Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА

на тему *Метод інтерпретування результатів виявлення фейкових новин за великою мовною моделлю*

Рівень вищої освіти *другий (магістерський)*

Галузь знань *12 – Інформаційні технології*
Шифр і найменування

Спеціальність *122 – Комп'ютерні науки*
Код і найменування

Освітня програма *Комп'ютерні науки*
Назва

Виконала: *студентка 2 курсу, група КНм-24-1* (Підпис) *Стефанія ВОВК*
Курс, група виконавця — Ім'я, ПРІЗВИЩЕ

Керівник: *док. філ., доцент кафедри КН* (Підпис) *Павло РАДЮК*
Науковий ступінь, посада — Ім'я, ПРІЗВИЩЕ

Нормоконтроль: *к.т.н., доцент кафедри КН* (Підпис) *Руслан БАГРІЙ*
Науковий ступінь, посада — Ім'я, ПРІЗВИЩЕ

До захисту допускаю:
Зав. кафедри КН, д.т.н., професор (Підпис) *Олександр Бармак*
Ім'я, ПРІЗВИЩЕ

10 грудня 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Факультет *інформаційних технологій*
Кафедра *комп'ютерних наук*
Освітній ступінь *магістр*
Галузь знань *12 – Інформаційні технології*
Спеціальність *122 – Комп'ютерні науки*

ЗАТВЕРДЖУЮ
Завідувач кафедри *комп'ютерних наук*

_____
(підпис)
д.т.н., професор Олександр БАРМАК

_____
«*28*» *08* *2025* року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

1. Тема кваліфікаційної роботи: *«Метод інтерпретування результатів виявлення фейкових новин за великою мовною моделлю»*

2. Завдання видано студенту *Стефанії ВОВК*
(Ім'я, ПРІЗВИЩЕ)

3. Керівник роботи *док. філ., доцент кафедри КН Павло РАДЮК*
(Ім'я, ПРІЗВИЩЕ)

4. Затверджені наказом університету від «*25*» *08* *2025* р. № *65*

5. Дата видачі завдання студенту: «*28*» *08* *2025* р.

6. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Метою кваліфікаційної роботи магістра є підвищення інтерпретованості та точності виявлення фейкових новин через створення нового методу на основі великих мовних моделей. Зміст пояснювальної записки охоплює аналіз методів прозорості LLM, проєктування алгоритмів генерації та візуалізації векторних представлень, а також програмну реалізацію інтерактивної системи з використанням підходу «людина-у-петлі». Вхідними даними слугували текстові масиви новин з еталонних наборів LIAR та GossipCop. За результатами експериментальних досліджень підтверджено результативність впровадження запропонованого методу, що дало змогу підвищити точність класифікації фейкових новин та забезпечити зрозуміле пояснення рішень моделі.

7. Календарний план виконання кваліфікаційної роботи:

№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	*Вибір напряму дослідження та узгодження теми кваліфікаційної роботи з керівником, складання календарного графіка виконання роботи*	*вересень 2025*	*Виконано*
2	*Ознайомлення з предметною областю, аналіз існуючих методів і моделей, формулювання мети та завдань дослідження, визначення об'єкта й предмета дослідження*	*вересень 2025*	*Виконано*
3	*Проєктування методу для розв'язання обраного завдання, опис архітектури рішення*	*жовтень 2025*	*Виконано*
4	*Програмна реалізація запропонованого методу*	*жовтень 2025*	*Виконано*
5	*Експериментальна перевірка одержаних результатів, порівняння з відомими підходами*	*листопад 2025*	*Виконано*
6	*Написання пояснювальної записки, оформлення відповідно до вимог, врахування зауважень керівника*	*листопад 2025*	*Виконано*
7	*Підготовка презентаційних матеріалів та попередній захист*	*листопад 2025*	*Виконано*
8	*Перевірка пояснювальної записки на відповідність вимогам оформлення (нормоконтроль) та перевірка на академічну доброчесність. Отримання відгуку керівника та рецензії.*	*грудень 2025*	*Виконано*
9	*Публічний захист кваліфікаційної роботи*	*грудень 2025*	*Виконано*

Виконавець: *студент групи КНм-24-1* _____ *Стефанія ВОВК*
Група виконавця | Підпис | Ім'я, ПРІЗВИЩЕ

Керівник: *док. філ., доц. каф. КН* _____ *Павло РАДЮК*
Науковий ступінь, посада | Підпис | Ім'я, ПРІЗВИЩЕ

Реферат

Кваліфікаційна робота магістра спрямована на підвищення рівня інтерпретованості та точності систем виявлення фейкових новин через створення методу інтерпретування результатів виявлення фейкових новин із використанням великих мовних моделей та технологій пояснюваного штучного інтелекту.

Актуальність теми. У сучасних умовах інформаційних атак та стрімкого поширення дезінформації автоматизовані системи виявлення фейків стають необхідним інструментом медіаграмотності та інформаційної безпеки. Сучасні великі мовні моделі (LLM) демонструють високу ефективність у задачах класифікації текстів, однак часто функціонують як “чорні скриньки”, що ускладнює розуміння логіки прийняття ними рішень. Брак прозорості знижує довіру користувачів до результатів роботи таких систем. Тому актуальним завданням є створення методів, які не лише точно класифікують новини на фейковість, а й надають зрозумілі пояснення (інтерпретації) власних рішень, а також дають змогу експертам впливати на процес навчання моделі для підвищення її точності.

Об’єкт дослідження – процес автоматизованого виявлення та аналізу фейкових новин у текстових даних.

Предмет дослідження – моделі та методи пояснюваного штучного інтелекту, алгоритми глибокого навчання та методи візуалізації векторних представлень текстів (ембедінгів) для інтерпретації результатів класифікації.

Мета роботи – підвищення рівня інтерпретованості та точності систем виявлення фейкових новин через проєктування методу, який забезпечує інтерактивний аналіз, валідування та ітеративне вдосконалення простору текстових ембедінгів із залученням експерта.

Для досягнення поставленої мети необхідно розв’язати такі **завдання**:

1. Провести аналіз методів забезпечення прозорості LLM, підходів до візуалізації текстових ембедінгів та наявних програмних рішень для виявлення дезінформації в текстових даних.

2. Спроєктувати метод інтерпретування результатів виявлення фейкових новин, що полягає в генеруванні високоінформативних текстових ембедінгів для новинних статей із використанням архітектур великих мовних моделей.

3. Розробити підсистему зниження розмірності та візуалізації багатовимірних ембедінгів у двовимірному просторі за спроєктованим методом.

4. Розробити архітектуру інтерактивної системи та виконати її програмну реалізацію, що надає інструменти для дослідження проєкцій, пояснення рішень та реалізує підхід “людина-у-петлі” для виявлення фейкових новин.

5. Провести експериментальне дослідження спроєктованого методу та його програмної реалізації за еталонними наборами даних та оцінити результативність запропонованих підходів.

Методи дослідження. У роботі використано методи машинного навчання (Logistic Regression, Random Forest) та глибокого навчання (Transformer-моделі) для класифікації текстів; методи пояснюваного штучного інтелекту (XAI, SHAP, LIME, Integrated Gradients) для інтерпретування результатів класифікації; методи зниження розмірності (t-SNE, UMAP) для візуалізації даних; об’єктно-орієнтований підхід для проєктування програмної системи.

Наукова новизна одержаних результатів. Удосконалено метод інтерпретування результатів виявлення фейкових новин унаслідок інтеграції великих мовних моделей із модулями пояснюваного штучного інтелекту та концепцією “людина-у-петлі”, що відрізняється від наявних рішень поєднанням послідовного оброблення тексту (трансформери), глибокого семантичного аналізу, візуалізації простору ознак та наданням можливості експерту інтерактивно впливати на параметри моделі та навчальну вибірку, що дало змогу підвищити точність класифікації на 2–4 % та забезпечити прозорість прийняття рішень.

Апробація результатів кваліфікаційної роботи магістра та публікації. Основні наукові та практичні результати пройшли апробацію на XVII Всеукраїнській науково-практичній конференції «Актуальні проблеми комп’ютерних наук (АПКН – 2025)» (м. Хмельницький, 14–15 листопада 2025 р.)

та опубліковані у фаховому виданні «Вісник Хмельницького національного університету. Технічні науки»:

1. Вовк С. В., Радюк П. М., Скрипник Т. К. Метод інтерпретування результатів виявлення фейкових новин за великою мовною моделлю. Актуальні проблеми комп'ютерних наук АПКН-2025 : матеріали XVII Всеукр. науково-практ. конф., м. Хмельницький, 14–15 листопада 2025 р. Хмельницький, 2025. С. 68–71. URL: https://elar.khmnu.edu.ua/handle/123456789/19862 (дата звернення: 03.12.2025)

2. Вовк С. В., Радюк П. М., Скрипник Т. К. Метод інтерпретування результатів виявлення фейкових новин за великою мовною моделлю. *Вісник Хмельницького національного університету. Технічні науки*. 2025. Т. 359, № 6(2). (Довідка з редакції).

Структура та обсяг роботи. Кваліфікаційна робота магістра складається із завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань з 52 найменувань та 5 додатків. Загальний обсяг кваліфікаційної роботи складає 123 сторінок, з поміж яких 94 сторінок основного тексту та 29 сторінок додатків. У роботі наведено 28 рисунків та 15 таблиць.

Ключові слова: фейкові новини, велика мовна модель, інтерпретованість, XAI, ембедінги, візуалізація даних, класифікація текстів, людина-у-петлі.

Зміст

Перелік скорочень

Скорочення, термін, позначення	Пояснення
БД	База даних
КН	Комп’ютерні науки
СКБД	Система керування базами даних
TF-IDF	Term Frequency – Inverse Document Frequency
LIME	Local Interpretable Model-agnostic Explanations
XAI	Explainable Artificial Intelligence
t-SNE	t-Distributed Stochastic Neighbor Embedding
UMAP	Uniform Manifold Approximation and Projection
ШІ	Штучний інтелект
IG	Integrated Gradients
SHAP	SHapley Additive exPlanations
LLM	Large Language Model
BERT	Bidirectional Encoder Representations from Transformers
REST	Representational State Transfer
gRPC	Google Remote Procedure Call
API	Application Programming Interface
HTTP	Hypertext Transfer Protocol
WSL	Windows Subsystem for Linux
NAM	Neural Additive Models
UI	User Interface
PCA	Principal Component Analysis
NLP	Natural Language Processing
GPT	Generative Pre-trained Transformer
LSTM	Long Short-Term Memory

Вступ

Актуальність теми. У сучасних умовах інформаційних атак та стрімкого поширення дезінформації автоматизовані системи виявлення фейків стають необхідним інструментом медіаграмотності та інформаційної безпеки. Сучасні великі мовні моделі (LLM) демонструють високу ефективність у задачах класифікації текстів, однак часто функціонують як “чорні скриньки”, що ускладнює розуміння логіки прийняття ними рішень. Брак прозорості знижує довіру користувачів до результатів роботи таких систем. Тому актуальним завданням є створення методів, які не лише точно класифікують новини на фейковість, а й надають зрозумілі пояснення (інтерпретації) власних рішень, а також дають змогу експертам впливати на процес навчання моделі для підвищення її точності.

Об’єкт дослідження – процес автоматизованого виявлення та аналізу фейкових новин у текстових даних.

Предмет дослідження – моделі та методи пояснюваного штучного інтелекту, алгоритми глибокого навчання та методи візуалізації векторних представлень текстів (ембедінгів) для інтерпретації результатів класифікації.

Мета роботи – підвищення рівня інтерпретованості та точності систем виявлення фейкових новин через проєктування методу, який забезпечує інтерактивний аналіз, валідування та ітеративне вдосконалення простору текстових ембедінгів із залученням експерта.

Для досягнення поставленої мети необхідно розв’язати такі **завдання**:

1. Провести аналіз методів забезпечення прозорості LLM, підходів до візуалізації текстових ембедінгів та наявних програмних рішень для виявлення дезінформації в текстових даних.

2. Спроєктувати метод інтерпретування результатів виявлення фейкових новин, що полягає в генеруванні високоінформативних текстових ембедінгів для новинних статей із використанням архітектур великих мовних моделей.

3. Розробити підсистему зниження розмірності та візуалізації багатовимірних ембедінгів у двовимірному просторі за спроєктованим методом.

4. Розробити архітектуру інтерактивної системи та виконати її програмну реалізацію, що надає інструменти для дослідження проєкцій, пояснення рішень та реалізує підхід “людина-у-петлі” для виявлення фейкових новин.

5. Провести експериментальне дослідження спроєктованого методу та його програмної реалізації за еталонними наборами даних та оцінити результативність запропонованих підходів.

Методи дослідження. У роботі використано методи машинного навчання (Logistic Regression, Random Forest) та глибокого навчання (Transformer-моделі) для класифікації текстів; методи пояснюваного штучного інтелекту (XAI, SHAP, LIME, Integrated Gradients) для інтерпретування результатів класифікації; методи зниження розмірності (t-SNE, UMAP) для візуалізації даних; об’єктно-орієнтований підхід для проєктування програмної системи.

Наукова новизна одержаних результатів. Удосконалено метод інтерпретування результатів виявлення фейкових новин унаслідок інтеграції великих мовних моделей із модулями пояснюваного штучного інтелекту та концепцією “людина-у-петлі”, що відрізняється від наявних рішень поєднанням послідовного оброблення тексту (трансформери), глибокого семантичного аналізу, візуалізації простору ознак та наданням можливості експерту інтерактивно впливати на параметри моделі та навчальну вибірку, що дало змогу підвищити точність класифікації на 2–4 % та забезпечити прозорість прийняття рішень.

Апробація результатів кваліфікаційної роботи магістра та публікації. Основні наукові та практичні результати пройшли апробацію на XVII Всеукраїнській науково-практичній конференції «Актуальні проблеми комп’ютерних наук (АПКН – 2025)» (м. Хмельницький, 14–15 листопада 2025 р.) [1] та опубліковані у фаховому виданні «Вісник Хмельницького національного університету. Технічні науки» (2025, Т. 359, № 6) [2].

Структура та обсяг роботи. Кваліфікаційна робота магістра складається із завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань з 52 найменувань та 5 додатків. Загальний обсяг кваліфікаційної роботи складає 123 сторінок, з поміж яких 94 сторінок основного тексту та 29 сторінок додатків. У роботі наведено 28 рисунків та 15 таблиць.

РОЗДІЛ 1 Аналіз проблем та рішень у сфері інтерпретації великих мовних моделей у задачах виявлення фейкових новин

1.1 Аналіз методів забезпечення прозорості та довіри в сучасних великих мовних моделях

У сучасному світі ШІ швидко стає ключовою технологією, що змінює підходи до обробки інформації, прийняття рішень та автоматизації процесів. Його застосування охоплює такі сфери, як медицина, фінанси, освіта, юриспруденція, оборонна промисловість та засоби масової інформації. Особливо помітним є вплив LLM, які здатні генерувати тексти, вести діалоги, перекладати мови та здійснювати складні аналітичні завдання [3]. Однак зростання можливостей ШІ супроводжується новими викликами, поміж яких головне місце займає прозорість та довіра до результатів його роботи [4]. Більшість сучасних моделей, зокрема глибокі нейронні мережі, функціонують як «чорні скриньки», ускладнюючи для користувача розуміння того, чому модель прийняла саме таке рішення.

