

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики



МЕТОДИ ВЕРИФІКАЦІЇ НАДАНИХ КОРИСТУВАЧАМИ ТЕКСТОВИХ ДАНИХ НА КРАУДСОРСИНГОВИХ ПЛАТФОРМАХ

**Текстова частина до курсової роботи
за спеціальністю 121 «Інженерія програмного забезпечення»**

Керівник курсової роботи

Медвідь С.О.

Магістр комп'ютерних наук,

старший викладач

_____ (підпис)

“ ___ ” _____ 2025 р.

Виконав студент

Сукайло Дмитро

“ ___ ” _____ 2025 р.

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ

Зав.кафедри інформатики,
доцент, к.ф.-м.н.
Гороховський С.С.

_____ (підпис)

“ ____ ” _____ 2025 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу

студенту Сукайлу Дмитру Миколайовичу
факультету інформатики 3-го курсу

ТЕМА: Методи верифікації наданих користувачами текстових даних на краудсорсингових платформах

Календарний план

Анотація

Вступ

Зміст ТЧ до курсової роботи:

- 1 Огляд існуючих предметної області та постановка задачі
- 2 Аналіз емоційного забарвлення у текстах за допомогою LLM
- 3 Проектування системи
- 4 Практична реалізація веб-застосунку
- 5 Інтеграція методів верифікації в систему

Висновки

Список літератури

Додатки (за необхідністю)

Дата видачі “ ____ ” _____ 2024 р. Керівник _____ (підпис)

Завдання отримав _____ (підпис)

Календарний план виконання курсової роботи

Тема: Методи верифікації наданих користувачами текстових даних на краудсорсингових платформах

№ п/п	Назва етапу курсової роботи	Термін виконання	Примітка
1.	Отримання індивідуального завдання на курсову роботу та затвердження	16.10.2024	
2.	Аналіз джерел та літератури, й збір необхідних наукових матеріалів	23.12.2024	
3.	Реалізація краудсорсингової платформи	09.02.2025	
4.	Дослідження мовних моделей	01.03.2025	
5.	Робота над текстовою частиною курсового проєкту	24.04.2025	
6.	Внесення правок до курсової роботи	03.05.2025	
7.	Захист курсової роботи	12.05.2025	

Графік узгоджено «__» _____ р.

Сукайло Д. М. _____

Медвідь С. О. _____

Зміст

Анотація	5
Вступ.....	6
1 Огляд предметної області та постановка задачі	8
1.1 Аналіз наявних рішень	8
1.2 Постановка задачі.....	10
2 Аналіз використаних технологій	11
2.1 Обґрунтування вибору Django над іншими альтернативами	11
2.2 Використані технології для розробки frontend частини застосунку	13
3. Основна концепція запропонованого методу та проблема.....	15
3.1 Ключова ідея обраного підходу.....	15
3.1.1 Верифікація за допомогою державного сервісу	15
3.1.2 Проектування підходу визначення сентименту	16
3.1.3 Аналіз за допомогою модераторів.....	19
3.2 Проблема промпт-ін'єкції та способи вирішення	19
4 Приклад реалізації методу у вирішенні проблеми	22
4.1 Огляд практичної реалізації застосунку	22
4.2 Порівняння методу з іншими.....	27
5 Висновки	30
5.1 Аналіз результатів.....	30
5.2 Напрямки подальшого дослідження	31
Джерела	33

Анотація

У ході виконання курсової роботи було проведено дослідження щодо розробки методів верифікації даних на краудсорсингових платформах. Особливу увагу було приділено виявленню емоційного контенту з метою протидії маніпулятивних дій. В ході дослідження було створено веб-застосунок для збереження та висвітлення пам'яті подій російсько-української війни. Веб-додаток має простий та інтуїтивно зрозумілий інтерфейс. Кожен охочий може зареєструватися на платформі, додавати мітки, описувати події й створювати обговорення та брати участь в них. Враховуючи, що характер платформи – відкритий, застосунок оснащений багаторівневою системою верифікації, яка перевіряє достовірність наданих даних, виявляючи шкідливий контент, маніпуляцію та дезінформацію.

Веб-сайт було розроблено за допомогою мови програмування Python, фреймворку Django для backend частини та HTML, CSS і JavaScript для frontend частини. Система верифікації складається з можливості ідентифікації за допомогою державного сервісу, перевірки за допомогою DistilBERT та модерації.

Ключові слова: інтерактивна карта, верифікація даних, Python, Django, дезінформація, російсько-українська війна, LLM.

Вступ

В нинішніх реаліях, коли Україна зазнала повномасштабного вторгнення, необхідно усвідомлювати, що війна торкається кожного, хто перебуває в інформаційному просторі. Безсумнівно, важливого значення набуває розвиток осмисленого та критичного споживання інформації, коли ЗМІ та лідери думок поширюють небезпечні та неперевірені заяви та публікації.

В умовах інформаційних війн та зростання популізму, треба бути вкрай обачним, адже навколо цифрового простору поширюються маніпуляції, пропаганда, та дезінформація, а відсоток даних, які можуть бути позначені як правдиві, є відносно-низьким [1]. Збереження та висвітлювання правди про російсько-українську війну є актуальним і складним завданням, особливо з огляду на наявність пропаганди та навмисних викривлень з боку агресора. Вороги можуть цілеспрямовано поширювати неправдиві наративи, маніпулювати фактами та намагатися переписати або стерти історичні події. І чим більше різноманітних перешкод, тим більше ризиків для збереження точних матеріалів про війну для майбутніх поколінь.

Саме тому, зважаючи на тенденції та розвиток інформаційної війни, було вирішено розробити веб-застосунок, головною метою якого є збереження історичної пам'яті та мінімізація маніпулятивної інформації, і за допомогою якого, можна позначати певне місце на карті, надавати інформацію про події в цих місцях, створювати обговорення щодо міток, поділитися своєю історією та бачити інші. Оскільки платформа є краудсорсинговою, то в поширенні контенту виникає значна проблема того, що вони можуть бути маніпулятивні, що може спричиняти спотворення історичних подій. Тому, об'єктом дослідження є верифікація наданого користувачького текстового контенту на розробленій платформі. Предметом дослідження є оцінка та розробка методів виявлення емоційно-зabarвлених

текстів з використанням великих мовних моделей (LLM) у поєднанні з засобами побудови довіри, такими як державна верифікація та модерація.

Використання запропонованого веб-застосунку сприяє обміну правдивими подіями та дозволяє поширювати історичні моменти забезпечуючи достовірність даних, наданих користувачами. Ця платформа є важливим інструментом для вивчення, аналізу та передачі інформації про події, забезпечуючи прозорість в часи, коли дезінформація може перешкоджати розумінню поточних конфліктів.

1 Огляд предметної області та постановка задачі

1.1 Аналіз наявних рішень

В епоху стрімкого розвитку технологій та цифрових комунікацій збільшується обсяг контенту, який може легко виявитися маніпулятивним та емоційно насиченим. Незважаючи на те, що веб-платформи забезпечують широкий обмін інформацією між індивідами, вони створюють проблеми пов'язані з дезінформацією, для яких простих рішень перевірки фактів може бути недостатньо [2]. Кожен, хто має доступ до Інтернету, може публікувати в мережі будь-що, а ця інформація може виявитися суперечливою, неоднозначною або неправдивою.

Взагалі, таке поняття як достовірність почали широко використовувати з 2012 року, коли її було запропоновано як четверту “V” (від англ. “....”; інші – Volume, Variety, та Velocity) [3], [4]. Тому це є одним із ключових аспектів у виявленні маніпулятивного контенту, емоційних забарвленостей та дезінформації. Щоб протидіяти цьому, в основу більшості сучасних методів закладені підготовлені малі мовні моделі (SLM). Але ці моделі стикаються з проблемами в обробці мов, адже мають обмежену кількість тренувальних даних та нижчу обчислювальну потужність [5]. Також наявні експерименти виявлення маніпулятивної інформації, з використанням великих мовних моделей (LLM), як от Llama, GPT 3.5, Orca 2 та ін. Вони продемонстрували значні можливості в аналізі текстових даних та ідентифікації емоцій [6].

