рпрограмний засіб у вигляді телеграм-бота для виявлення інформаційних вкидів. Отримані результати можуть бути корисні користувачам месенджеру Телеграм для перевірки своїх новин на наявність інформаційних вкидів.

5.2 Визначення рівня конкурентоспроможності розробки

В процесі визначення економічної ефективності науково-технічної розробки також доцільно провести прогноз рівня її конкурентоспроможності за сукупністю параметрів, що підлягають оцінюванню.

В якості аналога для розробки було обрано аналог B.S. Detector. Основними недоліками аналога є можливість аналізу лише веб-сайтів, на основі факт-чекінгу спільнотою користувачів. Також до недоліків можна віднести повільна швидкість виявлення вкидів.

У розробці дана проблема вирішується за допомогою попередньої перевірки методом семантичного аналізу, яка допомагає відсіяти інформацію, що точно не містить інформаційних вкидів. Також система випереджає аналог за такими параметрами як швидкість роботи.

Одиничний параметричний індекс розраховуємо за формулою**:**

$$q_i = \frac{P_i}{P_{базі}}. \qquad (5.1)$$

де q_i – одиничний параметричний індекс, розрахований за i-м параметром;

P_i – значення i-го параметра виробу;

$P_{базі}$ – аналогічний параметр базового виробу-аналога, з яким проводиться порівняння.

Загальні технічні та економічні характеристики розробки представлено в

таблиці 5.4.

Таблиця 5.4 – Основні техніко-економічні показники аналога та розробки, що проектується

Показник	Варіанти		Відносний показник якості	Коефіцієнт вагомості параметра
	Базовий (товар-конкурент)	Новий (інноваційне рішення)		
1	2	3	4	5
Доступність сервісу,%	98	100	1,02	20%
Швидкодія сервісу, мс	150	120	1,25	15%
Вартість обслуговування на одного користувача, грн	35	5	7	20%
Кількість виявлених врахливостей, шт	10	3	3,33	30%
Час повної перевірки, хв	70	25	2,8	15%

Нормативні параметри оцінюємо показником, який отримує одне з двох значень: 1 – пристрій відповідає нормам і стандартам; 0 – не відповідає.

Груповий показник конкурентоспроможності за нормативними параметрами розраховуємо як добуток частинних показників за кожним параметром за формулою:

$$I_{НП} = \prod_{i=1}^{n} q_i \ , \qquad (5.2)$$

де $I_{нп}$ – загальний показник конкурентоспроможності за нормативними параметрами;

q_i – одиничний (частинний) показник за i-м нормативним параметром;

n – кількість нормативних параметрів, які підлягають оцінюванню.

За нормативними параметрами розроблюваний пристрій відповідає вимогам ДСТУ, тому $I_{нп} = 1$.

Значення групового параметричного індексу за технічними параметрами визначаємо з урахуванням вагомості (частки) кожного параметра **:**

$$I_{ТП} = \sum_{i=1}^{n} q_i \cdot \alpha_i \,, \tag{5.3}$$

де $I_{тп}$ – груповий параметричний індекс за технічними показниками (порівняно з виробом-аналогом);

q_i – одиничний параметричний показник i-го параметра;

α_i – вагомість i-го параметричного показника, $\sum_{i=1}^{n} \alpha_i = 1$;

n – кількість технічних параметрів, за якими оцінюється конкурентоспроможність.

Проведемо аналіз параметрів згідно даних таблиці 4.4.

$$I_{mn} = 1{,}02 \cdot 0{,}2 + 1{,}25 \cdot 0{,}15 + 7 \cdot 0{,}2 + 3{,}33 \cdot 0{,}3 + 2{,}8 \cdot 0{,}15 = 3{,}21.$$

Груповий параметричний індекс за економічними параметрами розраховуємо за формулою:

$$I_{ЕП} = \sum_{i=1}^{m} q_i \cdot \beta_i \,, \tag{5.4}$$

де $I_{ЕП}$ – груповий параметричний індекс за економічними показниками;

q_i – економічний параметр i-го виду;

β_i – частка i-го економічного параметра, $\sum_{i=1}^{m} \beta_i = 1$;

m – кількість економічних параметрів, за якими здійснюється оцінювання.

Проведемо аналіз параметрів згідно даних таблиці .

$$I_{ЕП} = 0{,}75 \cdot 0{,}5 + 0{,}86 \cdot 0{,}5 = 0{,}80.$$

На основі групових параметричних індексів за нормативними, технічними та економічними показниками розрахуємо інтегральний показник конкурентоспроможності за формулою :

$$K_{ІНТ} = I_{НП} \cdot \frac{I_{ТП}}{I_{ЕП}} \,, \tag{5.5}$$

$$К_{ІНТ} = 1 \cdot 3,21 / 0,80 = 4.$$

Інтегральний показник конкурентоспроможності $К_{ІНТ} > 1$, отже розробка переважає відомі аналоги за своїми техніко-економічними показниками.