У відповідь на ці виклики виникає окремий напрям досліджень – пояснюваний штучний інтелект (Explainable AI або XAI), головною метою якого є підвищення інтерпретованості, прозорості та довіри до рішень, що приймаються ШІ-системами. Це особливо актуально у випадку з LLM, які іноді можуть генерувати так звані «галюцинації» – тексти або твердження, що виглядають правдоподібно, але насправді є некоректними, хибними або вигаданими. Така поведінка становить суттєву загрозу в контексті використання ШІ у сферах, де на кону – безпека, етика або суспільна довіра [5].

XAI надає інструменти, що дають змогу користувачам і розробникам краще розуміти, як і чому модель дійшла до певного висновку, даючи змогу виявити потенційні помилки, викривлення чи упередження в її роботі. Це, своєю чергою, сприяє формуванню більш відповідального використання ШІ, забезпеченню контролю над його рішеннями та запобіганню ризикам, що виникають при автоматизованій обробці інформації в критичних галузях – таких як медицина, право, освіта чи державне управління [6].

З огляду на ці виклики, дедалі актуальнішою стає потреба в застосуванні сучасних методів XAI, які дають змогу розкрити внутрішню логіку роботи LLM і зробити процес ухвалення ними рішень більш прозорим і контрольованим. У сфері XAI наразі домінують два головні підходи до досягнення інтерпретованості: post-hoc пояснення та інтерпретованість-за-дизайном. Поміж поширених post-hoc технік можна відзначити як локальні, так і глобальні підходи до інтерпретації, візуалізацію релевантних ознак, застосування моделей-замінників спрощеної структури, а також методи, що використовують аналіз градієнтів і механізмів уваги.

Локальна інтерпретація або LIME [7] є методом, призначеним для пояснення рішень моделей класифікації або регресії шляхом побудови інтерпретованої моделі, яка апроксимує поведінку основної моделі у вузькому околі конкретного прикладу. Іншими словами LIME зосереджується на інтерпретації прогнозу для окремого об’єкта, а не на побудові загального уявлення про всю модель [8].

Процес пояснення за допомогою методу LIME включає послідовність взаємопов’язаних етапів. Спочатку обирається конкретний приклад x для якого необхідно отримати пояснення. Після цього навколо нього формується локальна вибірка Z шляхом створення випадкових модифікацій x (наприклад, зміна окремих ознак або слів у тексті). Для кожного елемента $z \in Z$ обчислюються передбачення складної моделі $f(z)$, що дає змогу зафіксувати її поведінку у найближчому до x оточенні. Далі на основі отриманих пар $(z, f(z))$ навчається спрощена, інтерпретована модель g, яка мінімізує локальну втрату L, враховуючи ваги $\pi_x(z)$, що визначають ступінь близькості кожного z до x і, відповідно, фокусують апроксимацію на найближчих точках. Завершальним кроком є аналіз коефіцієнтів моделі g, які показують важливість окремих ознак для прогнозу $f(x)$.

Хоча LIME є потужним інструментом для локального пояснення та дає змогу краще розуміти причини прийняття конкретних рішень моделлю, він має певні обмеження. Зокрема, цей метод погано масштабується для моделей із мільярдами параметрів, оскільки генерація локальної вибірки й побудова

адекватної апроксимації в таких високовимірних просторах стає надзвичайно складною та ресурсомісткою [7].

Іншим методом, що часто використовується при обробці природної мови за участі LLM, є візуалізація уваги або візуалізація важливих ознак [9]. Цей підхід ґрунтується на аналізі механізму self-attention, який є ключовим компонентом трансформерних архітектур, зокрема моделей типу BERT, GPT тощо. У трансформерах кожне слово (або токен) аналізується не ізольовано, а з урахуванням інших слів у контексті. Це досягається за допомогою матриць уваги, які відображають ступінь важливості одного слова стосовно інших [10]. Візуалізація цих матриць дає змогу побачити (рисунок 1.1), які саме частини вхідного тексту найбільше впливають на прийняття рішення або формування відповіді моделлю.

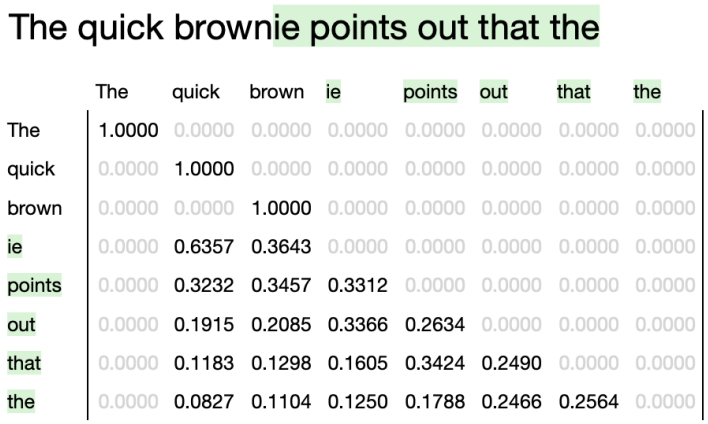


Рисунок 1.1 – Матриця уваги [8]

Крім того, візуалізація механізму уваги може бути реалізована у вигляді теплових карт (heatmaps), які демонструють інтенсивність взаємодії між парами слів у вхідній та вихідній послідовностях, або через графи залежностей, що ілюструють напрямки та силу впливу одних токенів на інші. Однак, попри високу наочність і зручність інтерпретації, цей метод має певні обмеження. Зокрема, увага не завжди корелює з причинно-наслідковими зв’язками. Інакше кажучи, модель може «спостерігати» одні частини тексту, але ґрунтувати своє рішення на

зовсім інших факторах. Це створює ризик хибної інтерпретації, коли увага сприймається як доказ впливу, хоча насправді вона лише відображає розподіл інформації, а не її значущість у прийнятті остаточного рішення [11].

Ще одним популярним методом пояснення роботи LLM є інтеграція градієнтів (Integrated Gradients або IG) [12]. Цей підхід належить до класу методів градієнтної інтерпретації та спрямований на визначення внеску окремих вхідних ознак у формування остаточного прогнозу моделі. Ідея методу полягає у тому, щоб обчислити сумарний вплив кожної ознаки, інтегруючи градієнти виходу моделі відносно цієї ознаки вздовж шляху від деякого базового вхідного вектора до фактичного вхідного прикладу. Перевагою IG є їхня теоретично обґрунтована властивість адитивності, що означає, що сума внесків усіх ознак приблизно дорівнює різниці між значенням моделі для фактичного прикладу і базового вектора [13]:

$$\sum_i IG_i(x) \approx f(x) - f(x'). \tag{1.1}$$

Цей метод широко використовується у завданнях обробки природної мови, оскільки дає змогу кількісно оцінити важливість окремих токенів у формуванні відповіді або класифікації. Крім того, інтегровані градієнти є масштабованими та можуть застосовуватися до складних моделей, включаючи трансформери. Однак для моделей із довгими послідовностями та високою розмірністю вхідних даних обчислення інтеграла може бути ресурсомістким.

Ще одним поширеним підходом у сфері XAI є використання моделей-замінників (surrogate models) [14] – спрощених інтерпретованих моделей, що наближено відтворюють поведінку складних моделей, зокрема LLM. До таких підходів належать LIME та SHAP, а також простіші алгоритми, як-от дерева рішень, лінійні регресії чи системи правил. Основна ідея полягає в побудові моделі g, яка б наближено відображала поведінку базової складної моделі f на заданій множині вхідних даних. Модель-замінник дає змогу простіше інтерпретувати процес прийняття рішень, виявляти потенційні упередження та підвищувати прозорість і довіру до системи.

Одним із прикладів застосування цього підходу є дослідження Марко Ріберо, Сінтії Анни та інших співавторів [14], де детально описано методику побудови локальних моделей-замінників для пояснення рішень «чорних скриньок». Втім, цей підхід має і певні обмеження. Зокрема, при апроксимації дуже складних моделей – таких як трансформери з мільйонами або мільярдами параметрів – моделі-замінники можуть втрачати точність, узагальненість або важливі нюанси в логіці прийняття рішень. У таких випадках пояснення можуть бути лише частковими або навіть оманливими, що обмежує ефективність цього підходу в складних сценаріях.

Окрім уже згаданих підходів, існує ще чимало інших методів інтерпретації, зокрема: пояснення на основі концептів (concept-based explanations), контрфактичні пояснення, аналіз змагальних прикладів для виявлення вразливостей; а також різноманітні методи декомпозиції, що дають змогу розкласти передбачення моделі на складові частини, щоб краще зрозуміти, як кожен компонент вплинув на результат. У сукупності ці підходи поглиблюють розуміння внутрішніх механізмів роботи моделей і сприяють створенню більш прозорих систем [12].

Post-hoc методи, такі як LIME, SHAP, IG, візуалізація уваги, зондування, моделі-замінники та інші, намагаються надати пояснення вже після отримання результату від моделі. Вони не впливають на структуру самої моделі, а лише створюють інтерпретативні структури або візуалізації, які допомагають краще зрозуміти, чому модель дійшла до певного результату.

Попри широке застосування, post-hoc пояснення мають фундаментальне обмеження – вони не гарантують повної прозорості або правдивості. Їхня інтерпретація може бути частковою, локально точною, але глобально хибною. Оскільки ці методи не є частиною самої моделі, вони не завжди відображають справжній процес прийняття рішень і можуть створювати ілюзію зрозумілості. Це особливо проблематично для LLM, поведінка яких нерідко є складною, непрогнозованою і чутливою до контексту. Як наслідок, довіра до таких моделей з боку користувачів та фахівців може суттєво зменшуватися.

У відповідь на це обмеження формується альтернативна парадигма – інтерпретованість-за-дизайном. В її основі лежить ідея про те, що саму модель необхідно будувати з урахуванням пояснюваності ще на етапі проєктування. Це означає створення таких архітектур, які або є повністю прозорими, наприклад, дерева рішень, логістична регресія, нейросимволічні моделі, або включають структурні елементи, які дають змогу контролювати або зрозуміти логіку прийняття рішень [15].

Дерева рішень [16] є класичним прикладом моделей, побудованих за принципом інтерпретованості-за-дизайном. Вони функціонують шляхом послідовного розгалуження даних на основі умов типу if-then, що забезпечує прозору логіку прийняття рішень. Кожна внутрішня вершина дерева відповідає перевірці певної ознаки, тоді як листові вузли представляють остаточне рішення або класифікацію. Завдяки такій структурі дерева рішень дають змогу отримати чітке та зрозуміле пояснення процесу класифікації або прогнозування. Їхня прозорість і структурована логіка роблять ці моделі особливо цінними в контекстах, де потрібна довіра до результатів моделі та можливість їх аудиту.

Нейросимволічні моделі [17], з іншого боку, є прикладом гібридного підходу, що поєднує переваги глибокого навчання з символічною логікою. Ці моделі використовують нейронні мережі для обробки неструктурованих або складних вхідних даних, таких як зображення, мова або текст, і поєднують їх із символічними компонентами – логічними правилами, онтологіями або змінними, які надають формальні пояснення висновків. Такий підхід дає змогу досягати як високої продуктивності на складних задачах, так і певного рівня інтерпретованості, завдяки можливості контролювати та пояснювати логіку обґрунтувань. Прикладом може слугувати архітектура, яка розширює LSTM або трансформери з логічними правилами, зокрема у випадках, коли необхідно перевірити відповідність об’єктів сцені певним концептуальним шаблонам, як-от «усі об’єкти відповідають структурі XYZ».

Ще одним напрямом реалізації інтерпретованості-за-дизайном є прозорі нейромережеві архітектури [18], зокрема Self-Explaining Neural Networks (SENN)

та ProtoPNet. Їхня ключова ідея полягає в тому, щоб інтегрувати у саму архітектуру механізми, які роблять внутрішні стани моделі зрозумілими для людини, наприклад:

– SENN побудовані так, щоб прогноз моделі пояснювався через обмежений набір семантично зрозумілих ознак. Мережа не просто видає результат, а й формує ваги цих ознак, показуючи, який саме внесок зробив кожен фактор у фінальне рішення;

– ProtoPNet [19] – прототипічна частинна мережа, що порівнює об’єкти з набором попередньо вивчених «прототипів». Для зображень це означає розбиття на семантично значущі фрагменти та пояснення рішення за принципом: «цей фрагмент схожий на ось цей прототип». Такий підхід забезпечує інтуїтивно зрозумілу класифікацію, але його складно без модифікацій застосувати до текстових або мультимодальних даних;

Проте повна інтерпретованість-за-дизайном погано масштабується до складних моделей, які потрібні для обробки неструктурованих, семантично багатих даних – таких як новинні тексти або діалоги. Саме тому набуває актуальності гібридний підхід, що поєднує обидві парадигми.

Отже, на основі проведеного аналізу вирішено розробити метод інтерпретування, що поєднує локальні підходи XAI (зокрема LIME, SHAP та IG) та, за можливості, елементи глобальної інтерпретації. Обрані методи дають змогу пояснювати рішення моделі на рівні окремих текстів, водночас забезпечуючи узгодженість між поясненнями для різних випадків. Додатково враховується концепція «інтерпретованості-за-дизайном», яка спрямована на зменшення «чорної скриньки» трансформерних моделей і наближення їх роботи до більш прозорих систем. Завдяки такому підходу модель, яка зробить прогнози, надасть користувачу зрозумілу аргументацію, що критично важливо для різних завдань.

1.2 Огляд методів аналізу та візуалізації текстових ембедінгів

У LLM центральну роль відіграють текстові ембедінги (вбудування) – щільні векторні подання слів, фраз, речень або навіть цілих текстів у багатовимірному просторі. Вони дають змогу перетворити текстові дані у формат, придатний для обробки нейромережею. Ембедінги зберігають семантичну інформацію: наприклад, слова з подібним значенням матимуть близькі вектори. Розуміння того, як моделі «кодують» семантичну інформацію, є критично важливим як для вдосконалення моделей, так і для забезпечення прозорості їхніх рішень. Завдяки цьому LLM здатні «розуміти» контекст, знаходити синоніми, узагальнювати чи робити логічні висновки [20].

Векторні представлення можна ефективно використовувати для розв’язання широкого спектра задач обробки природної мови, зокрема для категоризації текстів, виявлення емоційної тональності, пошуку релевантної інформації, генерації відповідей на запитання, автоматичного перекладу тощо – без потреби у створенні окремих моделей для кожного з цих завдань. Крім того, векторні подання демонструють високу продуктивність при роботі з об’ємними та неоднорідними текстовими корпусами [21]. Проте, щоб краще інтерпретувати та досліджувати ці векторні подання, особливо у контексті складних багатовимірних моделей, необхідно використовувати спеціальні методи їх аналізу та візуалізації. Для цього застосовуються як традиційні методи аналізу на кшталт зниження розмірності, так і інтерпретовані підходи XAI.