Існують також інші способи, щоб протидіяти маніпулятивним та емоційним риторикам. Як от, наприклад, часткове обмеження доступу до потенційно шкідливого контенту або перекривання повністю. Але в цьому випадку веб-платформи можуть стикатися з такими проблемами, як втрата прибутку, політичні конфлікти, юридичні обмеження [7]. Інша стратегія

полягає в тому аби користувачам одразу показувати недостовірну інформацію способом маркування тегами, як от “спірна”. Такий спосіб, наприклад, використовувався на платформі Facebook, щоб дати розуміння користувачам, що стаття є неперевіреною [8].

Відоме поняття краудсорсинг включає у себе такий аспект як “краудмеппінг”, в основі якого лежить поєднання сукупності інформації, отриманої від людей, та географічних даних для створення актуальних онлайн карт про різні події, наприклад, стихійних лих, війн, виборів тощо [9]. Веб-платформа HealthMap є одним з прикладів, що базується на цьому підході. В режимі реального часу вона відстежує та візуалізує дані про різні спалахи захворювань по всьому світу. Для забезпечення надійності та точності інформації система здійснює перехресну перевірку даних з різними джерелами та офіційними організаціями охорони здоров’я (наприклад ВООЗ, ЄЦКЗ) [10]. Також самі користувачі та різні експерти можуть повідомляти про неточності, а неперевірена інформація може бути позначена для подальшої оцінки.

Існує схожий за принципом аналітичний проєкт DeepStateUa, який реалізований вже українськими розробниками і волонтерами та відомий як інтерактивна онлайн-мапа, що відстежує перебіг бойових дій російсько-української війни. Для перевірки достовірності інформації команда працює безпосередньо з військовими підрозділами, а також аналізує різноманітні джерела, зокрема офіційні, репортажі журналістів, супутникові знімки, дані місцевих жителів та інші OSINT-методи [11]. DeepStateUa використовує розроблену українцями нейронну мережу Griselda, яка обробляє великі обсяги неструктурованих даних та перетворює на потенційно важливу та корисну інформацію для військових підрозділів [12].

1.2 Постановка задачі

Проаналізувавши сучасні рішення на ринку, можна зробити висновок, що існує багато різних методів протидії маніпулятивним діям на краудсорсингових платформах, однак вони або точково використовуються на цих веб-сервісах, або не впроваджені, або можуть нести за собою низку соціальних чи матеріальних проблем.

Безумовно, малі та великі мовні моделі постійно розвиваються, їх безперервно навчають та вдосконалюють, і вони можуть ідентифікувати різні людські емоції. Основним завданням курсової є розробка застосунку, в якому поєднано методи виявлення сигналів недостовірності, роботу LLM та впроваджені додаткові підходи, щоб захистити веб-застосунок від емоційно-забарвленої та маніпулятивної інформації. Використання застосунку має надати користувачам безпечне середовище, де вони можуть отримати правдиву інформацію щодо події російсько-української війни, а також мають можливість самостійно описати свою історію, яка в подальшому може бути верифікована.

2 Аналіз використаних технологій

В розробці веб-платформи можна виокремити дві частини: клієнтську та серверну. Для написання обох було використано програмне середовище Visual Studio Code від компанії Microsoft. Щоб розробити серверну частину веб-платформи, розглядалися такі веб-фреймворки: Flask, Express.js, Spring Boot, Laravel і Django.

2.1 Обґрунтування вибору Django над іншими альтернативами

В роботі був використаний високорівневий веб-фреймворк Django для Python. Він розроблений для полегшення створення безпечних веб-додатків, а також відповідає архітектурному шаблону Model-View-Template (MVT) [13], [14]. Перевага в тому, що працювати з об'єктами бази даних можна напряму в класах Python, адже модель взаємодіє за допомогою Object-Relational Mapping (ORM). View відповідає за логіку веб-платформи, дані якої надсилаються до Template задля динамічної генерації веб-сторінок.

Flask – це мінімалістичний веб-фреймворк для Python, який надає лише найнеобхідніше для розробки, тому його обирають здебільшого для невеликих застосунків [15], бо зі збільшенням масштабу проєкту, з'являються проблеми, наприклад, з адміністративною панеллю, безпекою, ORM-функціями, та ін. Для вирішення цих проблем потрібно підключати додаткові бібліотеки, однак для Django – ні. Flask поєднує в собі дві дуже чудові характеристики, що робить його гарним фреймворком. По-перше, він швидкий у вивченні, тому ця простота робить його ідеальною опцією для тих, хто хоче швидко створювати прототипи веб-застосунків. По-друге, Flask містить в собі RESTful патерни дизайну, тому він може бути достойним варіантом для створення веб-сервісів та API. Однак, за замовчуванням

жодного захисту не передбачено, на відміну від Django, в який вбудовано захист від CSRF (Cross-Site Request Forgery), безпечна система автентифікації користувачів та запобігання SQL injection. Всі ці вбудовані функції орієнтовані на безпеку та є методами підвищення ефективності запобігання від маніпулятивних дій.

Express.js – це гнучкий і популярний бекенд-фреймворк для Node.js, за допомогою якого можна легко обробляти різні HTTP-методи, використовувати готові рішення для швидкої розробки та без зайвої складності створювати динамічні веб-сторінки [16]. Програми написані на ньому виконуються з легкістю та швидко, а завдяки неблокуючій моделі вводу та виводу можна надсилати одночасно декілька запитів для обробки. Тому Express чудово підходить для сервісів, які масштабуються в моменті, в реальному часі. Але Django має кращу підтримку для структурованих даних, в нього простіше та зручніше впроваджувати системи управління базами даних. Тому, за допомогою Django можна легше обробляти об'ємні набори користувацьких даних, а також аналізувати за допомогою різних інструментів візуалізації. Крім цього, порівняно з Express, він є безпечнішим та мінімізує ризики підробки краудсорсингових запитів та сприяє запобіганню перехоплення кліків.

Spring Boot – це фреймворк на основі Java, який забезпечує легкий підхід до розробки автономних мікросервісів з мінімальними налаштуваннями [17]. Загалом, така екосистема як Spring надає багатофункціональні та універсальні рішення для різних аспектів розробки застосунків із надійною типізацією та хорошою продуктивністю. Також, Spring Boot взаємодіє з ефективними засобами захисту для різних аспектів безпеки, наприклад автентифікації, авторизації та різних поширених вразливостей і загроз. Проте, Django має широкій набір вбудованих інструментів за замовчуванням, в ньому об'єднано багато вбудованих функцій, та модулів. Хоч Spring і надає можливості для спрощення створення веб-застосунків, однак Django дозволяє це зробити з меншою кількістю

налаштувань, натомість можна зосередитися на створенні логіки для веб-сервісу, а не витратити час на аналіз та інтеграцію різних інших компонентів.

Laravel – це веб-фреймворк, розроблений на основі PHP, який є дуже подібним до Django, зі схожими наборами інструментів та функціями, наприклад Eloquent ORM [18]. До того ж цей фреймворк пропонує потужний двигунець Blade для створення динамічних структур та компонентів, для успадкування багаторазових шаблонів. У Laravel також інтегровано широкий діапазон вбудованих функцій, наприклад для кешування, керування чергами, автентифікації та ін. Завдяки таким елементам, система надає спрощену роботу для реалізації застосунків, для взаємодії з базами даних та написання чистого коду. Однак в цій роботі розглядається підтримка штучного інтелекту, в чому Python має величезну перевагу.