5.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов’язані з проведенням науково-дослідної роботи на тему "Система виявлення інформаційних викидів під час інформаційної війни", під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

5.3.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників ($З_о$) розраховуємо у відповідності до посадових окладів працівників, за формулою:

$$З_о = \sum_{i=1}^{k} \frac{M_{ni} \cdot t_i}{T_p}, \quad (5.6)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, T_p=21 дні.

$$З_о = 20000 \cdot 5 / 21 = 4545 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 5.5 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту	20000	909,1	5	4545
Програміст	12000	545,5	48	26182
Всього				30727

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт НДР на тему "Система виявлення інформаційних викидів під час інформаційної війни" розраховуємо за формулою:

$$З_р = \sum_{i=1}^{n} C_i \cdot t_i , \qquad (5.7)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}} , \qquad (5.8)$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), приймемо $Мм$=6500 грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б)**;**

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середнє число робочих днів в місяці, приблизно T_p = 21 дн;

$t_{зм}$ – тривалість зміни, год.

$C_1 = 6700{,}00 \cdot 1 \cdot 1{,}65 / (21 \cdot 8) = 65{,}8$ грн.
$З_{р1} = 65{,}8 \cdot 1 = 65{,}8$ грн.

Таблиця 5.6 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
1.Підготовчі	16	1	65,8	1052,9
2.Монтажні	40	3	88,8	3553,4
3.Складальні	40	5	111,9	4474,6
4.Налагоджувальні	40	2	72,4	2895,4
5.Випробувальні	16	4	59,8	957,1
Всього				12933,4

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 … 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{дод} = (З_о + З_р) \cdot \frac{Н_{дод}}{100\%}, \quad (5.9)$$

де $Н_{дод}$ – норма нарахування додаткової заробітної плати. Приймемо 11%.

$З_{дод}$ = (30727+12933,4) · 11 / 100% = 4802,67 грн.

5.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$З_н = (З_о + З_р + З_{дод}) \cdot \frac{Н_{зп}}{100\%} \quad (5.10)$$

де $Н_{зп}$ – норма нарахування на заробітну плату. Приймаємо 22%.

$Зн$ = (30727+12933,4+4802,67) · 22 / 100% = 10661,94 грн.

5.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою "Система виявлення інформаційних викидів під час інформаційної війни" передбачають комерціалізацію протягом 3-х років реалізації на ринку.

В цьому випадку основу майбутнього економічного ефекту будуть формувати:

ΔN – збільшення кількості споживачів яким надається відповідна інформаційна послуга у періоди часу, що аналізуються;

N – кількість споживачів яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки, приймемо 1 особа

$Ц_б$ – вартість послуги у році до впровадження інформаційної системи, приймемо 500,00 грн;

$\pm\Delta Ц_о$ – зміна вартості послуги від впровадження результатів, приймемо зростання на 50,00 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta П_i$ для кожного із 3-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою:

$$\Delta П_i = (\pm \Delta Ц_o \cdot N + Ц_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \quad (5.20)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2021 році ставка податку на додану вартість складає 20%, а коефіцієнт λ =*0,8333*;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту). Приймемо ρ =40*%*;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році ϑ =*18%*;

Збільшення чистого прибутку 1-го року:

$\Delta П_1 = (1 \cdot 50 + 500 \cdot 20000) \cdot 0{,}83 \cdot 0{,}4 \cdot (1\text{-}0{,}18/100\%) = 1879100$ грн.

Збільшення чистого прибутку 2-го року:

$\Delta П_2 = (1 \cdot 50 + 500 \cdot (20000 + 15000)) \cdot 0{,}83 \cdot 0{,}4 \cdot (1\text{-}0{,}18/100\%) = 3288460{,}1$ грн.

Збільшення чистого прибутку 3-го року:

$\Delta П_3 = (1 \cdot 50 + 500 \cdot (20000 + 15000 + 10000)) \cdot 0{,}83 \cdot 0{,}4 \cdot (1\text{-}0{,}18/100\%) = 4228005{,}9$ грн.

Приведена вартість збільшення всіх чистих прибутків *ПП*, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{t=1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \quad (5.21)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

Т – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, τ =18%;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$ПП = 1879100 /(1+0,18)^1+3288460,1 /(1+0,18)^2+4228005,9 /(1+0,18)^3=$$
$$= 63007712,52 \text{ грн.}$$

Величина початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{інв} \cdot ЗВ, \qquad (5.22)$$

де $k_{інв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{інв}$=2;

$ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 1303425,72 грн.