До традиційних підходів аналізу текстових вбудувань зазвичай відносять:

– методи зниження розмірності для візуалізації векторного простору ембедінгів, поміж яких найбільш поширені t-SNE, PCA (аналіз головних компонент), UMAP [22];

– аналіз подібності за допомогою метрик на кшталт косинусної подібності, який дає змогу оцінити, наскільки два вбудування близькі у семантичному сенсі [23];

– оцінка стабільності ембедінгів, що передбачає порівняння векторних представлень одного і того ж слова або речення в різних контекстах або при різних ініціалізаціях моделі;

– використання анотацій та метаданих, наприклад, приєднання інформації про частину мови, категорію або джерело тексту до ембедінгів може допомогти краще інтерпретувати отримані вектори або зрозуміти кластери в проєктованому просторі.

Щодо інтерпретованих підходів XAI аналізу векторних вбудувань доволі поширеним є застосування методу зондування (probing tasks). Такий метод дає змогу здійснити інтерпретацію того, яку саме лінгвістичну або семантичну інформацію модель закодовує у своїх представленнях [24].

Традиційні підходи до аналізу текстових ембедінгів забезпечують загальне розуміння простору векторних представлень та дають змогу виявити схожість або відмінність між текстовими елементами, а також оцінити стабільність і значущість ознак, які моделі захоплюють у процесі навчання. Водночас саме методи зниження розмірності, зокрема t-SNE та UMAP, є ключовими інструментами для візуального дослідження таких багатовимірних просторів.

Алгоритм t-SNE (t-Distributed Stochastic Neighbor Embedding) – це популярний алгоритм нелінійного зменшення розмірності, який ґрунтується на підході стохастичного вбудовування сусідів, запропонованому Джеффрі Хінтоном і Семом Роуейсом, а згодом алгоритм удосконалений, зокрема шляхом використання t-розподілу (розподілу Стьюдента) для зменшення перекриття кластерів у низьковимірному просторі [25]. Метою t-SNE є збереження локальної подібності між точками при переході від простору високої розмірності до двовимірного або тривимірного простору для подальшої візуалізації. Основні принципи роботи:

– у високовимірному просторі: t-SNE оцінює ймовірність, що точка x_i є сусідом точки x_i, використовуючи гаусівський розподіл;

– у низькорозмірному просторі: ті ж самі ймовірності моделюються за допомогою t-розподілу з одним ступенем свободи, що дає змогу краще відображати віддалені точки й уникати скупчення кластерів.

Алгоритм t-SNE має низку суттєвих переваг і водночас певні обмеження. До його сильних сторін належить те, що він добре підходить для візуалізації прихованих структур у даних, зокрема кластерів, і зберігає локальні відносини між точками, що робить його корисним для аналізу простору ознак.

Разом з тим, t-SNE має і недоліки. Він погано масштабується на великі набори через високу обчислювальну складність, не дає змогу відновити вихідні ознаки з проєкції та може давати різні результати при кожному запуску без фіксації випадкового стану.

Ще одним ефективним методом зниження розмірності для подальшої візуалізації векторних ембедінгів є UMAP (Uniform Manifold Approximation and Projection) [22], який ґрунтується на теорії багатовидів та алгебраїчній топології. Алгоритм прагне зберегти як локальну, так і глобальну структуру даних, що робить його корисним для аналізу складних високорозмірних просторів. Принцип роботи алгоритму UMAP складається з кількох етапів [26]:

– спочатку метод формує граф близькості у вихідному просторі, використовуючи підхід k-найближчих сусідів для оцінювання ймовірності сумісного розташування точок;

– далі він генерує аналогічний граф у просторі нижчої розмірності;

– після відбувається оптимізація шляхом мінімізації розбіжності між двома графами, використовуючи стохастичний градієнтний спуск.

На відміну від t-SNE, UMAP працює значно швидше, особливо на великих наборах даних, краще зберігає глобальну структуру, а також є оборотним, що дає змогу здійснювати зворотне перетворення. Проте метод має і певні недоліки, зокрема, він схильний до створення надто щільних кластерів, навіть коли у вихідному просторі дані розташовані більш рівномірно, а також демонструє чутливість до вибору гіперпараметрів, таких як кількість сусідів та мінімальна відстань між точками тощо [27].

Іншими методами у межах традиційного підходу до аналізу векторних представлень є методи обчислення подібності. До них належать такі базові метрики, як косинусна подібність, скалярний добуток (Dot Product), L1- та L2-норми. Ці підходи дають змогу враховувати контекстні особливості та структурні зв'язки між елементами, що робить їх особливо ефективними у задачах пошуку, класифікації та кластеризації векторних представлень [28, 29].

Косинусна подібність вимірює кут між двома векторами, не зважаючи на їхню довжину. Завдяки цьому вона добре підходить для задач обробки текстів, де важливо порівнювати напрямки векторів, але водночас такий підхід ігнорує масштаб та може бути чутливим до шуму. Інший підхід – скалярний добуток, який просто множить відповідні координати векторів і підсумовує результат. Цей метод легкий у реалізації та корисний для не-нормалізованих представлень, однак його значення сильно залежить від довжини векторів і може бути складним для інтерпретації.

Метод L1-норми оцінює подібність як суму абсолютних різниць між координатами, що робить його інтерпретованим і точним для розріджених даних. Проте він менш чутливий до великих відмінностей і не є масштабно-інваріантним. L2-норма, або евклідова відстань, є більш геометрично інтуїтивною та часто використовується в задачах кластеризації. Проте вона дуже залежить від масштабу та погіршує роботу у високорозмірних просторах, де відстані між точками стають менш інформативними.

Окрім традиційних методів оцінювання подібності ембедінгів, доцільно враховувати й додаткові підходи, які дають змогу глибше проаналізувати стабільність та інтерпретованість векторних представлень. Зокрема, аналіз Прокруста (Procrustes Analysis) [30], який забезпечує вирівнювання двох векторних просторів шляхом лінійного перетворення одного з них, дозволяючи таким чином виміряти структурну схожість між ембедінгами, отриманими, наприклад, з різних корпусів або моделей. Це особливо корисно порівнюючи динаміку значень ембедінгів у часових мовних моделях.

Інший підхід – оцінювання надійності тест-ретест (Test-Retest Reliability) [31] – передбачає перевірку стабільності ембедінгів при повторному тренуванні моделі або застосуванні до схожих текстів, що дає змогу виявити нестійкість, пов’язану зі стохастичними процесами в навчанні. Наприклад, можна обчислити кореляцію або відстань між ембедінгами одного й того ж слова, отриманими після декількох запусків навчання.

Нарешті, метод збагачення ембедінгів ознаковою інформацією (Feature-based Augmentation) [32] шляхом додавання семантичних тегів, синтаксичних категорій чи метаданих (наприклад, авторства чи жанру тексту). Такий метод може підвищити якість аналізу подібності, забезпечуючи додаткові осі варіативності векторного простору.

Отже, розглянуто різні підходи для аналізу текстових ембедінгів, поміж яких косинусна подібність, скалярний добуток, L1- та L2-норми, а також складніші функції втрат, як-от триплетна та контрастивна. Кожен із цих методів має свої переваги залежно від завдання: прості метрики добре працюють для оцінювання схожості між векторами, тоді як спеціалізовані функції втрат орієнтовані на покращення навчання моделей у багатовимірному просторі. Для реалізації системи виявлення та пояснення фейкових новин як основний інструмент обрано косинусну подібність, оскільки вона дає змогу точно оцінювати семантичну близькість між текстами та є стандартом у роботі з ембедінгами. Водночас архітектура системи залишається гнучкою й за потреби може бути розширена використанням альтернативних методів подібності або спеціалізованих функцій втрат.

1.3 Аналіз наявних програмних засобів та фреймворків для виявлення фейкових новин

З метою аналізу сучасних програмних рішень, орієнтованих на виявлення фейкових новин із використанням ШІ, відібрано кілька інструментів. Враховуючи, що більшість доступних платформ орієнтовані переважно на

англомовний інформаційний простір, дослідження зосереджене на двох таких системах та двох фреймворках. Було обрано – систему «Logically Intelligence», що розроблена компанією Logically, яка поєднує автоматизований аналіз контенту з участю експертів для виявлення дезінформації, а також інструмент XFake, що реалізує візуалізацію та пояснення прийнятих рішень, дозволяючи користувачам краще зрозуміти, на основі яких характеристик контент класифіковано як потенційно фейковий. Додатково можна розглянути два фреймворки інтерпретації моделей: Captum (Meta AI), як бібліотеку для PyTorch, що надає широкий спектр методів інтерпретації нейронних мереж та Transformers Interpret, який є інструментом для швидкої візуалізації впливу окремих слів і фраз на результати моделей з бібліотеки Hugging Face Transformers.

Logically Intelligence [33] – це потужна платформа на стику AI та людської експертизи, яка призначена для масштабного виявлення та аналізу наративів дезінформації. Вона дає змогу автоматично вилучати твердження з текстів, перевіряти їх правдивість, аналізувати наративи, ключові слова, джерела та поширення контенту. Платформа підтримує понад 57 мов, працює з відео й зображеннями, а також має інструменти для ініціювання розслідувань або повідомлень на платформи соцмереж. Система обробляє мільйони джерел і дає змогу відстежувати розвиток меседжів та їх взаємозв’язки. Основна перевага – поєднання автоматичного оцінювання з ручною перевіркою, що зменшує кількість помилок. Поміж недоліків – закритість алгоритмів, складний інтерфейс для недосвідчених користувачів і обмежена можливість самостійної перевірки фактів, зокрема помилкової класифікації під час пандемії [34].

Щоб ознайомитися з функціональністю системи, слід перейти на офіційний сайт компанії Logically та надіслати запит на отримання демоверсії. Після підтвердження доступу відкривається інтерфейс платформи, зокрема вкладка «Situation Rooms», яка забезпечує централізований моніторинг інформаційних загроз (рисунок 1.2).

Інтерфейс розділений на дві основні частини: ліворуч розташована панель керування з вкладками для навігації між аналітичними модулями, праворуч –

робоча область, яка, своєю чергою, поділена на два блоки. Один з них призначений для ідентифікації фейкових та правдивих новин, інший – для аналізу сентименту, структури контенту, ключових тем і авторів поширення. Такий поділ дає змогу швидко оцінити поточну інформаційну ситуацію та прийняти відповідні дії.

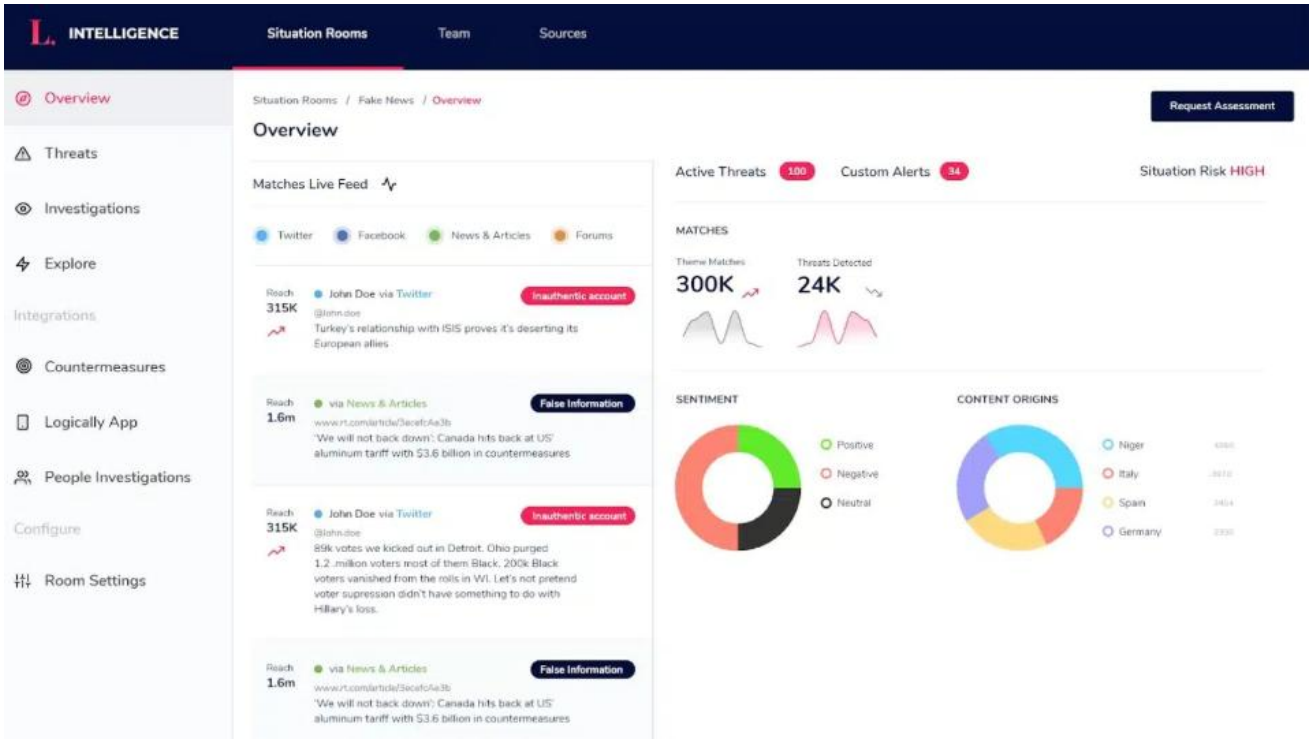


Рисунок 1.2 – Виявлення фейкових новин в Logical Intelligence [35]

Іншим застосунком є XFake [36] – інструмент для виявлення фейкових новин, головною перевагою якого є прозорість та зрозумілість результатів. Система застосовує методи машинного навчання для аналізу тексту та надає пояснення, чому певну новину класифіковано як перевірену або фейкову. Ключова функція платформи це інтерактивна візуалізація, що демонструє, які характеристики тексту (слова, тон, структура) вплинули на висновок моделі. XFake часто використовується в наукових дослідженнях для вивчення поведінки NLP-моделей. Алгоритми базуються на лінгвістичних та статистичних ознаках, зокрема частоті слів, емоційності, специфічних фразах. Платформа забезпечує прозорість процесу класифікації, підсвічує ключові ознаки, дає змогу аналізувати

вплив окремих факторів на результат та має відкритий код, що робить її придатною для розширення та наукових експериментів.

Проте платформа має також обмеження: продукт не призначений для роботи в реальному часі або на великих обсягах даних; працює лише з англомовними текстами; має низьку адаптивність до нових форм дезінформації без додаткового навчання та має обмежену масштабованість у порівнянні з комерційними продуктами.