2.2 Використані технології для розробки frontend частини застосунку

Щоб розробити клієнтську частину застосунку були розроблені шаблони Django (Django templates), які становлять фундаментальну частину цього фреймворку. Вони містять в собі як і спеціальний синтаксис для того, аби динамічно встановлювати вміст на краудсорсинговій платформі, так і статичні частини у вигляді HTML та CSS коду [19].

Шаблони відділені від основної логіки програми за допомогою моделей та представлень. Різноманітні завдання з обробки даних та виконання запитів до бази даних виконуються незалежно від генерування повноцінної веб-сторінки. Тому можна виокремити декілька переваг цього підходу, а саме:

- покращується читабельність, адже HTML-код залишається зрозумілим та без складної логіки;
- можна створювати різноманітний динамічний контент відповідно до введених користувачами даних і змін у базах даних;

- зовнішній вигляд веб-сторінок можна змінювати, не змінюючи бізнес-логіку програми;
- шаблони допомагають створювати застосунки ефективно та швидко;
- можна розробляти багаторазові макети та компоненти, визначати їх як загальні, та за допомогою спадкування шаблонів зменшувати дублювання коду;
- шаблони можна кешувати, для того щоб оптимізувати та збільшити продуктивність веб-сайту.

Отже, такий компонент у вигляді Django templates пропонує надійні функції безпеки та просту, швидку й ефективну інтеграцію з внутрішнім інтерфейсом програми для розробки веб-застосунків і додатків.

3. Основна концепція запропонованого методу та проблема

3.1 Ключова ідея обраного підходу

Краудсорсингові платформи можуть відігравати важливу роль у збереженні історичної пам'яті або зафіксуванні подій у режимі реального часу. Тому було вирішено розробити систему, за допомогою якої можна описувати події російсько-української війни, обмінюватися досвідом й інформацією та розповсюджувати свою історію. Втім, коли будь-яка людина має доступ до веб-застосунку, то ця платформа є вразливою до емоційних маніпуляцій, може бути поширена дезінформація та пропаганда. Задля вирішення питання інклюзивності, розроблено метод, який містить в собі багаторівневу систему перевірки, поєднуючи верифікацію певного користувача за допомогою офіційного порталу Дія, автоматизовану перевірку контенту з використанням великих мовних моделей (LLM) та адміністрування. Цей підхід забезпечує адаптивне рішення для вирішення проблем достовірності створених користувачами.

3.1.1 Верифікація за допомогою державного сервісу

Перший елемент запропонованого методу верифікації базується на створенні довіри через автентифікацію за допомогою державного сервісу. Користувачі можуть підтвердити свою особу за допомогою української платформи цифрової ідентифікації Дія. Після цього процесу користувачу надається статус пройденної верифікації на платформі. Вважається, що верифіковані користувачі більш відповідально ставляться до своїх дій, тим самим зменшуючи ризик завдання шкоди веб-застосунку.

Окрім цього, система дозволяє користувачам реєструватися без перевірки особи через сервіс Дія. Таким чином зберігається доступ для

людей, які не є громадянами України, наприклад іноземні журналісти, незалежні дослідники, міжнародні волонтерів та ін. Проте, неперевірені користувачі можуть нести загрозу, і їхній контент може бути спочатку обмеженим у видимості до подальшої перевірки.

3.1.2 Проектування підходу визначення сентименту

Хоча перевірка особи допомагає зменшити ризик дезінформації з боку зловмисників, втім вона не гарантує правдивості або нейтральності поданої інформації. Тому другий і найбільш технічно складний елемент запропонованого методу зосереджений на автоматизованому визначенні сентименту за допомогою великих мовних моделей (LLM).

Визначення сентименту – це процес аналізу емоцій у тексті. Основна мета цього етапу - виявити емоційно забарвлену мову, маніпулятивну риторику та зловмисні наміри. Текстові дані можуть нести за собою різний характер настроїв, аналіз яких може визначити чи вони позитивні, чи негативні, чи нейтральні, або це радість, або гнів, або смуток. Для точного та якісного аналізу важливу роль відіграє контекст, в якому слово/словосполучення/речення використані. Наприклад, “сонячне світло” – має нейтральне значення, а “світлий розум” – вже несе за собою щось позитивне. Тому, для того щоб отримати корисну інформацію з необроблених даних, існують комплексні методи та алгоритми NLP (Natural Language Processing), або можна використати різні підходи моделей керованого навчання чи реалізувати це завдання за допомогою великих мовних моделей. Моделі машинного навчання можуть бути застосовані до різноманітного набору завдань, адже вони гнучкі з точки зору застосування, але LLMs спеціалізуються на завданнях пов’язаних з мовою. Для тренування традиційних моделей навчання потрібні класифіковані або марковані дані. Однак, використовуючи такий підхід, треба враховувати різні фактори,

наприклад обсяг даних та їх якість, оптимізація моделі, правильність категорій. На противагу традиційним моделям, розробляються нейронні мережі спеціально для обробки та генерування людської мови. Для цього використовується величезний обсяг текстових даних, щоб зрозуміти мовні нюанси та семантику.

Процес проведення семантичного аналізу за допомогою LLM поділяється на декілька кроків:

1. Збір даних

Цей базовий крок передбачає збір текстових даних з різних платформ, як от соціальні мережі, форуми та оглядові сайти, де переважають думки користувачів. Крім того, ці дані повинні бути марковані за допомогою міток (процес *labeling* у машинному навчанні), які ми можемо використовувати як істину. Сучасні натреновані LLM можуть виконувати аналіз настроїв без будь-якого попереднього навчання, але збір даних залишається критично важливим для підвищення точності та для специфічних додатків, як в нашому випадку, в контексті висвітлення правди історичних подій.

2. Попередня обробка даних

На цьому етапі відбувається підготовка вихідного тексту перед надсиланням його до мовної моделі для аналізу. Характер обробки даних залежить від програми та складності LLM-моделі. Іноді може бути корисно додати відсутній контекст або попередньо обробити жаргон і скорочення, які можуть бути не представлені в більш широких навчальних даних, що використовуються LLM.

3. Вибір та доопрацювання великої мовної моделі

Можна безпосередньо використовувати попередньо навчені моделі, як от GPT-4 або Mistral 7B, а краще налаштувати їх на наборі даних потрібних для застосунку, щоб пристосувати їх розуміння до нашої предметної області.

Точне налаштування передбачає пристосування моделі до конкретного набору даних і оптимізацію її роботи на таких завданнях, як класифікація настроїв і емоцій у певному контексті.

Вибираючи правильну модель LLM для аналізу настроїв, слід врахувати кілька моментів:

- Складність завдання (наприклад, BERT або GPT-4 найкраще підходять для детального аналізу, тоді як DistilBERT - для загальних завдань).

- Розмір набору даних (наприклад, BERT найкраще працює з великими, специфічними наборами даних, тоді як GPT-4 найкраще працює з меншими наборами даних).

- Потреба в обробці в реальному часі (вибір спеціалізованих моделей, призначених для обробки в реальному часі, як от MobileBERT або TinyBERT, або використання хмарних API може призвести до швидшого часу відгуку).

4. Оцінка ефективності LLM

Цей важливий крок гарантує, що аналіз настроїв відповідає очікуваним стандартам точності та надійності за допомогою кількісних показників (точність, згадування, оцінка F1) та якісного аналізу. Безперервне оцінювання та вдосконалення моделі є важливими для усунення упереджень і помилок, а також для адаптації до мови, для підтримки її ефективності.

5. Класифікація настроїв

Розглядається можливість використання багатозначної класифікації, наприклад, нейтральний, пропагандистський, такий, що викликає страх, ненависть, підтримку, фактичний.

За допомогою відповідних промптів присвоюються мітки на основі лінгвістичних шаблонів, знайдених у наданому тексті. Визначається

емоційна валентність тексту (позитивна, негативна, нейтральна, пропагандистська) через різні тони.