$PV = k_{інв} \cdot ЗВ = 2 \cdot 1303425,72 = 2606851,44$ грн.

Абсолютний економічний ефект $E_{абс}$ для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV \qquad (5.23)$$

де $ПП$ – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 63007712,52 грн;

PV – теперішня вартість початкових інвестицій, 2606851,44 грн.

$E_{абс} = ПП - PV$ = 63007712,52 - 2606851,44 = 3700861,07 грн.

Внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1, \quad (5.24)$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, грн;

PV – теперішня вартість початкових інвестицій, грн;

$T_ж$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримування позитивних результатів від її впровадження, 3 роки.

$$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1 = (1+= 3700861,07 /2606851,44)^{1/3} - 1 = 0,57.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{мін}$:

$$\tau_{мін} = d + f, \quad (5.25)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні d =0,1;

f – показник, що характеризує ризикованість вкладення інвестицій, приймемо 0,25.

$\tau_{мін}$ = 0,1+0,25 = 0,35 < 0,57 свідчить про те, що внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Інформаційна технологія онтологічного моделювання бази знань з організації бібліотеки» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_в}, \tag{5.26}$$

де $E_в$ – внутрішня економічна дохідність вкладених інвестицій.

$T_{ок}$ = 1 / 0,57 = 1,8 р.

$T_{ок} <$ 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

5.5 Висновки до розділу

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою "Система виявлення інформаційних викидів під час інформаційної війни" становить 40 балів, що, свідчить про комерційну важливість проведення даних досліджень оскільки рівень комерційного потенціалу розробки вище середнього.

При оцінюванні рівня конкурентоспроможності, згідно узагальненого коефіцієнту конкурентоспроможності розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 4 рази.

Також термін окупності становить 1,8 роки, що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою "Система виявлення інформаційних викидів під час інформаційної війни".

ВИСНОВКИ

Результатом виконання магістерської кваліфікаційної роботи є система для виявлення інформаційних вкидів під час інформаційної війни. За допомогою розробленого засобу перевірено новини з різних телеграм-каналів та веб-сайтів.

Для виявлення інформаційних вкидів використовується семантичний аналіз та побудова і аналіз графів. З метою збільшення точності роботи системи здійснюється додаткова перевірка факт-чекінгу. Це захищає від помилкових результатів, в разі неправильного розуміння значень слів та контексту ботом.

Крім того, бот має функцію онлайн моніторингу телеграм-каналів з бази даних, в котру потрапляють канали з яких раніше виконувалась перевірка новин.

Можна зробити висновок, що використання збору інформації з телеграм-каналів та новинних ресурсів, з її подальшим аналізом та дослідження є перспективним методом в покращенні кібербезпеки окремих користувачів та суспільства в цілому. Також це допоможе покращити інформаційну безпеку нашої держави під час інформаційної війни.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Шпига П. С., Рудник Р. М. Основні технології та закономірності інформаційної війни //Проблеми міжнародних відносин. – 2014. – №. 8. – С. 326-339.

2. Цибенко Д. О. Особливості інформаційних воєн. – 2021.

3. Мельничук В. В. ФІЛОСОФСЬКЕ ОСМИСЛЕННЯ ІНФОРМАЦІЙНОЇ ВІЙНИ В СУЧАСНОМУ СВІТІ //11. Добродум ОВ, Мартинюк ЕІ, Нікітченко ОЕ ФІЛОСОФІЯ. – С. 106.

4. Котеньова І., Яковець А. В. Медіа як інструмент ведення інформаційної війни у міжнародних конфліктах //International scientific innovations in human life. Proceedings of the 11th International scientific and practical conference. – 2022. – С. 607-624.

5. тези

6. NewsGuard Technical Documentation 2023. URL: https://www.newsguardtech.com/ (дата звернення 22.10.2023)

7. What is Hoaxy? 2023. URL: https://hoaxy.osome.iu.edu/ (дата звернення 22.10.2023

8. Snopes Fact-Checking Platform 2022. URL: https://www.snopes.com/about/ (дата звернення 27.10.2023)

9. Limitations of Fact-checking on Debunking Covid-19 Misinformation on Facebook: Case of Faktograf. Hr 2020 URL: https://www.facebook.com/formedia/mjp/programs/third-party-fact-checking (accessed: 29.10.2023)

10. John Brandon. Why Microsoft Teams is so much better than zoom for collaboration. URL: https://www.forbes.com/sites/johnbbrandon/2021/01/17/why-microsoft-teams-is-so-much-better-than-zoom-and-slack-for-collaboration/?sh=5a64bc705cd7 (accessed: 15.10.2023)

11. Драговоз В. Ю. Протидія дезінформації та фейкам в інформаційному середовищі України у період війни Росії проти України. – 2022.