Система XFake забезпечує детальний аналіз новинних повідомлень, розкладаючи їх на різні типи n-грам для глибшого семантичного аналізу тексту. Крім цього, вона дає змогу виокремлювати лінгвістичні компоненти речень, що візуалізується на рисунку 1.3. Однією з ключових можливостей XFake є адаптація моделі для генерації пояснень на основі ансамблю дерев рішень, що демонструє логіку прийняття рішень: які саме ознаки мали найбільший вплив на класифікацію новини як правдивої або фейкової. Отже, користувач може не лише отримати результат, а й зрозуміти обґрунтування моделі.

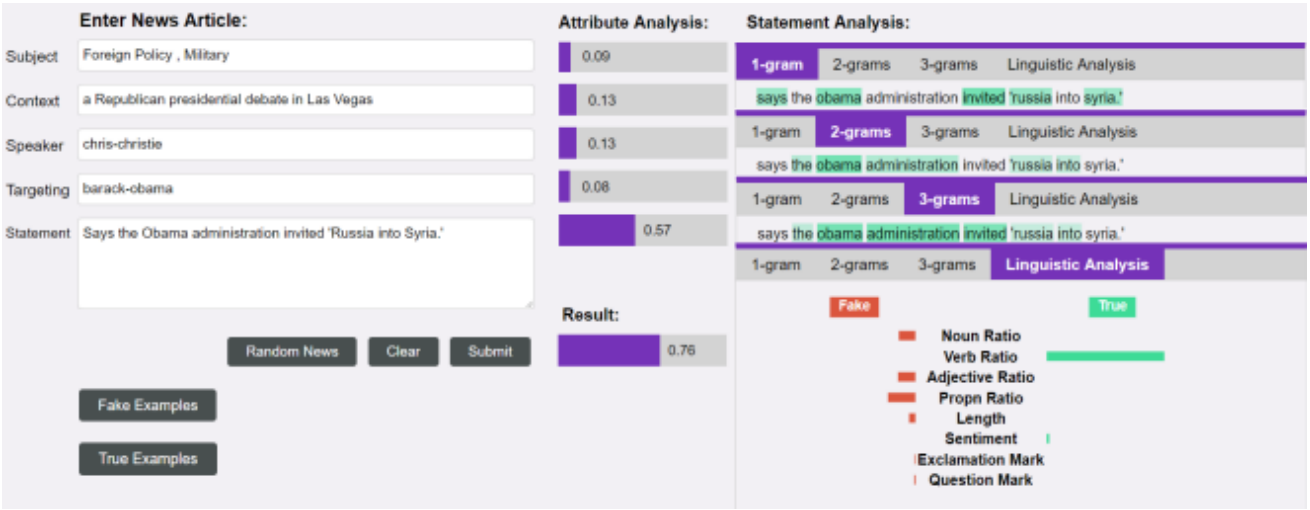


Рисунок 1.3 – Передбачення та пояснення рішень від XFake [36]

Серед інструментів для інтерпретації моделей варто виділити продукт від Meta AI – Captum [37], який створений для пояснення роботи моделей на базі PyTorch. Він підтримує широкий набір алгоритмів інтерпретації, зокрема Integrated Gradients, Layer Conductance, DeepLIFT, Saliency та інші. Captum дає

змогу визначати важливість вхідних ознак, аналізувати вплив окремих шарів та підмоделей, а також створювати візуалізації, які допомагають зрозуміти логіку прийняття рішень нейронною мережею. Перевагою фреймворку є гнучка інтеграція в наявні пайплайни PyTorch і можливість пояснювати як прості, так і складні багатомодальні моделі. Хоча даний продукт непогано працює з текстом, але все ж він краще може пояснювати саме зображення чи табличні дані, а також для LLM потрібна додаткова обгортка або API.

Інший інструмент – Transformers Interpret [38], який побудований на базі Captum і є надбудовою для бібліотеки Hugging Face Transformers, орієнтованою на зручну візуалізацію рішень моделей у задачах NLP. Фреймворк надає простий інтерфейс для визначення важливості токенів і фраз, відображаючи їх у вигляді інтерактивних кольорових підсвіток. Він підтримує популярні архітектури, такі як BERT, RoBERTa, DistilBERT та GPT-4. Основна мета фреймворку – зробити інтерпретацію трансформерних моделей доступною без необхідності глибокого занурення у внутрішню реалізацію алгоритмів, що особливо корисно у прикладних проєктах, зокрема для виявлення фейкових новин та пояснення причин класифікації.

У ході аналізу розглянуто наявні програмні рішення та бібліотеки, що застосовуються для задачі виявлення фейкових новин і пояснення роботи моделей. Встановлено, що методів і підходів для інтерпретації рішень нейронних мереж існує досить багато – від класичних LIME та SHAP до більш сучасних градієнтних і концептуальних методів. Водночас більшість готових інструментів зосереджені на локальному поясненні без інтеграції механізмів покращення моделі через участь користувача. Реалізацій, які б безоплатно підтримували підхід «людина-в-петлі» та давали змогу адаптивно вдосконалювати інтерпретацію, практично немає. Саме тому розробка подібної системи є актуальною на сучасному етапі розвитку інструментів XAI.

1.4 Мета та постановка задачі з вимогами

Актуальною проблемою сучасних систем виявлення дезінформації є «непрозорість» роботи складних нейромережевих архітектур. Сучасні LLM забезпечують високу точність класифікації, проте не надають користувачеві зрозумілих пояснень щодо причин ухвалення конкретного рішення, що знижує довіру до таких систем.

Метою роботи є підвищення рівня інтерпретованості та точності систем виявлення фейкових новин через проєктування методу, який забезпечує інтерактивний аналіз, валідування та ітеративне вдосконалення простору текстових ембедінгів із залученням експерта.

Для досягнення поставленої мети необхідно виконати такі завдання:

1. Провести аналіз методів забезпечення прозорості LLM, підходів до візуалізації текстових ембедінгів та наявних програмних рішень для виявлення дезінформації в текстових даних.

2. Спроєктувати метод інтерпретування результатів виявлення фейкових новин, що полягає в генеруванні високоінформативних текстових ембедінгів для новинних статей із використанням архітектур великих мовних моделей.

3. Розробити підсистему зниження розмірності та візуалізації багатовимірних ембедінгів у двовимірному просторі за спроєктованим методом.

4. Розробити архітектуру інтерактивної системи та виконати її програмну реалізацію, що надає інструменти для дослідження проєкцій, пояснення рішень та реалізує підхід “людина-у-петлі” для виявлення фейкових новин.

5. Провести експериментальне дослідження спроєктованого методу та його програмної реалізації за еталонними наборами даних та оцінити результативність запропонованих підходів.

Основні вимоги до методу та системи є такими:

– інтерпретованість: система має надавати локальні (для окремого тексту) та глобальні (для корпусу) пояснення рішень моделі;

– інтерактивність: забезпечення можливості візуального аналізу простору ембедінгів та ручного корегування параметрів класифікації;

– точність: показники метрик якості виявлення не повинні бути нижчими за базові підходи;

– масштабованість: архітектура має підтримувати оброблення великих корпусів текстових даних та можливість інтеграції нових моделей трансформерів.

РОЗДІЛ 2 Проєктування методу та інформаційної системи інтерпретування результатів виявлення фейкових новин за великою мовною моделлю

2.1 Схема методу інтерпретування результатів виявлення фейкових новин за великою мовною моделлю

Перед початком програмної реалізації методу інтерпретування результатів виявлення фейкових новин за великою мовною моделлю визначаються ключові етапи його роботи та функціональні завдання системи. Оскільки основна мета полягає у класифікації текстів новин і подальшому поясненні отриманих результатів, спроєктований метод структуровано у шість послідовних кроків – від надходження сирих текстових даних до формування інтерпретованого прогнозу (рисунок 2.1). На виході створюється класифікаційний висновок із ймовірнісною оцінкою та набір пояснень, що містить ключові слова й фрагменти, які вплинули на результат, а також інтерактивні візуалізації латентного простору новин, побудовані на основі методів зниження розмірності UMAP та t-SNE.

Крок перший – Попередня обробка тексту. Першим етапом є формування уніфікованого формату вхідних даних, що охоплює заголовок, текст новини, метадані (джерело, дату, посилання) та супровідні атрибути. Оскільки готові датасети часто містять неузгодженості, цей етап передбачає ретельну попередню обробку: очищення від технічних артефактів, HTML-тегів, емодзі та некоректних символів, нормалізацію регістру, лематизацію (або стемінг), видалення стоп-слів і стандартизацію дат та числових позначень. Така підготовка покращує якість ембедінгів і підвищує стабільність моделей [39]. Важливо також збалансувати класи у вибірці, забезпечуючи рівне представлення правдивих і фейкових новин, щоб уникнути зміщення моделі в бік домінантного класу. За наявності специфічних особливостей, як-от доменні назви чи мультимовні дані, приклади необхідно привести до спільного стандарту, наприклад до єдиного мовного формату [40].

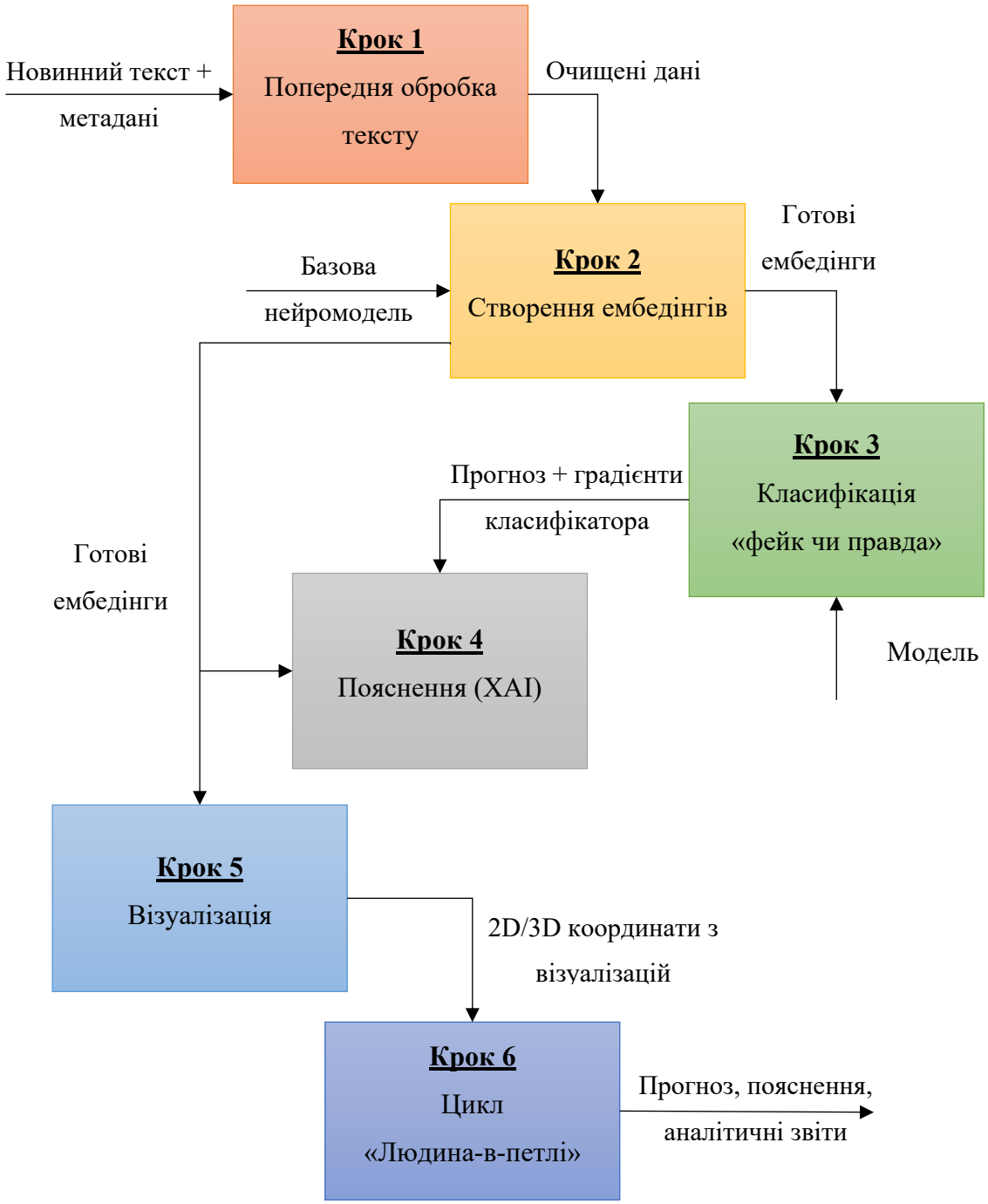


Рисунок 2.1 – Схема методу інтерпретування результатів виявлення фейкових новин за великою мовною моделлю

Крок другий – Створення ембедінгів. Після попередньої обробки тексту новини дані подаються у вигляді векторних представлень (ембедінгів), придатних для роботи класифікаційних моделей. Ембедінги перетворюють сирий текст у числовий простір високої розмірності, де семантично схожі слова та речення

розташовані близько одне до одного. Для цього використовуються попередньо навчені трансформерні моделі, адаптовані під завдання NLP.

У межах проєкту заплановано застосування двох моделей різного рівня складності. Базова модель створює ембедінги за допомогою Sentence-BERT (SBERT), оптимізованої для отримання високоякісних векторних представлень речень і абзаців. Завдяки сіамській архітектурі SBERT обчислення будуть виконуватися швидше, а отримані ембедінги придатні для подальшого порівняння, класифікації та кластеризації текстів. Складна модель передбачає використання DistilBERT, стисненої версії BERT, яка зберігає близько 95% точності оригіналу, працює швидше та потребує менше обчислювальних ресурсів.

Крок третій – Класифікація тексту. Після формування векторних подань тексту за допомогою відповідної моделі (SBERT або DistilBERT) третім етапом є безпосередня класифікація новин. У базовому підході ембедінги, отримані від Sentence-BERT, передаються до окремого класифікатора, що працює з числовими ознаками. Для цього використовується логістична регресія – класичний алгоритм машинного навчання, до яких належать семантичні ембедінги. Модель формує лінійну межу, яка максимально розділяє приклади класів «фейк» і «правда», забезпечуючи стабільність результатів навіть при обмеженій кількості тренувальних даних. У такій конфігурації кожен ембедінг розглядається як точка у багатовимірному семантичному просторі, а логістична регресія оптимально підбирає ваги ознак для підвищення точності класифікації [41].

На відміну від базової конфігурації, складна модель на основі DistilBERT об’єднує створення ембедінгів і класифікацію в єдину трансформерну архітектуру, що працює кінець-до-кінця. Це дає змогу уникнути розділення процесу на два окремі етапи та отримати більш глибоке контекстне кодування тексту, що зазвичай підвищує точність розпізнавання «фейкових» новин у порівнянні з класичним підходом SBERT + окремий класифікатор.