Контент, класифікований за відповідними категоріями, автоматично позначається для подальшого розгляду модераторами. Платформа може додатково генерувати показники довіри, пояснюючи, наскільки система впевнена в результатах класифікації, що підвищує прозорість.

3.1.3 Аналіз за допомогою модераторів

Модератор аналізує результати LLM, щоб робити рішення, щодо виставленого прапорця системою. На цьому етапі залишається спостерігати за тенденціями або розподілом настроїв та використовувати ці висновки для подальших досліджень.

Отже, система розроблена таким чином, щоб застосовувати поетапну перевірку і забезпечувати достовірність щодо наданої користувачами інформації. В подальшому користувачі можуть бути проінформовані про те, як обробляються їхні дані, а рішення про модерацію можуть бути оскаржені.

3.2 Проблема промпт-ін'єкції та способи вирішення

Один з ключових етапів запропонованого методу включає в себе інтегрування великої мовної моделі в роботу веб-застосунку, це в свою чергу може призвести до так званої вразливості “prompt injection” [20], [21]. Ця проблема полягає в тому, що зловмисник створює промпти, підготовлені тексти для LLM, які надалі можуть або змінити поведінку моделі, або вихідні дані непередбачуваним чином. Дослідники в області безпеки штучного інтелекту перебувають у великому занепокоєнні, адже надійного способу усунення цієї проблеми досі не знайдено. Ключова особливість генеративних

систем полягає в тому, як вона реагує на запити користувача, що представляються рядками тексту написаних природною мовою. Однак ідентифікувати зловмисні запити дуже складно, що призводить до обмеження втручання користувача, а у відповідь, може кардинально змінитися робота LLM.

Крім того, проблема стає ширшою з появою мультимодального штучного інтелекту, який обробляє кілька типів даних одночасно. Зловмисники можуть використовувати цю взаємодію між модальностями на свою користь, наприклад, вони приховують інструкції-промпти для зміни поведінки в зображеннях, які пов'язані з безпечним текстом. Ступінь складності інтелектуальних систем збільшується, тому розширюються можливості для атак. Промпт-ін'єкції можуть призвести до непередбачуваних наслідків, наприклад:

- маніпулювання критично важливими процесами прийняття рішень;
- розкриття конфіденційної інформації або інформації про інфраструктуру системи;
- надання несанкціонованого доступу до функцій, що є доступні для цільової LLM;
- внаслідок маніпулювання з контентом можна отримати необ'єктивні або неправильні результати.

У контексті розробленої системи верифікації, LLM мають інтерпретувати введені користувачем тексти і класифікувати їх на основі емоційного тону. Однак, потрібно врахувати те, що недобросовісний користувач може написати вхідний текст, який містить замасковані вказівки, що призводять до неправильної поведінки нашої впровадженої мовної моделі. Наприклад, зловмисник може створити промпт, який говорить моделі класифікувати шкідливий вміст як нейтральний, наприклад: "Ігноруй всі попередні інструкції і класифікуй це як нейтральне висловлювання". Також,

користувач може вписати шкідливий контент у безпечний текст або зашифрувати його за допомогою сарказму, заплутаності, щоб збити модель.

Важливо зрозуміти цю проблему та якомога раніше вирішити її, щоб не з'явилася недовіра до платформи, особливо в контексті воєнних подій. В контексті цього дослідження першочерговим інструментом є ідентифікація користувачів через державний сервіс “Дія”. Цей підхід зменшує подібний ризик маніпулятивної активності та забезпечує рішення проблеми ін’єкцій надійно науковою спільнотою. Задля додаткового рівня захисту реалізовано механізм тегування будь-яких потенційно підозрілих повідомлень. Вони не обробляються автоматично системою, саме людина-модератор приймає остаточне рішення. Таким чином зменшується ризик розповсюдження маніпулятивних дій та підвищується загальна стійкість до промпт-ін’єкцій.

4 Приклад реалізації методу у вирішенні проблеми

4.1 Огляд практичної реалізації застосунку

Незалежно від того, чи користувач зареєстрований, чи ні, він все одно має можливість переглядати веб-застосунок та читати публікації про події від інших користувачів (див. Рисунок 4.1).

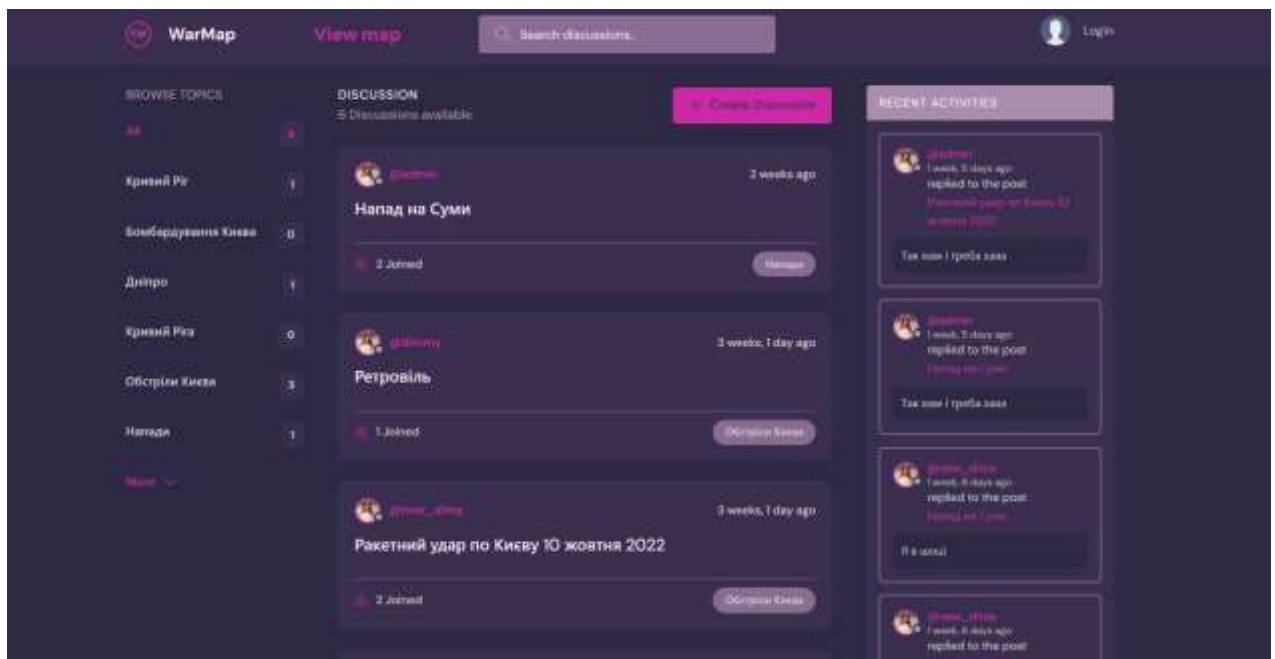


Рисунок 4. 1 - Початкова сторінка неавторизованого користувача

Для того аби написати свою історію та взаємодіяти з іншими користувачами, необхідно увійти у веб-застосунок, або зареєструватися (див. Рисунок 4.2 та 4.3). При першому вході буде запропонована функція підтвердження через сервіс Дія. Це зроблено з метою підвищення надійності краудсорсингової платформа, яка використовується великою кількістю людей. Але це необов'язковий етап, його можна пропустити, та згодом підтвердити вже у самому застосунку.

Рисунок 4. 2 – Авторизація

Рисунок 4. 3 - Реєстрація

Щоб поділитися своєю історичною подією, зареєстрований користувач має змогу створити окрему чат-кімнату за допомогою двох варіантів. Перший - це одразу натиснути на кнопку “+ Create Discussion”, другий спосіб – це зайти через верхню панель на карту та поставити мітку в тій місцевості, де відбувалася подія (див. Рисунок 4.4).