12. Рудзевич А., Павлівна М. Методи машинного навчання в сентимент аналізі текстової інформації : дис. – КПІ ім. Ігоря Сікорського, 2020.

13. Холявка Є. П. Інформаційна технологія виявлення мережевих атак в критичних інформаційних системах. – 2019.

14. Сушко А. О. Виявлення аномалій у поведінці користувачів за допомогою Machine Learning. – 2022.

15. Бісікало О. В. Статистичний аналіз складних залежностей у тексті //Вісник Національного університету Львівська політехніка. Серія: Інформаційні системи та мережі. – 2015. – №. 814. – С. 228-236.

16. Поплавський В. О. Автоматизоване виявлення фейкових новин за допомогою перехресної перевірки джерел з використанням машинного навчання : дис. – КПІ ім. Ігоря Сікорського, 2023.

17. Безлюдний Ю. С. Система розпізнавання та аналізу пропаганди на базі моделі текстової класифікації та методів статистичної обробки даних : дис. – КПІ ім. Ігоря Сікорського, 2023.

18. Документація NodeTelegramBotAPI URL: https://www.npmjs.com/package/node-telegram-bot-api (дата звернення 11.11.2023)

19. «Майже ЗМІ?» URL: https://imi.org.ua/monitorings/majzhe-zmi-yak-telegram-manipulyuye-audytoriyeyu-i49222 (дата звернення 17.11.2023)

20. «Word2vec Tutorial» URL: https://www.tensorflow.org/text/tutorials/word2vec (дата звернення 19.11.2023)

21. Кучевська О. С. Теорія графів та мережеві зв’язки в соціальних мережах : дис. – КПІ ім. Ігоря Сікорського, 2023.

22. «MySQL Documenatation» URL: https://dev.mysql.com/doc/ (дата звернення 20.11.2023)

23. phpMyAdmin URL : https://www.phpmyadmin.net/ (дата звернення 19.11.2023)

24. «DotEnv Documentation» URL: https://www.dotenv.org/docs/ (дата звернення 19.11.2023)

25. «NLP.js Documentation» URL: https://github.com/axa-group/nlp.js/blob/master/docs/v3/README.md (дата звернення 21.11.2023)

26. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.

ДОДАТКИ

Додаток А
ПРОТОКОЛ ПЕРЕВІРКИ
МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Система виявлення інформаційних вкидів під час інформаційної війни

Автор роботи: П`ятак Богдан Олегович

Тип роботи: магістерська кваліфікаційна робота

Підрозділ кафедра захисту інформації ФІТКІ
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність – 82,3 %. Схожість – 17,7 %.

Аналіз звіту подібності (відмітити потрібне):

√ 1. Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

ϒ 2. Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.

ϒ 3. Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ Валентина КАПЛУН
(підпис)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ Богдан П’ЯТАК
(підпис)

Керівник роботи _____ Олеся Войтович
(підпис)