Крок четвертий – Створення пояснень. Однією з важливих вимог до розробленого методу є не лише здатність моделі автоматично класифікувати новини як «фейкові» або «правдиві», а й уміння пояснювати, чому було прийняте

те чи інше рішення. Тому четвертий етап роботи системи присвячений формуванню інтерпретаційних пояснень, які роблять результати класифікації прозорими та зрозумілими для користувача.

У межах базової моделі використовується поєднання двох локальних підходів XAI. Метод SHAP дає змогу кількісно оцінити внесок кожної ознаки (токена або виміру ембедінга) у кінцевий прогноз, забезпечуючи строго обґрунтоване математичне тлумачення. Паралельно застосовується LIME, який будує спрощену локальну модель навколо конкретного тексту та виділяє слова, що найбільше вплинули на результат. Це дає змогу користувачеві побачити не лише числовий прогноз, а й зрозуміти логіку, за якою модель ухвалила рішення. Для складнішого підходу на основі DistilBERT додатково використовується метод Integrated Gradients. Він аналізує, як змінюються градієнти моделі під час переходу від нейтрального (базового) вектора до реального тексту, завдяки чому можна визначити реальний вплив кожного токена на результат класифікації.

Важливою частиною етапу інтерпретації є організація збереження всіх отриманих пояснень. Результати фіксуються у форматі JSON, де зберігаються прогноз моделі разом із ймовірністю, перелік ключових слів із вагами за SHAP або LIME, а також значення атрибуції за Integrated Gradients. Така уніфікована структура полегшує подальший аналіз, дає змогу повторно використовувати отримані дані та забезпечує інтеграцію з інструментами візуалізації в інтерактивних аналітичних системах.

Крок п’ятий – Візуалізація ембедінгів. Для глибшого розуміння можливих помилок моделі на етапі класифікації важливо візуалізувати простір ознак, знизивши розмірність ембедінгів. Це відбувається на п’ятому етапі методу, де векторні подання текстів проєктуються у простір меншої розмірності, що дає змогу оцінити структуру даних та якість попередніх кроків. Отримана проєкція робить дані інтерпретованими: можна побачити, як новини розміщуються у латентному просторі, наскільки чітко розділяються класи «фейк» і «правда», а також виявити можливі кластери чи аномалії. Одним із кращих методів для такої задачі є UMAP, який формує дво- або тривимірні мапи, де тексти групуються

відповідно до їхньої семантичної спорідненості. Це дає змогу простежити тематичні зони, наприклад групи політичних або сенсаційних новин. Поряд з UMAP часто застосовується t-SNE, який краще зберігає локальну структуру та дає змогу виявляти дрібні семантичні кластери, хоч і є більш ресурсомістким. Поєднання цих методів дає ширший погляд на структуру даних і забезпечує додатковий рівень інтерпретації. Результати проєкції можуть зберігатися у вигляді векторних файлів (наприклад, .csv, .npy, а також .svg чи .eps для графічних матеріалів), що дає змогу повторно використовувати їх без виконання дорогих обчислень і інтегрувати у подальші етапи аналізу.

Крок шостий – Робота циклу «людина-в-петлі». Шостий етап методу передбачає інтеграцію результатів моделі в аналітичний процес за участю експерта через підхід «людина-в-петлі», що забезпечує ітеративну оцінку та корекцію роботи системи. На цьому етапі модель не функціонує як автономна система: її передбачення, пояснення за допомогою LIME, а також візуалізації, такі як матриця помилок, ROC-крива та інші метрики, оцінюються користувачем. За потреби експерт може змінювати параметри моделі або додавати нові тексти для покращення навчання і покриття рідкісних чи складних патернів. Після внесення змін модель перенавчається, і результати знову виводяться у вигляді всіх необхідних даних, включаючи інтерактивні пояснення для окремих новин. Цей процес повторюється ітеративно: експерт переглядає результати, вносить корективи та запускає повторне навчання, поки система не досягне бажаного рівня точності та стабільності. У базовій конфігурації це дає змогу підвищити показники логістичної регресії до приблизно 65–75%, тоді як складніші моделі на основі DistilBERT можуть демонструвати значно вищі значення точності при належному донавчанні. Такий підхід гарантує підвищення якості класифікації та інтерпретованості результатів через постійне втручання експерта.

Для кращої ілюстрації роботи запропонованого методу розглянемо приклад обробки однієї новини, яка проходить повний цикл від вхідних сирих даних до інтерпретованого результату. Водночас буде показано проміжні формати

представлення даних на кожному з етапів методу. Узагальнене відображення цього процесу наведено у вигляді таблиці 2.1.

Таблиця 2.1 – Входи та виходи на кожному кроці

Крок	**Вхідні дані**	**Вихідні дані**	**Формат**	**Приклад**
Попередня обробка	Текст новини	Очищений текст	TXT	«breaking news: ...» → «breaking news»
Подання тексту	Очищений текст	Вектор ембедінгів	NumPy/CSV	[0.123, -0.045, …, 0.876]
Класифікація	Ембедінг	Прогноз	JSON	{«fake»: 0.82, «real»: 0.18}
Пояснення	Прогноз + текст	Важливі слова/фрази	JSON/HTML	{«trump»: 0.23, «scandal»: 0.15}
Візуалізація	Ембедінги	2D-координати	CSV/SVG	(x=0.45, y=−0.12)
Людина-в-петлі	Параметри моделі	Оновлений НД/модель	НД/ Checkpoint	Нова версія моделі → v2

Запропонований метод інтерпретування результатів виявлення фейкових новин за великою мовною моделлю ґрунтується на послідовному перетворенні даних: від попередньої обробки тексту до формування пояснень і інтерактивної взаємодії з користувачем. Кожен крок має чітко визначені вхідні та вихідні дані, що дає змогу інтегрувати модуль класифікації, XAI-методи та візуалізації в єдину архітектуру. Завдяки цьому користувач не лише отримує прогноз моделі, а й може зрозуміти його підґрунтя та впливати на подальше навчання системи.

2.2 Математична модель інтерпретування результатів виявлення фейкових новин за великою мовною моделлю

Для кращого розуміння поставленого завдання необхідно формалізувати математичну модель інтерпретування результатів виявлення фейкових новин за допомогою LLM. Для цього нижче подано узагальнення ключових обчислень, які складають основу побудови системи виявлення та пояснення. Запропонована модель охоплює такі блоки: побудову ембедінгів, класифікацію текстів, застосування методів пояснення, зниження розмірності та оцінку кластеризації.

Модель у методі інтерпретування починає роботу зі створення ембедінгів, які відображають як новинні тексти, так і їх метадані у числову форму, придатну для обчислень. Ембедінги дають змогу зберегти семантичну інформацію тексту, тобто у векторному представленні зберігається значення смислу слів, фраз та контексту, а не лише окремих термінів. Це дає змогу порівнювати тексти між собою, виявляти схожі за змістом новини, а також формувати кластери документів за тематичною ознакою. Векторні представлення є ключовими для подальшої класифікації, інтерпретації рішень моделі та аналізу структурованих зв'язків між документами.

Функція побудови ембедінгів формалізується наступним чином [42]:

$$\psi : D \to \mathbb{R}^k, \tag{2.1}$$

де D є простором вхідних текстових документів, а k – розмірність простору ембедінгів.

Для конкретної новини d отримаємо векторне подання

$$v = \psi(d). \tag{2.2}$$

У роботі метода планується використання моделей класу Sentence-BERT. Ця архітектура спеціально розроблена для формування векторних представлень на рівні речень і документів: вона враховує контекст слів у реченні, створює семантично змістовні ембедінги та зберігає інформацію про схожість між текстами. Sentence-BERT забезпечує стабільне та узгоджене векторне

представлення як коротких, так і довгих текстів, що є важливим для підвищення точності класифікації, побудови пояснень рішень моделі та аналізу структури новинних даних. Крім того, сучасні трансформерні архітектури дають змогу враховувати додаткову інформацію, наприклад метадані новин (дата публікації, автор, категорія), шляхом інтеграції відповідних векторних представлень у загальний ембедінг документа.

Окрім отримання ембедінгів моделями-трансформерами, для представлення текстів можна також застосувати класичний метод TF-IDF. Цей метод дає змогу оцінити відносну важливість кожного слова у документі, враховуючи його частотність у конкретному тексті та поширеність у всьому корпусі. TF-IDF широко використовується в інформаційному пошуку, автоматичній класифікації текстів та виявленні тематичної схожості між документами, оскільки він підкреслює характерні терміни та зменшує вплив загальних слів [43]:

$$TF\text{-}IDF(s,t) = TF(s,t) \cdot log\frac{N}{DF(s)}, \tag{2.3}$$

де s – слово, t – документ, N – кількість документів у корпусі та $DF(s)$ – кількість документів, що містять s.

Векторні подання документів можуть бути сформовані не лише трансформерними моделями або за допомогою TF-IDF, а й іншими методами представлення тексту. Для кількісного оцінювання подібності між отриманими векторами найчастіше застосовується косинусна міра, яка характеризує кут між векторами у багатовимірному просторі [44]:

$$\cos(d_i, d_j) = \frac{d_i \cdot d_j}{\|d_i\|\|d_j\|}, \tag{2.4}$$

де d_i та d_j це векторні представлення відповідних документів.

Косинусна подібність забезпечує оцінку схожості текстів на основі напрямку їхніх векторів незалежно від абсолютної величини.

Після отримання векторних представлень новин наступним етапом є їх класифікація за ознакою «фейкова чи правдива». Для цього вводиться функція

$$f(v) \in [0,1], \tag{2.5}$$

яка відображає ймовірність того, що новина є фейковою на основі її векторного подання e.

На практиці це означає, що для кожного документа модель видає значення, яке можна інтерпретувати як рівень впевненості у його належності до класу «фейкова новина». Щоб навчити модель правильно оцінювати цю ймовірність, необхідно мінімізувати різницю між прогнозованими значеннями $f(e)$ та істинними мітками документів. У рамках бінарної задачі класифікації стандартним інструментом для цього є функція втрат крос-ентропії, яка формально визначається наступним чином [45]:

$$L = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log f(e_i) + (1 - y_i)\log(1 - f(e_i))], \tag{2.6}$$

де n – кількість документів у навчальній вибірці, $y_i \in \{0,1\}$ – істинна мітка для i-го документа (0 – новина правдива, 1 – фейкова).

Крос-ентропійна втрата забезпечує диференційовану міру помилки, що дає змогу оптимізувати параметри моделі та досягати мінімальної невідповідності між прогнозованою ймовірністю і реальними мітками. Використання цієї функції втрат є загальноприйнятим підходом для бінарної класифікації та забезпечує стабільну збіжність під час навчання моделей на основі градієнтного спуску. Крім того, вихідне значення $f(e)$ може бути інтерпретоване як «рівень довіри» моделі до того, що новина є фейковою, що надалі дає змогу не лише класифікувати тексти, але й будувати системи пояснення рішень та оцінювання ризику поширення фейкової інформації.

Для забезпечення прозорості та інтерпретованості рішень моделі в системі застосовуються різні класи методів XAI. Основними підходами обрано кілька методів – локальну адитивну модель-замінник SHAP, метод градієнтного пояснення IG та локальну інтерпретацію за допомогою LIME. Метод SHAP належить до локальних адитивних моделей-замінників. Він дає змогу апроксимувати складну модель f простою, інтерпретованою функцією g у локальному околі конкретного прикладу x [46]:

$$g(x) \approx f(x), f(x) \approx \phi_0 + \sum_{j=1}^{d} \phi_j z_j, \quad (2.7)$$

де ϕ_j – внесок ознаки j у прогноз моделі, z_j це значення ознаки у локальному представленні.

SHAP дає змогу не лише визначити, які ознаки найбільше впливають на конкретне передбачення, але й побудувати прозорі інтерпретації для користувача або аналітика.

Іншим методом є Integrated Gradients, який базується на інтегралі градієнтів виходу моделі відносно вхідних ознак уздовж шляху від базового (нейтрального) прикладу x' до поточного прикладу x [47]:

$$IG_i(x) = (x_i - x_i') \cdot \int_0^1 \frac{dF\left(x' + a(x - x')\right)}{dF_i} da, \quad (2.8)$$

де x_i – значення ознаки i для даного прикладу, а F – вихід моделі. IG дає змогу отримати точні оцінки внеску кожної ознаки, враховуючи всю зміну моделі від базового стану до фактичного прикладу.

Метод добре працює з диференційованими моделями та забезпечує більш згладжене та стабільне пояснення, що дає змогу відстежувати, як конкретні ознаки впливають на прогноз у межах всього шляху інтеграції.

Останнім методом для інтерпретації рішень моделі обрано LIME, що оцінює внесок окремих ознак у прогноз моделі для конкретного прикладу. Для обчислення локальної важливості ознак LIME використовує ваги w локальної лінійної моделі, оптимізованої на околі обраного прикладу x [48]:

$$w^* = agr \min_w L(f, g, \pi_x) + \Omega(\mathrm{g}), \quad (2.9)$$

де f – вихідна модель, g це локальна інтерпретуюча модель (лінійна), π_x функція ваг, що визначає близькість сусідніх прикладів до x, L – функція втрат між прогнозами f і g, а $\Omega(g)$ це регуляризація, що обмежує складність локальної моделі.

Це особливо корисно для аналізу того, які ознаки моделі визначають класифікацію як «фейкова» чи «правдива» новина, та для перевірки, наскільки локальні пояснення відповідають очікуванням експертів. Для кращого розуміння

послідовності обчислень локальних методів, таких як SHAP та IG, в додатку А на рисунку А.1 подано алгоритмічну схему побудови локального пояснення прогнозу моделі. Схема ілюструє послідовність обчислень: від отримання вхідного прикладу та результату моделі до визначення внеску окремих ознак у прийняте рішення.

Перед аналізом структурних властивостей векторних представлень новин часто застосовують методи зниження розмірності, що дають змогу відобразити багатовимірні ембедінги у дво- або тривимірний простір для візуалізації та подальшого аналізу.

Найкраще з поставленою задачею справляються методи t-SNE та UMAP. Перший орієнтований на збереження локальної структури даних, забезпечуючи компактне групування схожих прикладів у проєкції, тоді як інший поєднує високу швидкість обчислень із кращою глобальною узгодженістю, що дає змогу одночасно відображати як локальні, так і загальні закономірності в даних. Використання цих методів дає змогу досліджувати взаємозв'язки між документами та підготовлює основу для оцінювання якості кластеризації.

Для кількісного оцінювання якості розбиття даних на кластери зазвичай застосовуються такі метрики, як коефіцієнт силуету та індекс Девіса-Боулдіна (DBI). Обидві метрики мають на меті вимірювання ступеня відокремленості та компактності кластерів, проте інтерпретація їхніх значень відрізняється. Високі значення коефіцієнта силуету (близькі до 1) свідчать про чітке відокремлення кластерів, тоді як від'ємні значення вказують на можливу некоректність кластеризації. У випадку індексу Девіса-Боулдіна, навпаки, менші значення є індикатором більш компактних та добре розділених кластерів.