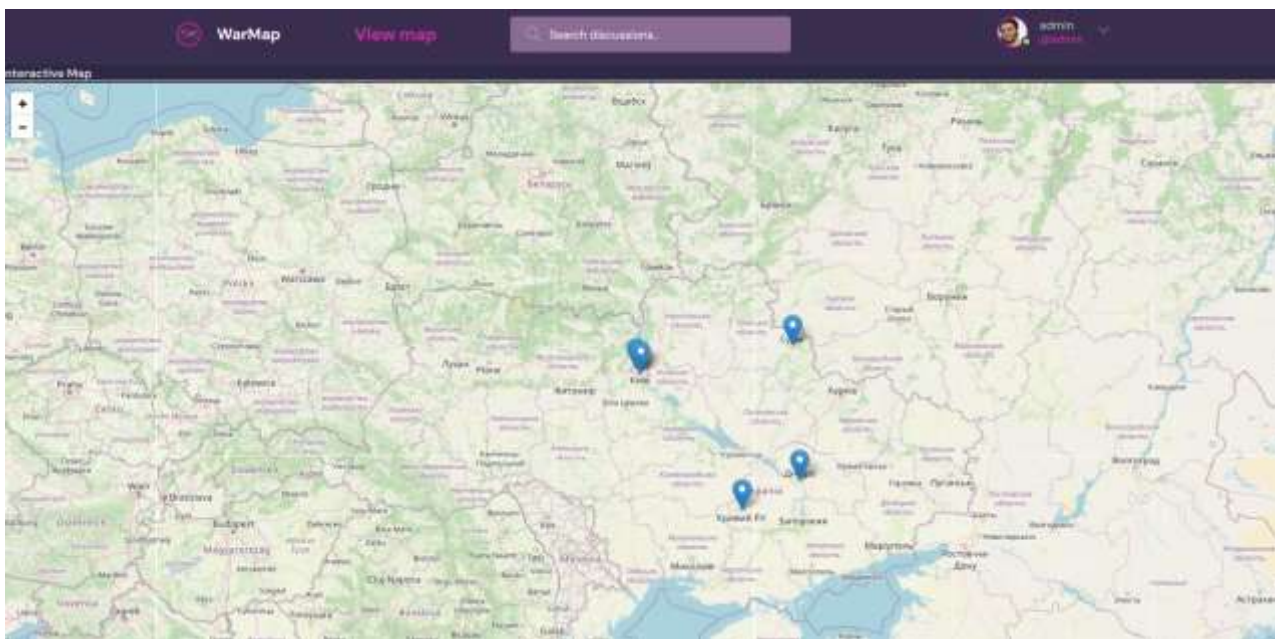
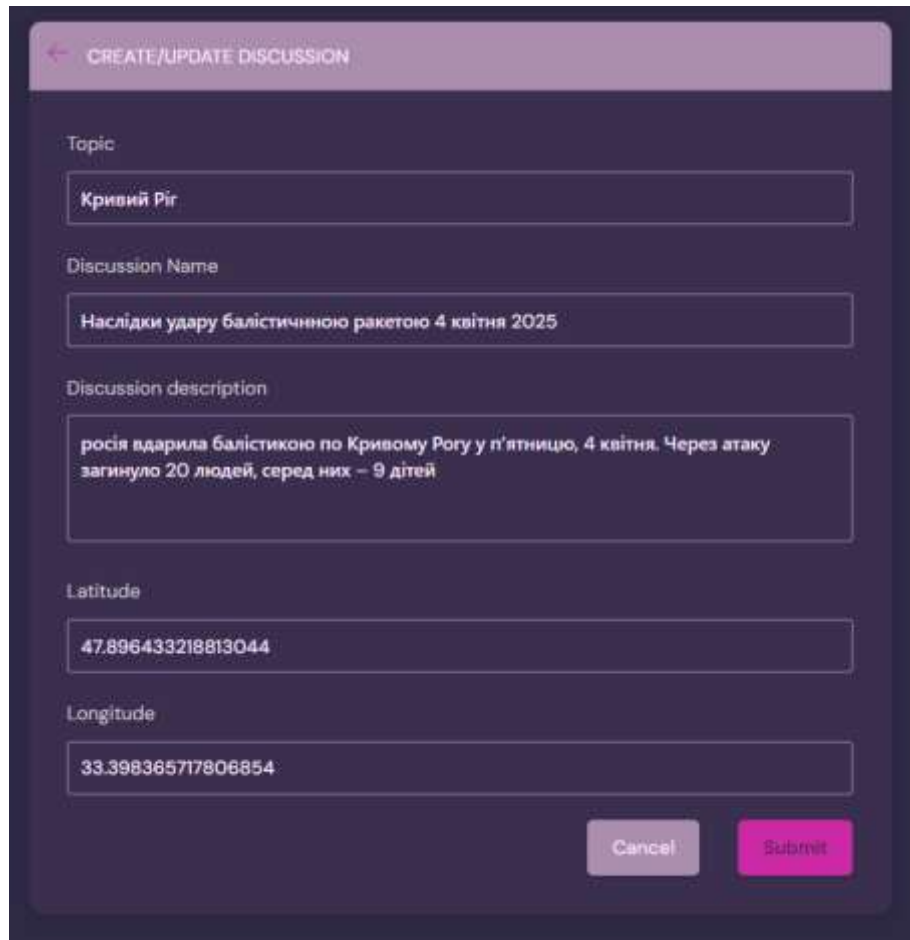


Рисунок 4. 4 - карта з мітками

Після вибору місця користувачеві надається формочка (див. Рисунок 4.5) для позначення загальної теми до якої можна занести цю подію, також потрібно заповнити назви обговорення, та опис за бажанням.



The screenshot shows a mobile application interface for creating or updating a discussion. The form is titled "CREATE/UPDATE DISCUSSION" and includes the following fields:

- Topic:** Кривий Ріг
- Discussion Name:** Наслідки удару балістичною ракетою 4 квітня 2025
- Discussion description:** росія вдарила балістикою по Кривому Рогу у п'ятницю, 4 квітня. Через атаку загинуло 20 людей, серед них – 9 дітей
- Latitude:** 47.896433218813044
- Longitude:** 33.398365717806854

At the bottom right of the form, there are two buttons: "Cancel" and "Submit".

Рисунок 4. 5 - Форма створення обговорення

Створюється окрема кімната (див. Рисунок 4.6), в якій відвідувачі можуть обмінюватися думками та ділитися своїм досвідом. В обговоренні може брати участь кожна авторизована людина. Загалом кімната складається з таких компонентів: її опису, хто створив обговорення, чатом для обміну інформацією та панеллю учасників, що активні. Користувач, який створив обговорення, може змінювати його опис, для того щоб важлива інформація була закріплена зверху (див. Рисунок 4.5).