ДОДАТОК Б

Текст програми

```
TOKEN="6583477110:AAF4Oz0N7qrtWeZkyNcgIRoG8prcZR9Ncwk"

DATABASE_NAME="hr401068_education"
DATABASE_HOST="hr401068.mysql.tools"
DATABASE_USER="hr401068_education"
DATABASE_PASSWORD="c_2YP8e8!d"

require('dotenv').config()

const databaseConfig = {
  host: process.env.DATABASE_HOST,
  user: process.env.DATABASE_USER,
  database: process.env.DATABASE_NAME,
  password: process.env.DATABASE_PASSWORD,
};

module.exports = databaseConfig;

const keyboardConstants = Object.freeze({
  MainKeyboard: {
    reply_markup: {
      resize_keyboard: true,
      one_time_keyboard: true,
      keyboard: [
        [
          {
            text: 'Онлайн моніторинг',
          },
          {
            text: 'Перевірка конкретної новини',
          },
        ],
      ],
    },
  },
  CheckKeyboard: {
    reply_markup: {
      resize_keyboard: true,
      one_time_keyboard: true,
      keyboard: [
        [
          {
            text: 'Телеграм канал',
          },
```

```
          {
            text: 'Сайт',
          },
        ],
      ],
    },
  },
});

module.exports = keyboardConstants;


const queriesConstants = Object.freeze({
  GetInfo: (link) => `SELECT * FROM checked WHERE link="${link}"`,
  WriteInfo: (isChannel, link, result) => `INSERT INTO checked(isChannel, link,
result) VALUES (${isChannel},"${link}",${result})`,
  GetAllInfo: `SELECT * FROM checked`
});

module.exports = queriesConstants;


const executeQuery = require('./database.service.js');
const queriesConstants = require('../constants/queries.constants.js');

const readInfo = async (link) => {
  const result = await executeQuery(
    queriesConstants.GetInfo(link)
  );

  return result;
};

const GetAllInfo = async () => {
  const result = await executeQuery(queriesConstants.GetAllInfo);

  return result;
};


const TelegramBot = require('node-telegram-bot-api');
const keyboardConstants = require('./constants/keyboard.constants');
const { readInfo, writeInfo, GetAllInfo } = require('./services/readWrite.service');
const analyze = require('./dump/analyze');
require('dotenv').config()



const Token = process.env.TOKEN;
```

```
const ENV = true;
let isChannel;

const bot = new TelegramBot(Token, { polling: true })
console.log('Bot started')

bot.on('polling_error', console.log)

bot.on('text', async (msg) => {
    const { id } = msg.chat

    if (msg.text === '/start') {
        bot.sendMessage(id, 'Привіт, я бот для виявлення інформаційних вкидів, обери режим роботи', keyboardConstants.MainKeyboard)
        return;
    }

    if (msg.text === 'Онлайн моніторинг') {
        const allResults = await GetAllInfo();

        let responseForUser = 'Моніторю телеграм-канали з бази даних (ви можете розширити мою базу даних, запросивши перевірку новини з каналу, який ще не перевірявся) \n';
        bot.sendMessage(id, responseForUser)
       /* let responseForUser1 = ' '
        for (let i = 0; i < allResults.length; i++) {
            if (allResults[i].result === 0) continue;
            responseForUser1 += `Новина/Джерело: ${allResults[i].link}\nРезультат перевірки: ${allResults[i].result !== 0 ? 'Новина містить ознаки інформаційного вкиду, очікується додаткова перевірка\n' : 'Інформаційного вкиду не виявлено\n'}`
        }*/
        let responseForUser1 = 'Виявлено інформаційний вкид!\n Телеграм канал Соловьев публікує : USA Today: могут ли выборы 2024 года в США привести к коллапсу общества? Некоторые «выживальщики» так думают — и они готовы.\n Дана новина є інформаційним вкидом.'
        const f = () => {
            bot.sendMessage(id, responseForUser1)
        }
        setTimeout(f, 120000)
        return;
    }

    if (msg.text === 'Перевірка конкретної новини') {
        bot.sendMessage(id, 'Що будем перевірять?', keyboardConstants.CheckKeyboard)
        return;
    }

    if (msg.text === 'Телеграм канал') {
```

```
        bot.sendMessage(id, 'Будь ласка, перешліть новину для перевірки')
        isChannel = true;
        return;
    }

    if (msg.text === 'Сайт') {
        bot.sendMessage(id, 'Будь ласка, відправте посилання на новину')