Коефіцієнт силуету розраховується за такою формулою [49]:

$$s(i) = \frac{b(i)-a(i)}{\max\{a(i),b(i)\}}, \tag{2.10}$$

де $a(i)$ це середня відстань між елементом i та іншими елементами його кластера, а $b(i)$ – мінімальна середня відстань до елементів іншого кластера.

Тоді як індекс Девіса-Боулдіна обраховується за формулою [49]:

$$DBI = \frac{1}{K}\sum_{i=1}^{K} \max_{j \neq i} \frac{\sigma_i + \sigma_j}{d(c_i, c_j)}, \quad (2.11)$$

де σ_i це внутрішньокластерне відхилення для кластера i, $d(c_i, c_j)$ є відстанню між центроїдами кластерів i та j, а K – загальна кількість кластерів.

Розглянуті вище математичні положення формують методологічну основу для побудови та інтерпретації моделей виявлення фейкових новин. Функція втрат крос-ентропії забезпечує навчання класифікаційних моделей, мінімізуючи розбіжність між прогнозованими ймовірностями та істинними мітками. Методи інтерпретації, зокрема SHAP, IG та LIME, дають змогу кількісно оцінювати внесок ознак у формування рішення, що сприяє прозорості та довірі до моделі. Підходи до зниження розмірності (t-SNE, UMAP) та подальші метрики оцінювання кластеризації забезпечують інструменти для візуального аналізу структур даних та кількісного оцінювання якості їхнього групування. Сукупність цих методів утворює комплексний апарат для створення інтерпретованих і надійних систем виявлення та аналізу фейкової інформації.

2.3 Інформаційна структура системи за методом інтерпретування результатів виявлення фейкових новин

2.3.1 Проєктування структурної схеми та взаємодії компонентів

Відповідно до розробленого методу, була спроєктована інформаційна структура системи, яка включає всі необхідні компоненти для реалізації її функціональних можливостей. Водночас важливою умовою є забезпечення узгодженої взаємодії між окремими модулями, що гарантує цілісність і стабільність роботи. З цією метою була побудована архітектурна схема системи, яка відображає основні компоненти та інформаційні потоки між ними (рисунок 2.3).

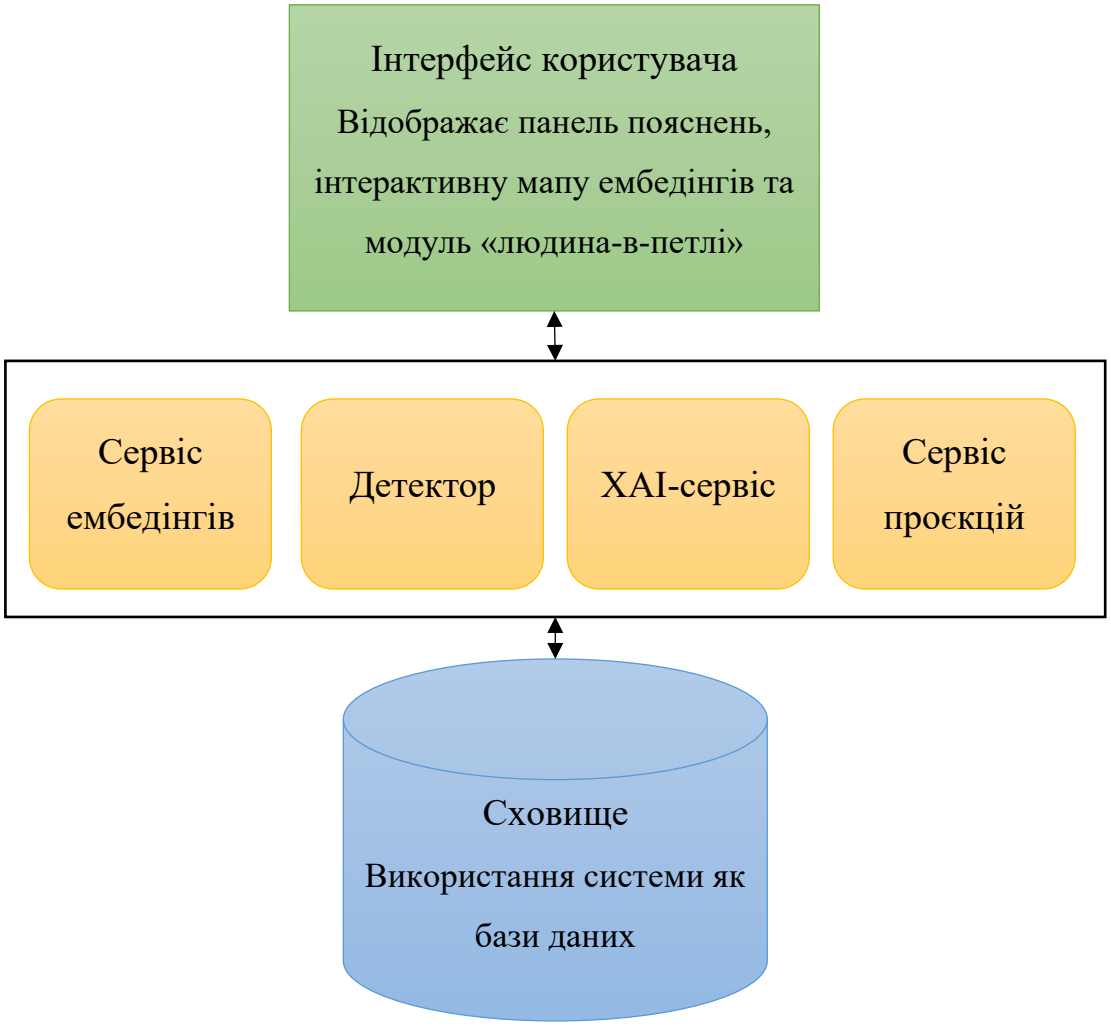


Рисунок 2.3 – Архітектурна схема системи

Система має трирівневу структуру, що включає веб-інтерфейс (UI), набір спеціалізованих сервісів та централізоване сховище даних.

Взаємодія з користувачем реалізується через веб-інтерфейс, який забезпечує можливість завантаження текстів новин, перегляду результатів класифікації, отримання інтерпретаційних пояснень, аналізу візуалізацій та надання зворотного зв’язку. Кожна із зазначених функцій реалізується за допомогою відповідних сервісів, що послідовно виконують обчислювальні завдання та передають результати в інтерфейс. Компонент «сервіси» включає чотири ключові підсистеми:

– детектор (класифікаційний API), який здійснює автоматичне віднесення текстів новин до визначених класів (наприклад, «фейк» чи «не фейк»);

– сервіс ембедінгів, що функціонує на основі моделі Sentence-BERT і формує векторні подання текстів. Ці подання зберігаються у сховищі та використовуються для пошуку подібних документів, кластеризації та побудови візуалізацій;

– XAI-сервіс, призначений для забезпечення інтерпретації результатів класифікації. Він реалізує методи SHAP, IG та LIME, формуючи інтерпретації, які зберігаються у базі даних і можуть бути проаналізовані користувачем;

– сервіс проєкцій, що використовує методи UMAP та t-SNE для зниження розмірності векторних представлень та побудови візуалізацій, які дають змогу досліджувати внутрішню структуру корпусу.

Усі зазначені сервіси взаємодіють із централізованим сховищем даних, яке є ключовим елементом інформаційної структури системи. У сховищі зберігаються тексти новин, відповідні мітки та метадані, векторні подання та їх унікальні ідентифікатори, пояснення, сформовані XAI-сервісом, координати візуалізацій, а також зворотний зв’язок від користувачів.

Для кращого розуміння функціонування системи з реалізованим методом була побудована діаграма активності, яка ілюструє послідовність дій користувача та внутрішніх процесів системи під час отримання пояснення результатів класифікації новини (рисунок 2.4). Процес починається з вибору або завантаження новини у веб-інтерфейсі, після чого формується запит до класифікаційного сервісу. Якщо у базі даних уже збережені обчислені ембедінги та прогноз, вони завантажуються; у протилежному випадку модель генерує їх і заносить до сховища. Далі викликається XAI-сервіс, який із використанням методів SHAP, IG або LIME, формує інтерпретаційні пояснення. Результати повертаються у форматі JSON і відображаються у веб-інтерфейсі у вигляді ключових слів, концептів чи візуальних графів.

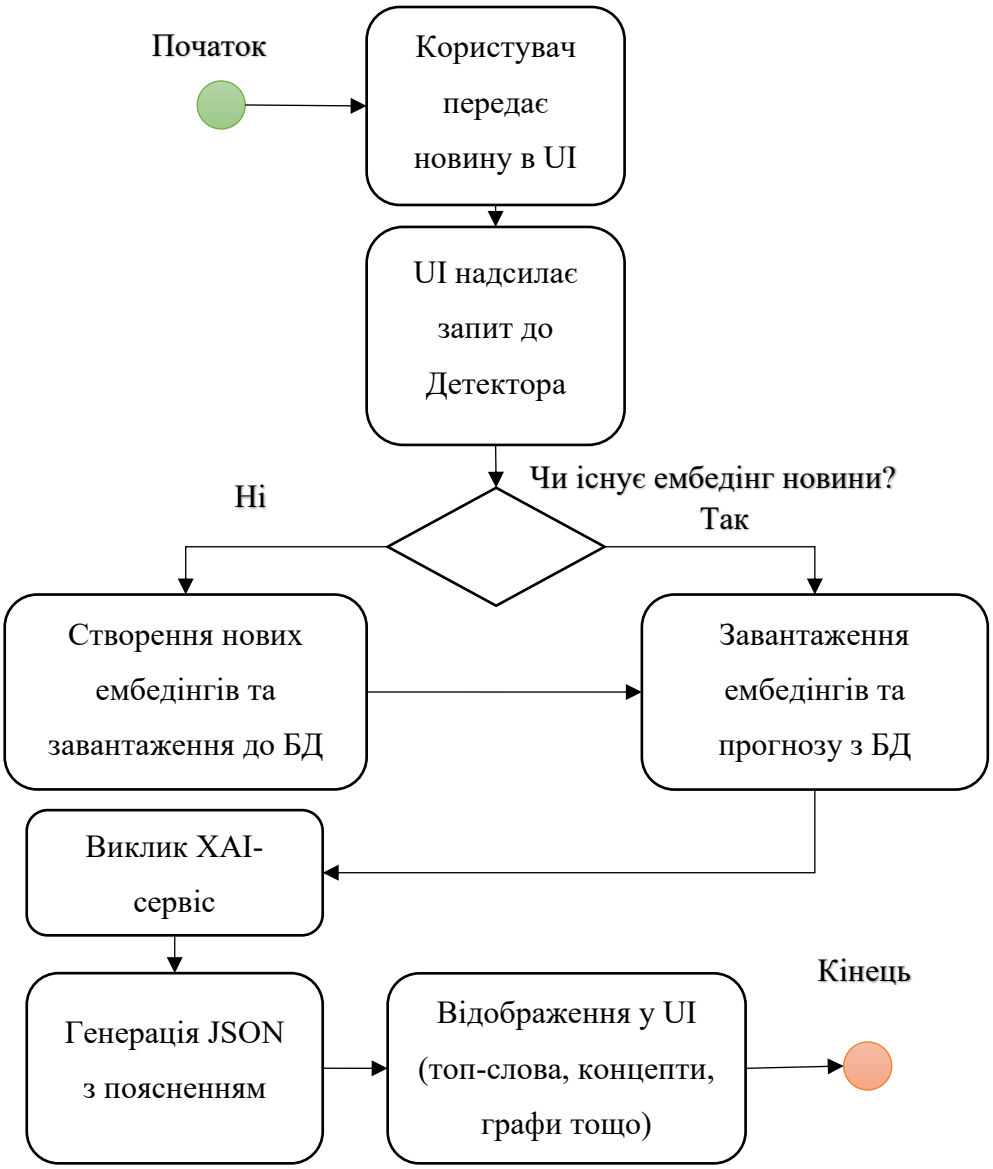


Рисунок 2.4 – Діаграма активності «Перегляд пояснень для окремої новини»

Поряд із діаграмою активності, що відображає процес отримання пояснень, доцільним є подання діаграми, яка ілюструє роботу модуля «людина-в-петлі» (рисунок 2.5). У контексті даної роботи процес зосереджений на взаємодії користувача з параметрами моделі: після аналізу результатів навчання або візуалізацій користувач обирає нові значення параметрів та ініціює повторне тренування моделі. Ці зміни формуються у вигляді JSON-запиту, який передається до сервісу навчання та заноситься до бази даних як окрема подія зміни конфігурації. Зібраний фідбек надалі використовується для оновлення навчальних даних і ознак у межах циклу «людина-в-петлі».

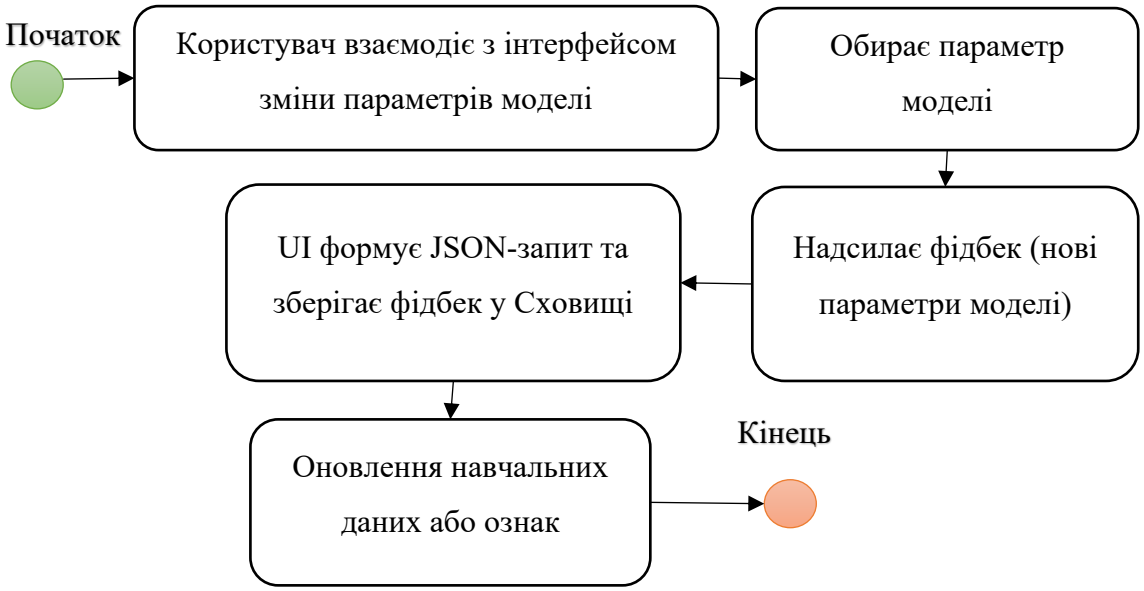


Рисунок 2.5 – Діаграма активності «Фідбек на параметр»

Розроблена проєктна архітектура системи забезпечує логічний поділ на підсистеми та чітко визначає їхню взаємодію через стандартизовані інтерфейси. Схема компонентів та активностей демонструє, як дані проходять через усі етапи обробки – від збору та збереження в сховищі до пояснюваної класифікації та інтерактивного відображення результатів. Така архітектура гарантує масштабованість системи, її прозорість та інтегрованість усіх сервісів у єдине середовище.