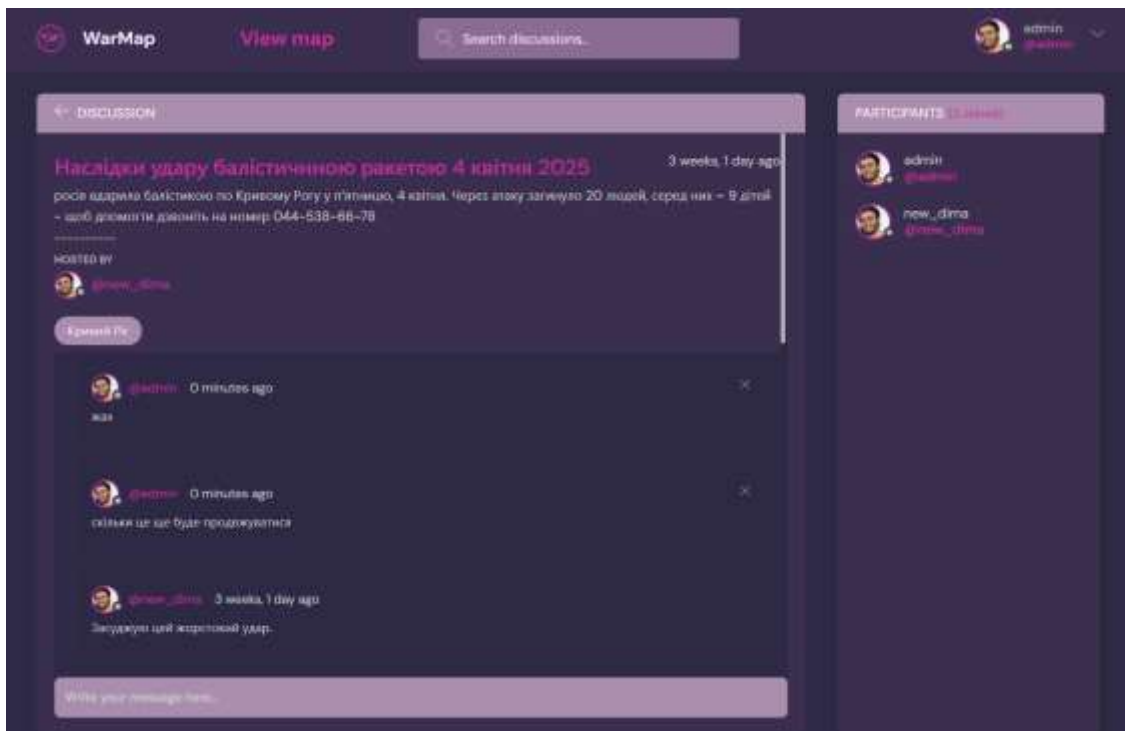


Рисунок 4. 6 - Кімната для обговорення

Зайшовши на карту, є можливість переглянути місцезнаходження (див. Рисунок 4.7), що задіяні в обговореннях; для цього потрібно клікнути на мітку, яка зацікавила, і буде показано повідомлення щодо цієї події.

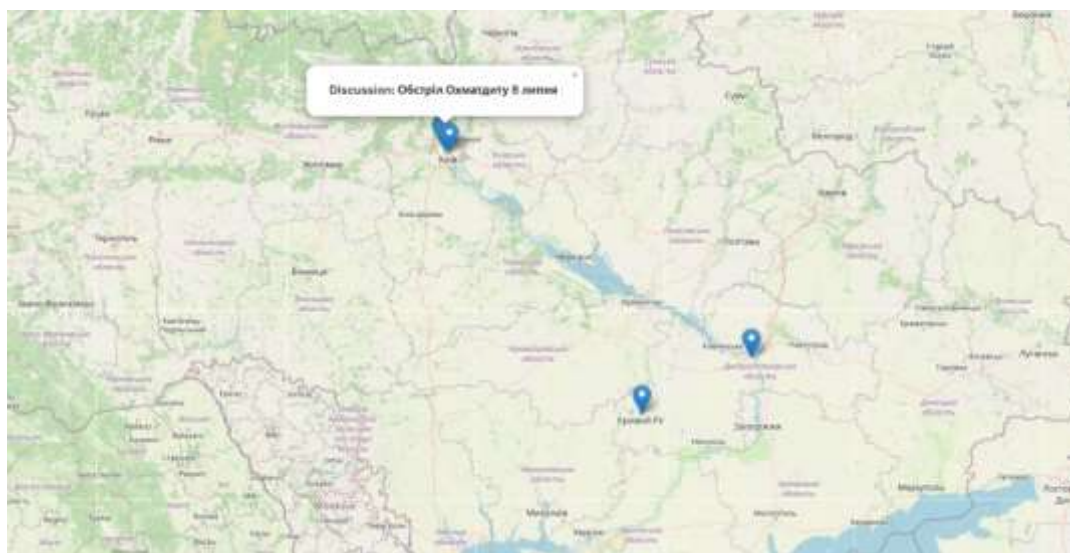


Рисунок 4. 7 - Перегляд назви події

Якщо користувач зробив чат-кімнату та відніс її до такої до загальної теми, яка вже попередньо була створена – то кімната автоматично додається

в цей розділ. Користувацький інтерфейс є зручним та інтуїтивним, адже все знаходиться на одній сторінці (див. Рисунок 4.8). До того ж реалізовано зручний пошук кімнат за ключовим словом.

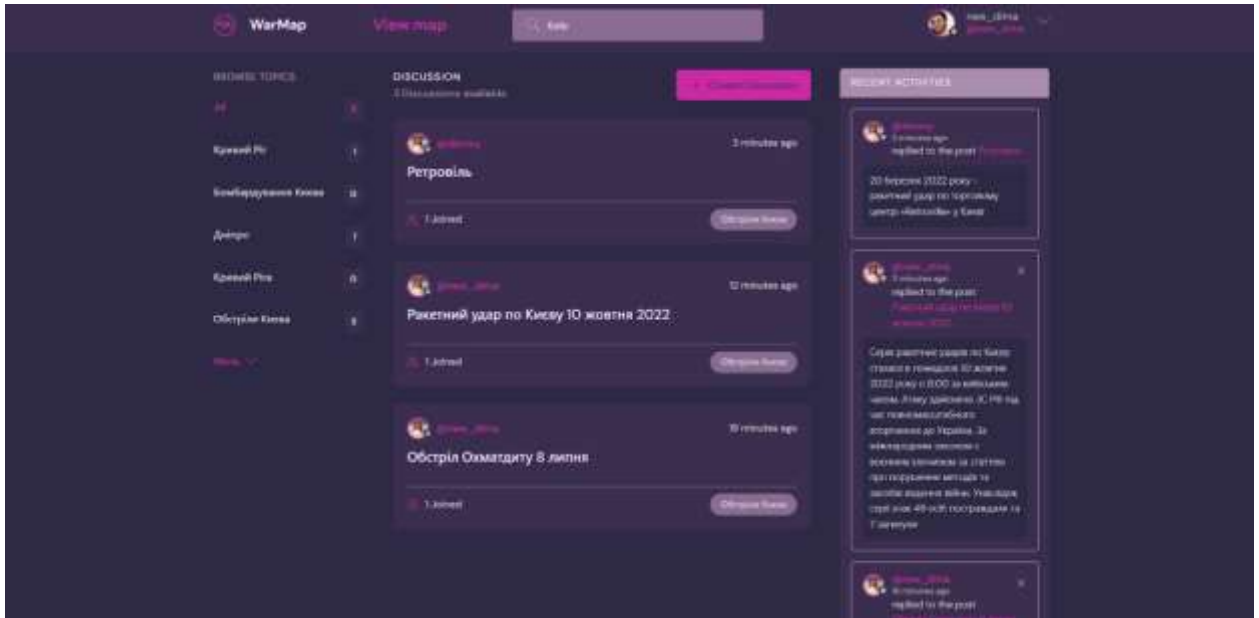


Рисунок 4. 8 - Інтерфейс платформи

Натиснувши на ім'я користувача, можна безпосередньо перейти на його профіль (див. Рисунок 4.9) та побачити інформацію про нього, чи він верифікований, останню активність, в яких обговореннях брав участь та кімнати, що створені безпосередньо цією людиною.

Загалом, інтерфейс платформи розроблено так, щоб він був інтуїтивно-зрозумілий, зручний та візуально привабливий для користувачів. Реалізовано легкий пошук по будь-яким словам, наприклад увівши назву міста можна подивитися теми обговорень щодо нього. Також ліворуч на веб-платформі розроблена панель швидкого доступу до наявних обговорень, що надає учасникові зручності в пошуку по веб-сайту.