        isChannel = false;
        return;
    }

    const founded = await readInfo(msg.text)
    if (founded.length === 0) {
        bot.sendMessage(id, 'Новина раніше не аналізувалась, починаємо аналіз')
        const result = analyze(ENV, msg.text);
        const response = result ? 'Знайдено ознаки інформаційного вкиду, очікується
додаткова перевірка' : 'Ознак інформаційного вкиду не виявлено';
        await writeInfo(isChannel || true, msg.text, result);
        const t = () => {
            bot.sendMessage(id, response)
        }
        setTimeout(t, 63000)
        return;
    }

    const response = !!founded[0].result ? 'Знайдено ознаки інформаційного вкиду,
очікується додаткова перевірка' : 'Ознак інформаційного вкиду не виявлено';
    bot.sendMessage(id, 'Новина раніше аналізувалась\n' + response);
    return;
})

const writeInfo = async (isChannel, link, result) => {
  try {
    const tmp = await executeQuery(queriesConstants.WriteInfo(isChannel, link,
result));
    return true;
  } catch (error) {
    return false;
  }
};

module.exports = {
  readInfo,
  writeInfo,
  GetAllInfo
};

{
  "name": "infovkudbot",
```

```
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "start": "node src/index.js",
    "test": "jest"
  },
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "@types/node-telegram-bot-api": "^0.63.3",
    "jest": "^29.7.0",
    "prettier": "^3.1.0"
  },
  "dependencies": {
    "dotenv": "^16.3.1",
    "mysql2": "^3.6.5",
    "node-telegram-bot-api": "^0.64.0",
    "senti": "^0.0.2",
    "sentiment": "^5.0.2"
  }
}
```

```
import Util from "./util";
import LetterToSound from "./rita_lts";

const SP = ' ', E = '';

class Analyzer {

  constructor(parent) {
    this.cache = {};
    this.RiTa = parent;
    this.lts = undefined;
  }

  analyze(text, opts) {
    let words = this.RiTa.tokenizer.tokenize(text);
    let tags = this.RiTa.pos(text, opts);
    let features = {
      phones: E,
      stresses: E,
      syllables: E,
      pos: tags.join(SP),
      tokens: words.join(SP)
    }
```

```
  for (let i = 0; i < words.length; i++) {
    let { phones, stresses, syllables } = this.analyzeWord(words[i], opts);
    features.phones += SP + phones;
    features.stresses += SP + stresses;
    features.syllables += SP + syllables;
  }
  Object.keys(features).forEach(k => features[k] = features[k].trim());

  return features;
}

computePhones(word, opts) {
  if (!this.lts) this.lts = new LetterToSound(this.RiTa);
  return this.lts.buildPhones(word, opts);
}

phonesToStress(phones) {
  if (!phones) return;
  let stress = E, syls = phones.split(SP);
  for (let j = 0; j < syls.length; j++) {
    if (!syls[j].length) continue;
    stress += syls[j].includes('1') ? '1' : '0';
    if (j < syls.length - 1) stress += '/';
  }
  return stress;
}

analyzeWord(word, opts = {}) {

  let RiTa = this.RiTa;


  let result = RiTa.CACHING && this.cache[word];
  if (typeof result === 'undefined') {

    let slash = '/', delim = '-';
    let lex = this.RiTa.lexicon();
    let phones = word, syllables = word, stresses = word;
    let rawPhones = lex.rawPhones(word, { noLts: true })
      || this._computeRawPhones(word, lex, opts);

    if (rawPhones) {


      if (typeof rawPhones === 'string') {
        let sp = rawPhones.replace(/1/g, E).replace(/ /g, delim) + SP;
        phones = (sp === 'dh ') ? 'dh-ah ' : sp;
        let ss = rawPhones.replace(/ /g, slash).replace(/1/g, E) + SP;
        syllables = (ss === 'dh ') ? 'dh-ah ' : ss;
```

```
        stresses = this.phonesToStress(rawPhones);
      }
      else {

        let ps = [], syls = [], strs = [];
        rawPhones.forEach(p => {
          let sp = p.replace(/1/g, E).replace(/ /g, delim);
          ps.push((sp === 'dh ') ? 'dh-ah ' : sp);
          let ss = p.replace(/ /g, slash).replace(/1/g, E);
          syls.push((ss === 'dh ') ? 'dh-ah ' : ss);
          strs.push(this.phonesToStress(p));
        });
        phones = ps.join("-");
        syllables = syls.join("/");
        stresses = strs.join("-");

      }
    }

    result = { phones, stresses, syllables };