2.3.2 Проєктування структури бази даних та опис сутностей

Оскільки всі дані, що обробляє система, повинні зберігатися у централізованому сховищі, було спроєктовано структуру бази даних. Для цього побудовано реляційну діаграму (рисунок 2.6), яка відображає логічні сутності, їх атрибути та взаємозв’язки.

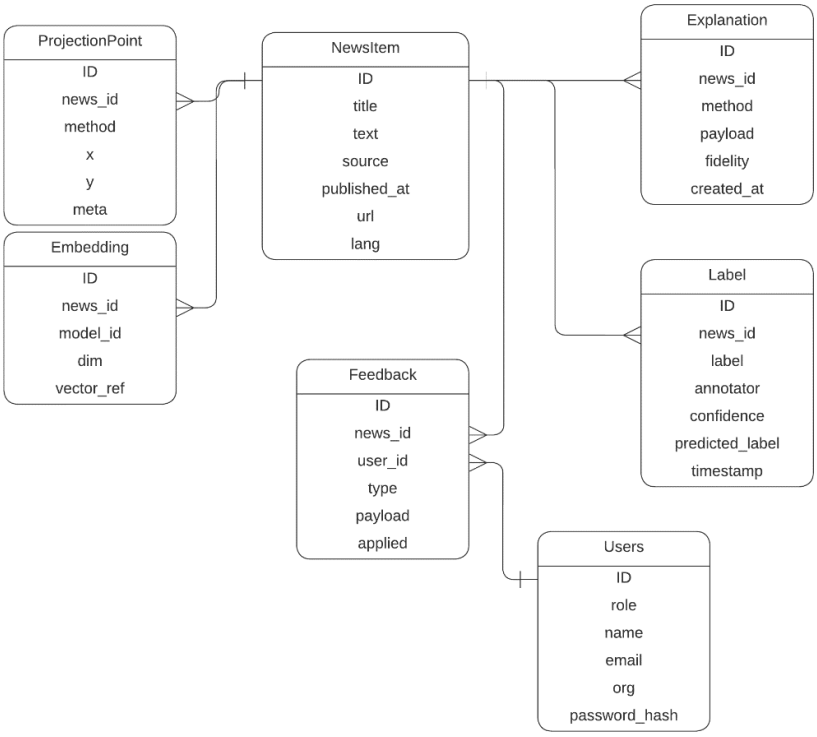


Рисунок 2.6 – Реляційна діаграма бази даних

Щоб надати більш формалізований опис, у таблиці 2.2 подано перелік полів та їх опис для головної таблиці «NewsItem». Вона зберігає основну інформацію про новини та може бути адаптована залежно від обраного набору даних.

Таблиця 2.2. Опис атрибутів головної таблиці NewsItem

Атрибут	**Тип даних**	**Опис**
ID	bigint unsigned	Унікальний первинний ключ
title	text	Заголовок новини
text	text	Повний текст новини
source	varchar	Джерело публікації
published_at	timestamp	Дата та час публікації
url	varchar	Посилання на оригінал
lang	varchar	Мова тексту

Допоміжні таблиці, такі як «Label», «Embedding», «Explanation», «ProjectionPoint» та «Feedback», відображають роботу основних компонентів методу, тобто сервісів, що реалізують відповідні етапи обробки.

Таблиця «Label» відповідає за збереження інформації щодо анотацій і міток. Вона містить такі ключові атрибути: ID – первинний унікальний ключ; news_id – зовнішній ключ, що вказує на новину, до якої відноситься мітка; label, який зберігає істинну категорію або клас новини; confidence як рівень впевненості у правильності прогнозованої мітки; predicted_label як спрогнозована мітка моделлю; час та опис створеної мітки. Дана таблиця безпосередньо використовується сервісом класифікації.

Важливе значення має також таблиця «Feedback», що застосовується у циклі «людина-в-петлі». Вона містить посилання на новину та користувача, який зробив корекцію, а також зберігає деталі внесених змін і позначку про те, чи враховано цей фідбек під час подальшого навчання моделі.

Інші три таблиці забезпечують збереження результатів роботи сервісів, що реалізують різні етапи методу. Таблиця «Embedding» містить дані, пов’язані зі створенням векторних представлень текстів. У ній зберігається ідентифікатор новини, параметри побудови ембедінга (наприклад, розмірність вектора), а також шлях до збереженого векторного представлення у файловій системі чи сховищі. Наступна таблиця «Explanation» відповідає за збереження результатів пояснювальних методів. Вона включає інформацію про використаний алгоритм пояснення (наприклад, SHAP чи Integrated Gradients), структуровані результати пояснення (топ-слова, їхні ваги/важливості), а також метрику якості чи узгодженості пояснення з прогнозом моделі. І остання таблиця «ProjectionPoint» використовується для роботи з результатами візуалізації простору ембедінгів. У ній зберігаються координати точок після зниження розмірності (наприклад, методом UMAP або t-SNE), ідентифікатор відповідного методу проєкції та метадані, що дають змогу відновити карту в інтерактивному вигляді.

Запропонована структура є достатньою для реалізації всіх основних функцій системи. Вона охоплює кожен ключовий аспект обробки новин: від

зберігання сирих текстів і метаданих до пояснень рішень моделі. Завдяки такій структурі не лише зберігаються тексти та результати їхньої обробки, а і є можливість підтримувати цикл «людина-в-петлі», коли користувач може впливати на систему, надаючи нові анотації чи коригуючи пояснення. Логічні зв’язки між сутностями забезпечують цілісну інтеграцію всіх підсистем, що дає змогу підтримувати повний життєвий цикл аналізу даних – від завантаження корпусу до пояснення та перевалідації прогнозів.

Висновки до розділу 2

У другому розділі магістерської кваліфікаційної роботи подано проєктування методу та архітектури вебзастосунку для інтерпретації результатів виявлення фейкових новин на основі великої мовної моделі. Запропонований метод структуровано у шість послідовних кроків, що охоплюють повний цикл обробки новинного тексту: від попередньої підготовки даних до розрахунку класифікаційних метрик і формування локальних та глобальних інтерпретацій роботи моделей.

Створена архітектура системи включає три ключові компоненти: модуль відображення результатів, обчислювальний сервіс і підсистему зберігання даних. Для бази даних спроєктовано реляційну структуру, яка забезпечує збереження всіх необхідних сутностей – текстів новин, метаданих, міток, ембедингів, інтерпретацій, а також інформації про користувачів і наданий ними зворотний зв’язок, що може бути використаний для подальшого вдосконалення моделей. Для реалізації архітектури можна застосувати мікросервісний підхід, де кожний сервіс виконує чітко визначену функцію та може розгортатися й масштабуватися незалежно від інших компонентів.

РОЗДІЛ 3 Програмна реалізація інформаційної системи виявлення фейкових новин у вигляді вебзастосунку

3.1 Вибір технологічного стеку для програмної реалізації системи

Для забезпечення коректної роботи всіх підсистем необхідно обрати відповідні засоби розробки, які б задовольняли вимоги щодо продуктивності, масштабованості та підтримки сучасних стандартів.

Серед поширених платформ для побудови серверної логіки можна виділити ASP.NET Core, Spring Boot, а також веб-фреймворки на мові Python – Django та FastAPI. Кожен із зазначених інструментів має власні переваги та обмеження, тому для обґрунтованого вибору проведено їх порівняння за кількома ключовими критеріями: продуктивність REST API, підтримка контейнеризації та кросплатформності. Узагальнені результати подано у таблиці 3.1.

Таблиця 3.1 – Порівняння інструментів для бекенду

Технологія	**Продуктивність**	**Контейнери**	**Складність**	**Коментар**
ASP.NET Core	Висока	Docker support	Середня	Добре підходить для високонавантажених REST API
Spring Boot	Висока	Docker support	Висока (Java stack)	Надійний, але важчий у деплої
Django	Середня	Docker support	Середня	Має ORM, але менш продуктивний
FastAPI	Висока	Docker support	Середня	Сучасний, зручний для мікросервісів

Серед розглянутих платформ для бекенд-частини системи було обрано ASP.NET Core, оскільки цей фреймворк характеризується високою продуктивністю, нативною підтримкою асинхронних викликів та доброю масштабованістю. Водночас для реалізації компонентів, пов’язаних із роботою

моделей трансформерів та методів інтерпретації, доцільно використовувати Python. Це зумовлено наявністю великої кількості спеціалізованих бібліотек, активною спільнотою розробників та простотою інтеграції з іншими сервісами за допомогою REST або gRPC.

Щодо клієнтської частини, для розробки інтерфейсу користувача було розглянуто чотири популярні фреймворки: React, Blazor, Angular та Vue.js. Для їх порівняння було визначено основні критерії вибору, поміж яких: рівень інтерактивності, зручність інтеграції з обраним бекендом, підтримка візуалізаційних механізмів (Canvas/WebGL), а також складність реалізації інтерфейсу (таблиця 3.2).

Таблиця 3.2 – Порівняння інструментів для фронтенду

Технологія	**Продуктивність**	**Інтеграція з API**	**Візуалізація**	**Складність**
React	Висока	Проста	Чудова підтримка через D3.js, three.js тощо	Середня
Blazor	Середня	Добра	Обмежена (через interop)	Середня
Angular	Висока	Добра	Є бібліотеки, але складніші	Висока
Vue.js	Середня	Добра	Є плагіни, легше ніж Angular	Низька

Поміж проаналізованих фреймворків для фронтенду було обрано React, оскільки він відзначається високою масштабованістю, широкою спільнотою користувачів та багатою екосистемою бібліотек для роботи з інтерактивними візуалізаціями (зокрема, D3.js, Recharts). Крім того, React забезпечує зручну інтеграцію з ASP.NET Core API, що дає змогу ефективно організувати обмін даними між клієнтською та серверною частинами системи.

Останнім, але не менш важливим компонентом архітектури є вибір СКБД для організації надійного сховища. Для порівняння були розглянуті три поширені рішення: PostgreSQL, MySQL та MongoDB (таблиця 3.3). Основними критеріями відбору є здатність СКБД ефективно працювати з текстовими даними, підтримка складних зв'язків між сутностями, а також можливість масштабування у випадку зростання обсягів даних.

Таблиця 3.3 – Порівняння систем зберігання

СКБД	**Тип**	**Переваги**	**Обмеження**
PostgreSQL	Реляційна	Сильні зв'язки, JSONB, підтримка NLP-розширень	Трохи складна конфігурація
MySQL	Реляційна	Простота, поширеність	Обмеженіша робота з JSON
MongoDB	Документна	Гнучкі JSON документи	Слабкі зв'язки, денормалізація

Для забезпечення коректної роботи всіх підсистем та враховуючи переваги гібридного підходу до зберігання даних, було обрано PostgreSQL. Дана СКБД поєднує класичну реляційну модель із підтримкою типу даних JSONB, що дає змогу одночасно зберігати структуровані метадані та напівструктуровану інформацію, зокрема пояснення моделей чи векторні подання текстів.

Отже, обрана комбінація інструментів забезпечує баланс між продуктивністю (ASP.NET Core + React), гнучкістю для ML-моделей (Python), інтерпретованістю (SHAP/IG/LIME) та простотою розгортання (Docker Compose). Це дає змогу реалізувати всі ключові вимоги системи, зберігаючи можливість масштабування у майбутньому, наприклад через Kubernetes.

3.2 Програмна реалізація компонентів вебзастосунку та організація їхньої взаємодії

Реалізований програмний комплекс побудований за мікросервісною архітектурою та включає чотири основні логічні компоненти: клієнтський застосунок (UI), серверну частину (Backend API), ML-сервіс, що відповідає за обробку даних і машинне навчання, а також сховище даних на основі PostgreSQL. Узгоджену роботу всіх компонентів забезпечує Docker-інфраструктура, яка автоматизує процес запуску, контейнеризації та мережевої взаємодії між сервісами. Такий підхід гарантує ізольованість середовищ, стабільність роботи та можливість масштабування системи.

Кожен компонент виконує чітко визначені функції, а загальна структура системи представлена на діаграмі (рисунок 3.1), що демонструє взаємозв'язки між ключовими модулями й допоміжними підсистемами.

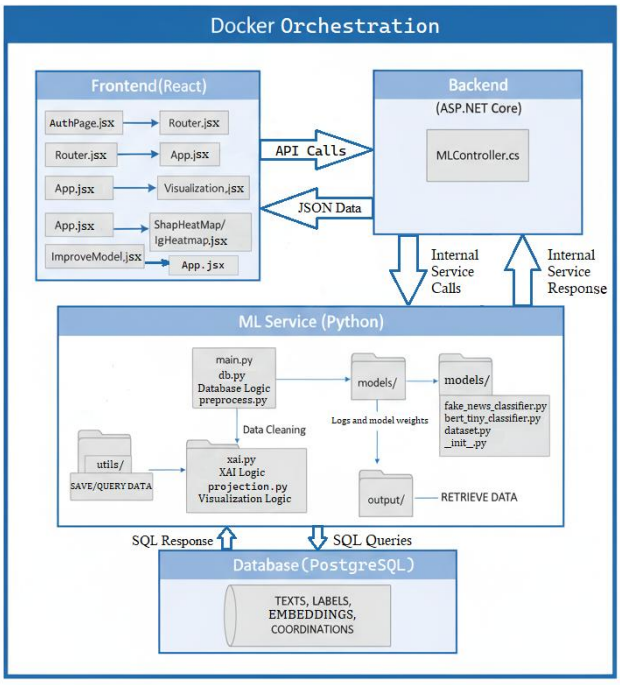


Рисунок 3.1 – Архітектура програмної системи на основі Docker-оркестрації

Обмін даними між частинами вебзастосунку здійснюється через REST API та прямі запити ML-сервісу до бази даних. Це забезпечує узгоджений потік

інформації, стабільність обчислень та розділення відповідальностей між програмними модулями.

Клієнтська частина реалізована на основі React і відповідає за взаємодію користувача з системою. UI надсилає запити на Backend API через REST-інтерфейс та відображає отримані дані у вигляді графіків, таблиць, текстових пояснень та інших візуальних компонентів. Архітектура фронтенду охоплює такі ключові частини:

– Router.jsx забезпечує маршрутизацію та навігацію між сторінками вебзастосунку;

– AuthPage.jsx керує процесами реєстрації та авторизації користувачів;

– App.jsx – головний компонент, що агрегує функціонал візуалізацій, пояснень результатів класифікації, а також інструменти покращення моделі;

– Visualization.jsx відповідає за відображення візуалізацій даних (t-SNE, UMAP);

– ShapHeatMap/IgHeatmap/LimeHeatmap.jsx – компоненти, що відповідають за виведення пояснень для окремих текстів;

– ImproveModel.jsx надає інтерфейс для зміни параметрів моделі та додавання нових даних до навчального набору.