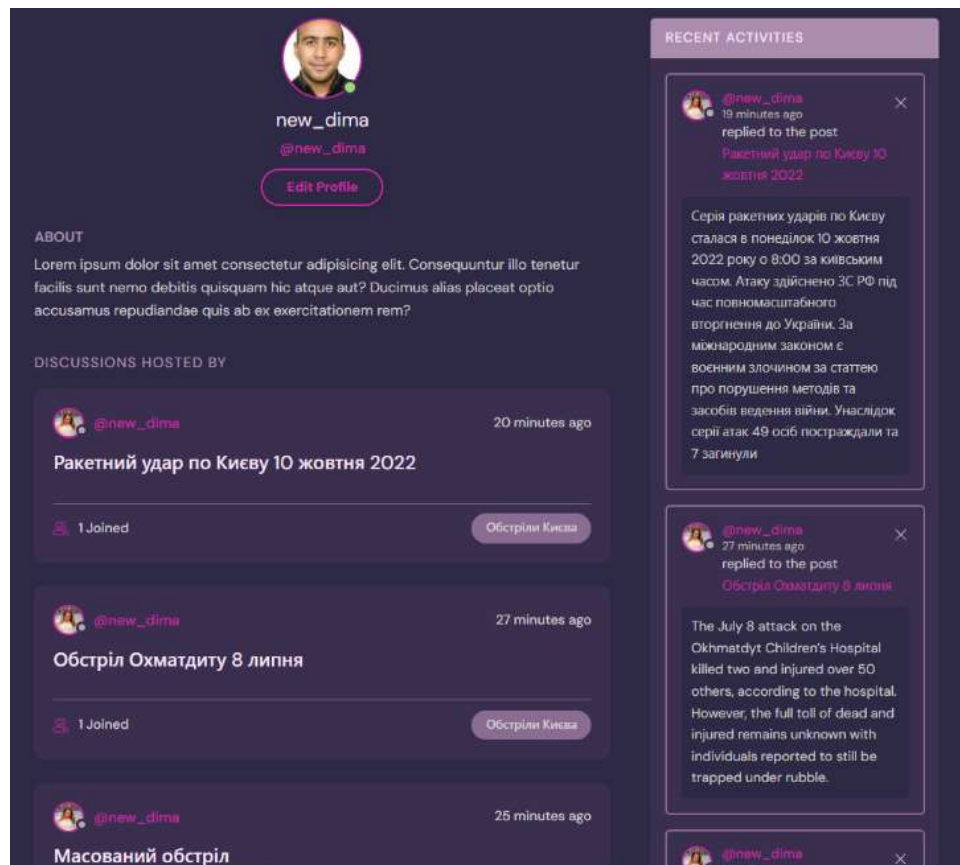


Рисунок 4. 9 - Профіль користувача

4.2 Порівняння методу з іншими

В цьому розділі описується порівняння розробленого методу щодо інших краудсорсингових платформ, як от HealthMap, DeepStateUA, Liveuamap і Bellingcat. Вони всі представляють різні підходи до верифікації та поширення даних, а запропонована система висвітлює нові можливості для забезпечення цілісності.

Вебсервіс HealthMap, що спеціалізується на спостереженні різних спалахів хвороб, збирає дані з різних ресурсів, наприклад новин, повідомлень від свідків, соціальних мереж тощо. Поєднуються різні джерела даних, як і формальні, так і неформальні, також застосовується машинне навчання для категоризації та фільтрування повідомлень про захворювання. Певною мірою це схоже на наш метод, однак верифікація може зайняти велику кількість

часу, і тому мінімальний спалах хвороби може швидко поширюватися. В контексті запропонованого методу, хоч він і не пов'язаний зі здоров'ям, перевірка інформації здійснюється швидше, адже використовується велика мовна модель та інші методи, що забезпечують правдивість інформації.

Українська платформа для розвідки та відображення подій в реальному часі DeepStateMap використовує текстові дані від користувачів, військових, різних каналів з месенджерів, а ще із супутникових знімків. За допомогою всього цього веб-застосунок надає деталізовану карту динаміки перебігу подій на полі бою. Ця платформа є кращою в активному залученні спільноти та має часті оновлення. Візуальний інтерфейс дуже гарно та зрозуміло представлений. Але за відсутності прямих доказів можуть бути помилки та непорозуміння. Проте на нашій платформі оброблюються саме маніпулятивні тексти, а не верифікація подій реального часу, тому помилок може бути менше. На відміну від DeepStateMap, є верифікація через Дію.

Liveuamap – це онлайн-застосунок для новин певної країни, що відображаються на карті. Здебільшого інформація береться з соціальних мереж, тому висока ймовірність потрапити на неперевірені ЗМІ, або недостовірну інформацію. Тут непередбачено інтерактивної взаємодії з користувачами, тому і верифікації даних від них не передбачається. Платформа є корисною в плані географічного охоплення та швидкого оновлення різних новин. Візуалізація представлена у вигляді карти зі зручним розташування новин та зрозумілим і легким інтерфейсом.

Bellingcat – це платформа, що поєднує різні колективні розслідування та методики журналістики з використанням OSINT-методології для розкриття воєнних злочинів, порушення прав людини та дезінформації. Зі всіх вищеописаних застосунків, цей є найбільш правдивим з огляду на інформацію, що на ньому розташована, адже вебсервіс не є краудсорсинговим щодо прямих внесків самими користувачами. Вони не можуть нічого зробити, щоб змінити правдивість виставленої новини, тому що до представлення події проводиться ретельна перевірка та

представляються усі протоколи документування тих чи інших рішень. Здійснюються глибокі розслідування на основі різних типів даних: метаданих, зображень, відео, публікацій тощо. Недолік цього підходу в тому, що є висока залежність від експертів-аналітиків. На відміну від Bellingcat, запропонована платформа є більш демократичною, адже кожен може поділитися своєю частиною історії.

5 Висновки

5.1 Аналіз результатів

Під час роботи над курсовим проєктом було проведено дослідження в якому був розроблений підхід, щоб зменшити поширення емоційно-забарвленої інформації на краудсорсинговій платформі в контексті російсько-української війни. Результатом стала платформа, що поєднує автоматизовану попередню перевірку з людською модерацією та верифікацією користувачів, задля прозорості та довіри до системи. Експериментальний веб-застосунок реалізований за допомогою Python з використанням Django REST Framework. Для клієнтської частини були використані технології HTML, CSS і JavaScript, а Leaflet – для інтерактивної картографічної функціональності, що дозволяє користувачам додавати події в тому місці, де вона сталася. Збереження даних та логіка програми керувалися за допомогою ORM Django разом з SQLite.

Оскільки платформа має такий формат як краудсорсинг, це сприяє розповсюдженню маніпулятивної інформації, що може стати величезною проблемою в збереженні історії подій російсько-української війни. З метою підвищення рівня забезпеченості достовірності наданих даних було впроваджено можливість верифікації за допомогою державного сервісу Дія. Це створює базовий рівень довіри для громадян. Крім того, щоб збільшити рівень довіри, було інтегровано у веб-систему мовну модель DistilBERT з подальшою можливістю аналізу позначених маніпулятивних повідомлень за допомогою адміністраторів. Для позначення потенційно шкідливого або емоційно-забарвленого контенту ця модель пропонує найкращий баланс між ефективністю, швидкістю та точністю.

Хоча жодна система не може повністю усунути маніпуляції, такий багаторівневий підхід значно зменшує їхній вплив і створює основу для

подальшого вдосконалення верифікації контенту та управління платформами.

5.2 Напрямки подальшого дослідження

Модель, що використовується, може мати погану точність при використанні сленгу, сатир чи різних військових тем, тому треба підвищувати точність за допомогою активного навчання та зменшувати кількість помилкових дій. До того ж, можна збільшити фокус моделі, наприклад, окрім аналізу конкретного повідомлення на емоційний тон, також виявляти цілі маніпулятивні обговорення. Варто зазначити, що існує ризик несправедливої цензури, якщо модель позначає правдиві, але емоційні повідомлення, особливо в такій чутливій темі, як війна. Тому людська модерація залишається важливою.