    Object.keys(result).forEach(k => result[k] = result[k].trim());


    if (RiTa.CACHING) this.cache[word] = result;
  }

  return result;
}

_computeRawPhones(word, lex, opts) {
  return word.includes("-")
    ? this._computePhonesHyph(word, lex, opts)
    : this._computePhonesWord(word, lex, opts);
}


_computePhonesHyph(word, lex, opts) {
  let rawPhones = [];
  word.split("-").forEach(p => {
    let part = this._computePhonesWord(p, lex, opts, true);
    if (part && part.length > 0) rawPhones.push(part);
  });
  return rawPhones;
}


_computePhonesWord(word, lex, opts, isPart) {
  let rawPhones, RiTa = this.RiTa;
  if (isPart) rawPhones = lex.rawPhones(word, { noLts: true });
```

```
    if (!rawPhones && word.endsWith('s')) {
      let sing = RiTa.singularize(word);
      rawPhones = lex.rawPhones(sing, { noLts: true });
      rawPhones && (rawPhones += '-z');
    }



    let silent = RiTa.SILENT || RiTa.SILENCE_LTS || (opts && opts.silent);


    if (!rawPhones) {
      let ltsPhones = this.computePhones(word, opts);
      if (ltsPhones && ltsPhones.length) {
        if (!silent && lex.size()) {
          console.log("[RiTa] Used LTS-rules for '" + word + "'");
        }
        rawPhones = Util.syllablesFromPhones(ltsPhones);
      }
    }

    return rawPhones;
  }
}

const HAS_LETTER_RE = /[a-zA-Z]+/;

export default Analyzer;


class Concorder {

  constructor(parent) {
    this.RiTa = parent;
  }

  concordance(text, options) {

    this.words = Array.isArray(text) ? text : this.RiTa.tokenize(text);
    this.ignoreCase = options && options.ignoreCase || false;
    this.ignoreStopWords = options && options.ignoreStopWords || false;
    this.ignorePunctuation = options && options.ignorePunctuation || false;
    this.wordsToIgnore = options && options.wordsToIgnore || [];

    this._buildModel();

    let result = {};
```

```
    for (let name in this.model) {
      result[name] = this.model[name].indexes.length;
    }
    return result;
  }

  kwic(word, opts) {

    let numWords = 6;
    if (typeof opts === 'object') {
      numWords = opts['numWords'];

      if (opts['text'] && opts['text'].length) this.concordance(opts['text'], opts);
      if (opts['words'] && opts['words'].length) this.concordance(opts['words'],
opts);
    }
    else if (typeof opts === 'number') {
      numWords = opts;
    }

    if (typeof numWords !== 'number') numWords = 6;

    if (!this.model) throw Error('Call concordance() first');
    let result = [];
    let value = this._lookup(word);
    if (value) {
      let idxs = value.indexes;
      for (let i = 0; i < idxs.length; i++) {
        let sub = this.words.slice(Math.max(0, idxs[i] - numWords),
          Math.min(this.words.length, idxs[i] + numWords + 1));
        if (i < 1 || (idxs[i] - idxs[i - 1]) > numWords) {
          result.push(this.RiTa.untokenize(sub));
        }
      }
    }
    return result;
  }

  count(word) {
    let value = this._lookup(word);
    return value && value.indexes ? value.indexes.length : 0;
  }



  _buildModel() {
    if (!this.words || this.words.length == 0) throw Error('No text in model');
    this.model = {};
    for (let j = 0; j < this.words.length; j++) {
```

```
      let word = this.words[j];
      if (this._isIgnorable(word)) continue;
      let _lookup = this._lookup(word);



      if (!_lookup || typeof _lookup !== 'object') {
        _lookup = { word: word, key: this._compareKey(word), indexes: [] };
        this.model[_lookup.key] = _lookup;
      }
      _lookup.indexes.push(j);
    }
  }

  _isIgnorable(key) {
    if ((this.ignorePunctuation && this.RiTa.isPunct(key)) ||
      (this.ignoreStopWords && this.RiTa.isStopWord(key))) return true;
    for (let i = 0; i < this.wordsToIgnore.length; i++) {
      let word = this.wordsToIgnore[i];
      if (key === word || (this.ignoreCase && key.toUpperCase() ===
word.toUpperCase())) {
        return true;
      }
    }
  }

  _compareKey(word) {
    return this.ignoreCase ? word.toLowerCase() : word;
  }

  _lookup(word) {
    let key = this._compareKey(word);
    return this.model[key];
  }
}

export default Concorder;
```

Додаток В

ІЛЮСТРАТИВНА ЧАСТИНА

СИСТЕМА ПОШУКУ ТА АНАЛІЗУ НЕБЕЗПЕЧНОГО КОНТЕНТУ ІНФОРМАЦІЙНИХ РЕСУРСІВ

Узагальнена архітектура системи

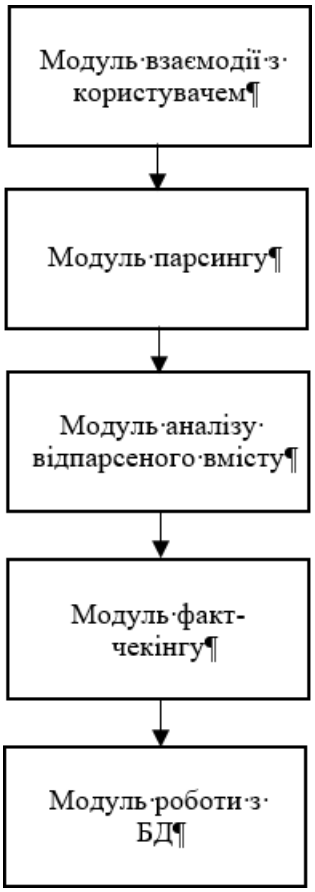

Схема роботи модулю взаємодії з користувачем

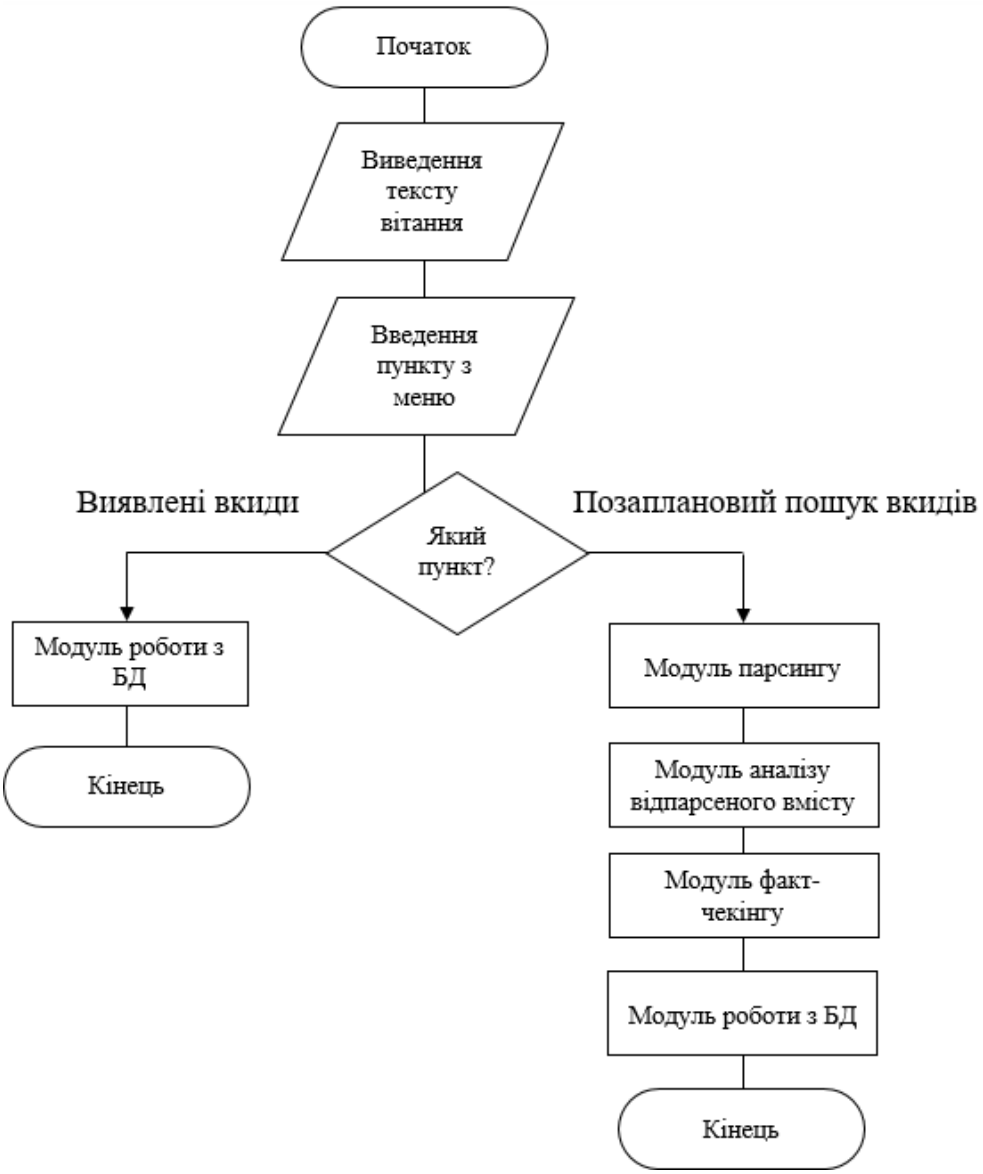

Схема роботи модулю парсингу

Схема роботи модулю запису в базу даних

Схема роботи модулю аналізу

Схема роботи модулю факт-чекінгу

Токенізація тексту

```
tokenizator(text, options) {

  this.words = Array.isArray(text) ? text : this.RiTa.tokenize(text);
  this.ignoreCase = options && options.ignoreCase || false;
  this.ignoreStopWords = options && options.ignoreStopWords || false;
  this.ignorePunctuation = options && options.ignorePunctuation || false;
  this.wordsToIgnore = options && options.wordsToIgnore || [];

  this._buildModel();

  let result = {};
  for (let name in this.model) {
    result[name] = this.model[name].indexes.length;
  }
  return result;
}
```

Правила для визначення значущості висловлювання

```
let rules = type === SING ? SING_RULES : PLUR_RULES;
for (let i = 0; i < rules.length; i++) {
  let rule = rules[i];
  if (rule.applies(check)) {
    dbug && console.log(word + ' (' + (type === SING ? 'singularize' : 'pluralize')
      + ') hit ' + (type === SING ? 'singular' : 'plural')
      + (i < rules.length - 1 ? ' rule #' + i : ' DEFAULT rule'), rule);
    return rules[i].fire(word);
  }
}
```

Результат тестування бота

Результат тестування функції моніторингу