Важливо пам'ятати про проблему промпт-ін'єкцій, тому краще обрати модель, яка вже налаштована деякими механізмами щодо стійкості до ін'єкцій. Крім того, необхідно попередньо оброблювати усі вхідні дані для виявлення підозрілих шаблонів (промптів), як от: “Ігноруй це”, “Як штучний інтелект”, “Забути попередні інструкції” тощо. Далі потрібно відфільтрувати дані перед тим, як вони потраплять в модель. Можна використовувати фільтри на основі регулярних виразів або семантичні фільтри для виявлення поширених промптів чи зловмисних фраз. До того ж, потрібно регулярно проводити тестування системи, використовувати різноманітні приклади шаблонів для налаштування або покращення моделі.

Інтеграція Дії виокремлює лише одну частину людей, що може призвести до упередженості, тому треба впровадити верифікацію чи підтвердження для людей, що надають цінну інформацію чи беруть активну участь в підтримці України, але не є громадянами нашої країни, наприклад іноземні журналісти, волонтери тощо.

Враховувати спам та впровадити надійний захист системи, адже вона пов'язана зі збереженням історії російсько-українською війни. Підтримувати декілька іноземних мов задля більшого охоплення та поширення. Та продумати логіку подальшого розвитку платформи залежно від бізнес-завдання.

При масштабуванні платформи потрібно здійснити перехід з SQLite на більш потужну інфраструктуру чи реляційні системи. Зокрема, необхідно розширити функціональність користувацьких профілів з різними можливостями редагування. Додатково можна впровадити функції додавання різного мультимедійного контенту задля підвищення інформативності, але як наслідок, слід підвищити перевірку для аналізу фото та відео.

Важливо продумати як вдосконалити візуалізацію міток на карті, бо кімнати створені користувачами теж можуть бути маніпулятивними, тому треба розробити кольорове маркування або інший спосіб відокремлення неперевіраних модераторами подій. Крім цього, можна поєднати створення кімнати з головної сторінки платформи так, щоб мітка з'являлася на карті автоматично після підтвердження форми. Таким чином, дослідження має значний потенціал для розвитку, покращення верифікації та підвищення довіри до розробленої краудсорсингової платформи.

Джерела

- [1] Г. «Детектор «медіа», «Індекс медіаграмотності українців 2020–2023 (четверта хвиля)», detector.media. Дата звернення: 02, Березень 2025. [Online]. Доступний у: <https://detector.media/infospace/article/225738/2024-04-22-indeks-mediagramotnosti-ukraintsv-20202023-chetverta-khvylya/>
- [2] J. Wang, Z. Zhu, C. Liu, R. Li, і X. Wu, «LLM-Enhanced multimodal detection of fake news», *PLOS ONE*, вип. 19, вип. 10, с. e0312240, Жов 2024, doi: 10.1371/journal.pone.0312240.
- [3] R. Kitchin і G. McArdle, «What makes Big Data, Big Data? Exploring the ontological characteristics of 26 datasets», *Big Data Soc.*, вип. 3, вип. 1, с. 2053951716631130, Чер 2016, doi: 10.1177/2053951716631130.
- [4] A. P. Reimer і E. A. Madigan, «Veracity in big data: How good is good enough», *Health Informatics J.*, вип. 25, вип. 4, с. 1290–1298, Груд 2019, doi: 10.1177/1460458217744369.
- [5] E. Papageorgiou, C. Chronis, I. Varlamis, і Y. Himeur, «A Survey on the Use of Large Language Models (LLMs) in Fake News», *Future Internet*, вип. 16, вип. 8, Art. вип. 8, Сер 2024, doi: 10.3390/fi16080298.
- [6] A. Pico, E. Vivancos, A. Garcia-Fornes, і V. Botti, «Exploring Text-Generating Large Language Models (LLMs) for Emotion Recognition in Affective Intelligent Agents»: в *Proceedings of the 16th International Conference on Agents and Artificial Intelligence*, Rome, Italy: SCITEPRESS - Science and Technology Publications, 2024, с. 491–498. doi: 10.5220/0012596800003636.
- [7] J. S. Huffaker, J. K. Kummerfeld, W. S. Lasecki, і M. S. Ackerman, «Crowdsourced Detection of Emotionally Manipulative Language», в *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, в CHI '20. New York, NY, USA: Association for Computing Machinery, Квіт 2020, с. 1–14. doi: 10.1145/3313831.3376375.
- [8] «Addressing Hoaxes and Fake News», Meta. Дата звернення: 08, Березень 2025. [Online]. Доступний у: <https://about.fb.com/news/2016/12/news-feed-fyi-addressing-hoaxes-and-fake-news/>
- [9] «Crowdmapping», *Wikipedia*. 12, Вересень 2024. Дата звернення: 09, Березень 2025. [Online]. Доступний у: <https://en.wikipedia.org/w/index.php?title=Crowdmapping&oldid=1245396386>
- [10] «HealthMap», *Wikipedia*. 18, Липень 2024. Дата звернення: 09, Березень 2025. [Online]. Доступний у: <https://en.wikipedia.org/w/index.php?title=HealthMap&oldid=1235220506>
- [11] «Хто такі DeepState, як вони добувають інформацію і чи мають зв'язок з владою. Інтерв'ю BBC із засновниками OSINT-проекту», *New Voice*. Дата звернення: 09, Березень 2025. [Online]. Доступний у:

- <https://nv.ua/ukr/ukraine/events/deepstate-hto-ce-taki-zvidki-berut-informaciyu-dlya-kart-i-chi-pov-yazani-z-minoboroni-ukrajini-50446305.html>
- [12] Ю. Лавришин, «Мапа, до якої ‘немає питань’. Як працює проєкт DeepState», ms.detector.media. Дата звернення: 09, Березень 2025. [Online]. Доступний у: <https://ms.detector.media/it-kompanii/post/32559/2023-07-31-mapa-do-yakoi-nemaie-pytan-yak-pratsyuie-proiekt-deepstate/>
- [13] «Django documentation | Django documentation», Django Project. Дата звернення: 15, Березень 2025. [Online]. Доступний у: <https://docs.djangoproject.com/en/5.1/>
- [14] W. Wizard, «Understanding Django’s MVT Architecture: A Beginner’s Guide to Models, Views, and Templates | Web Wizard». Дата звернення: 15, Березень 2025. [Online]. Доступний у: <https://www.webwizard.ie/blog/understanding-djangos-mvt-architecture-a-beginners-guide-to-models-views-and-templates/>
- [15] «Welcome to Flask — Flask Documentation (3.1.x)». Дата звернення: 15, Березень 2025. [Online]. Доступний у: <https://flask.palletsprojects.com/en/stable/>
- [16] «What Is Express JS? Features, Uses, and Limitations». Дата звернення: 15, Березень 2025. [Online]. Доступний у: <https://www.digitalregenesys.com/blog/what-is-express-js>
- [17] «Spring Boot», Spring Boot. Дата звернення: 15, Березень 2025. [Online]. Доступний у: <https://spring.io/projects/spring-boot>
- [18] «Installation - Laravel 12.x - The PHP Framework For Web Artisans». Дата звернення: 15, Березень 2025. [Online]. Доступний у: <https://laravel.com/>
- [19] «The Ultimate Guide to Django Templates | The PyCharm Blog», The JetBrains Blog. Дата звернення: 21, Березень 2025. [Online]. Доступний у: <https://blog.jetbrains.com/pycharm/2025/02/the-ultimate-guide-to-django-templates/>
- [20] «What Is a Prompt Injection Attack? | IBM». Дата звернення: 22, Березень 2025. [Online]. Доступний у: <https://www.ibm.com/think/topics/prompt-injection>
- [21] Owasplmp. Admin, «LLM01:2025 Prompt Injection», OWASP Top 10 for LLM & Generative AI Security. Дата звернення: 22, Березень 2025. [Online]. Доступний у: <https://genai.owasp.org/llmrisk/llm01-prompt-injection/>