Backend створено на основі ASP.NET Core, який виконує роль проміжного шару між UI та ML-сервісом. Основним елементом цього шару є MLController.cs – універсальний контролер, що приймає всі HTTP-запити з фронтенду, маршрутизує їх до відповідних ендпоїнтів ML-сервісу та повертає сформовану відповідь клієнту. Серверна частина не містить логіки машинного навчання, її задача полягає у забезпеченні безпечної, типізованої та надійної комунікації між клієнтом та обчислювальною частиною системи.

ML-сервіс є центральною логічною частиною комплексу. Він відповідає за обробку даних, генерацію ембедінгів, навчання моделей, класифікацію текстів, побудову пояснень, створення візуалізацій тощо. У його структурі виділяються такі основні компоненти:

– db.py керує взаємодією з PostgreSQL: зберіганням текстів, міток, ембедінгів, координат для візуалізацій та XAI-даних тощо;

– preprocess.py виконує очищення, фільтрацію, нормалізацію та балансування класів;

– main.py – центральний модуль, що описує всі REST-ендпоїнти ML-сервісу та координує виклики до інших підмодулів;

– models/ – директорія з реалізаціями моделей машинного навчання. Містить такі файли як: fake_news_classifier.py, що реалізує базову модель (SentenceBERT для ембедінгів + логістична регресія для класифікації), bert_tiny_classifier.py містить логіку для складнішої моделі (DistilBERT), __init__.py забезпечує коректну ініціалізацію моделей та dataset.py виконує додаткову обробку набору даних, зокрема для DistilBERT;

– utils/ – директорія для утиліт, включає такі два файли – xai.py, що містить логіку для створення пояснень (SHAP, градієнти) та projection.py, що реалізує методи візуалізації (t-SNE, UMAP);

– output/ – каталог для збереження навчених моделей, ваг, пояснень, графіків та інших артефактів.

ML-сервіс отримує запити від Backend API, виконує обчислення або звертається до бази даних, формує відповідь та повертає її назад на серверну частину.

У ролі сховища даних виступає така СКБД як PostgreSQL, яка зберігає всі необхідні дані для функціонування системи, починаючи від текстів новин та їх міток до ембедінгів текстів, координат для візуалізацій, SHAP/IG/LIME дані та даних користувача. ML-сервіс безпосередньо взаємодіє з базою для читання, оновлення і збереження даних, забезпечуючи узгодженість та структурованість збереженої інформації.

Docker виступає важливим елементом системи, оскільки дає змогу об’єднати фронтенд, бекенд, ML-сервіс і базу даних у єдиний контейнеризований застосунок. Docker Compose автоматизує запуск усіх компонентів, забезпечує

їхню ізоляцію, керування мережевими ресурсами та спрощує розгортання системи на будь-якому середовищі.

Для кращого розуміння принципів роботи та взаємодії API між основними частинами системи, на рисунку 3.2 наведено схему, що детально ілюструє повний шлях обробки запиту – від моменту його ініціації користувачем на фронтенді до отримання кінцевого результату. На схемі відображено послідовність передавання даних між компонентами, а також логіку викликів, що забезпечують узгоджену роботу фронтенду, бекенду, ML-сервісу та бази даних у межах контейнеризованого середовища Docker.

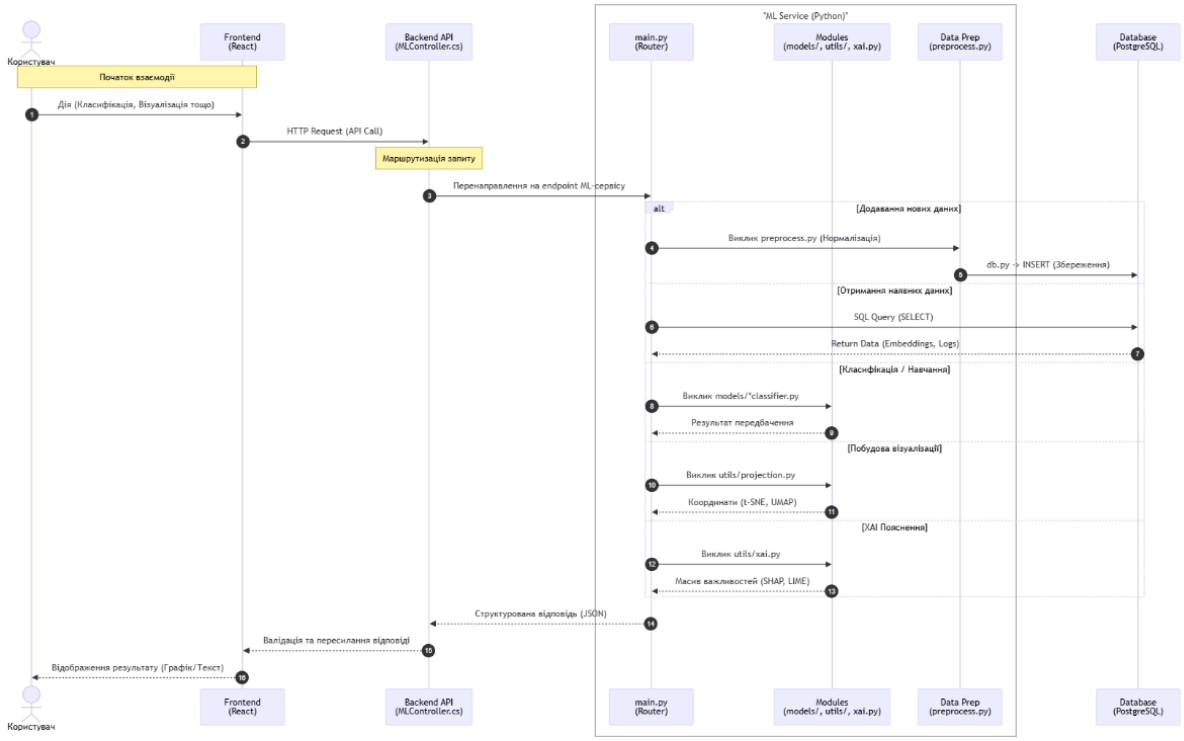


Рисунок 3.2 – Діаграма послідовності взаємодії мікросервісів при обробці запиту користувача

Послідовність взаємодії між компонентами системи реалізована через ланцюжок REST API-викликів, що починається на клієнтському рівні. Користувач здійснює дію у вебінтерфейсі (класифікація тексту, генерація візуалізації, перегляд пояснення, додавання нових даних тощо), після чого фронтенд формує відповідний HTTP-запит і надсилає його до Backend API. На серверній частині запит обробляється контролером MLController.cs, який виконує роль

маршрутизатора: він визначає тип запиту та перенаправляє його на відповідний ендпоїнт ML-сервісу. Подальша логіка обробки запиту виконується безпосередньо в ML-сервісі.

Обробка запиту в ML-сервісі включає кілька послідовних етапів:

– центральний модуль main.py отримує HTTP-запит та ініціює відповідний робочий сценарій;

– якщо запит передбачає додавання нових прикладів до навчального набору, main.py викликає preprocess.py для очищення та нормалізації даних, а після цього – db.py для їх збереження у PostgreSQL;

– якщо запит стосується отримання наявних даних (візуалізацій, пояснень, ембедінгів тощо), main.py формує SQL-запит до бази даних та повертає отримані результати;

– запити, пов’язані з класифікацією або навчанням моделей, передаються до модулів models/ (fake_news_classifier.py, bert_tiny_classifier.py);

– запити, що стосуються побудови візуалізацій, обробляються через utils/projection.py (t-SNE, UMAP);

– формування пояснень для XAI-методів (SHAP, IG, LIME) здійснюється за допомогою utils/xai.py.

Після завершення обчислень ML-сервіс формує структуровану відповідь (результат класифікації, масив координат для візуалізації, вектор важливостей слів тощо) та повертає її Backend API. Бекенд приймає цю відповідь, здійснює мінімальну обробку або валідацію та пересилає дані назад до фронтенду. На завершальному етапі клієнтський застосунок відображає результат користувачеві у відповідному форматі – графіку, heatmap-пояснення, таблиці параметрів або текстовому повідомленні.

У підсумку, на основі розробленої архітектури створено цілісний програмний комплекс, у якому клієнтська частина, сервер, модуль машинного навчання та база даних функціонують як узгоджена система в межах контейнеризованого середовища Docker. Така архітектура забезпечує оптимальні

умови для реалізації механізмів інтерпретації результатів виявлення фейкових новин, отриманих за допомогою LLM.

3.3 Алгоритмічна реалізація методів машинного навчання та інтерпретації

Реалізація системи ґрунтується на інтеграції класичних моделей машинного навчання, сучасних підходів до побудови текстових векторних представлень (ембедінгів), а також методів XAI, що дають змогу інтерпретувати результати роботи класифікаторів. Архітектурно програмний комплекс побудований за об’єктно-орієнтованим принципом, де кожна модель (логістична регресія та DistilBERT) інкапсульована у власний клас, що реалізує інтерфейси навчання, прогнозування та збереження метрик. Окремий модуль відповідає за генерацію пояснень. Нижче подано опис ключових алгоритмів, принципів організації обчислень та UML-діаграми класів (рисунки 3.3–3.4), що демонструють структуру реалізованого програмного забезпечення.

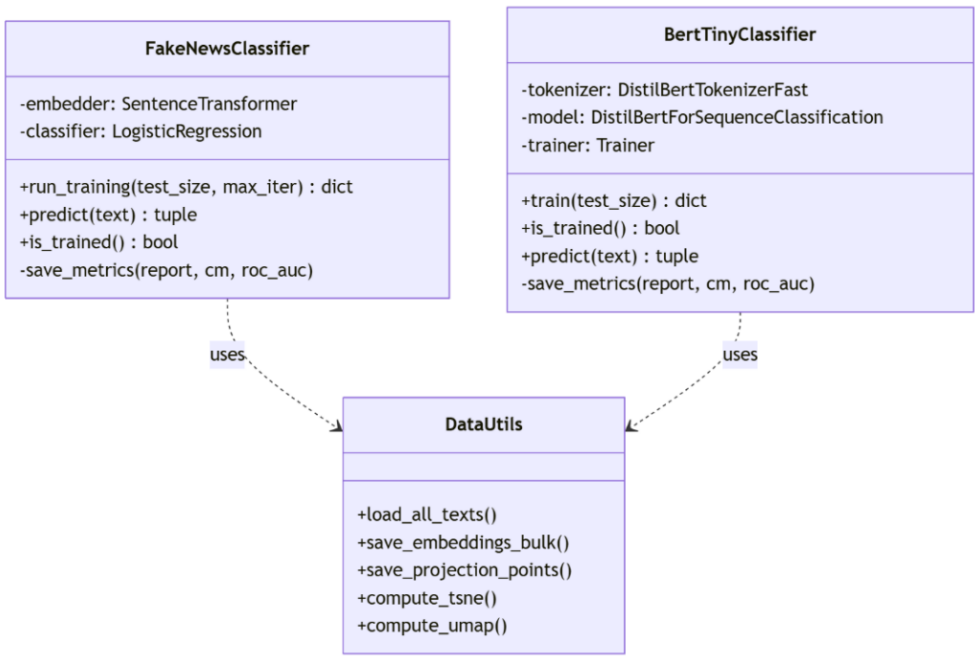


Рисунок 3.3 – Діаграма класів модуля навчання

Рисунок 3.4 – Структура модуля генерації пояснень

Ключовим етапом обробки даних є перетворення текстів у числовий простір ознак. Для базової моделі використано архітектуру Sentence-BERT (модифікація all-MiniLM-L6-v2). Вона перетворює вхідний текст у вектор фіксованої довжини (d=384), де семантично схожі тексти мають меншу косинусну відстань. Ембедінги після обчислення зберігаються у базу даних для повторного використання, що допомагає уникнути повторних обчислень при візуалізації або інкрементальному донавчанні. Разом із тим, підхід, заснований на Sentence-BERT, має низку обмежень. По-перше, якість моделей, що працюють з такими ембедінгами, повністю залежить від обраної семантичної моделі, а її зміна вимагає повного перерахунку векторів. По-друге, Sentence-BERT має обмеження за довжиною вхідної послідовності (приблизно 256–512 токенів), що може спричинити втрату частини інформації у дуже довгих текстах. По-третє, обчислення ембедінгів для великого корпусу новин може бути суттєво ресурсомістким, особливо при відсутності GPU. Попри легкість моделі all-MiniLM-L6-v2, для масштабних датасетів часто виникає потреба у пакетній обробці або кешуванні результатів.

Логіка навчання моделі реалізована через два основні класи: FakeNewsClassifier, що відповідає за підхід Sentence-BERT + Logistic Regression, та BertTinyClassifier, який реалізує fine-tuning моделі DistilBERT.

У випадку класичного методу логістична регресія навчається на матриці ембедінгів X, отриманих із Sentence-BERT, у поєднанні з відповідними мітками y. Для компенсації дисбалансу між класами застосовуються збалансовані ваги (class_weight="balanced"), що дає змогу моделі краще працювати при нерівномірному розподілі фейкових і нефейкових новин. Для отримання ймовірнісних оцінок у моделі використовується метод predict_proba() класу sklearn.linear_model.LogisticRegression, що забезпечує коректні значення довіри моделі. Такий підхід характеризується низькою обчислювальною складністю – порядку O(N·d), де d = 384 це розмірність Sentence-BERT ембедінгів. Це дає змогу моделі тренуватися дуже швидко навіть на звичайному CPU, а обсяг пам’яті, необхідний для зберігання ваг, залишається мінімальним. Водночас як і всі лінійні моделі, логістична регресія не здатна повноцінно враховувати складні нелінійні залежності в текстах. Це може обмежувати кінцеву якість класифікації, однак для задач з високовимірними векторними представленнями вона залишається стабільною та передбачуваною базовою моделлю.

Другий підхід базується на донавчанні попередньо тренованої трансформерної моделі distilbert-base-uncased, реалізованої через бібліотеку Hugging Face Transformers. Використання оптимізатора AdamW та оновлення всіх параметрів моделі дає змогу DistilBERT адаптуватися до конкретних мовних патернів, характерних для корпусів фейкових новин. На відміну від підходу з попередньо обчисленими ембедінгами, fine-tuning дає змогу глибшій моделі засвоювати як семантичні, так і контекстуальні залежності, які можуть бути втрачені при простій векторизації. Проте цей підхід значно вимогливіший з обчислювального погляду. Обчислювальна складність fine-tuning’у масштабується приблизно як $O(N \cdot L \cdot H^2)$, де L це кількість шарів, а H – розмір прихованого представлення. Це суттєво більше, ніж у класичних алгоритмів, але дає змогу досягати значно кращої якості класифікації завдяки моделюванню

нелінійних структур у тексті. Після завершення навчання система автоматично будує основні діагностичні візуалізації – матрицю помилок та ROC-криву, що дає змогу оцінити якість моделі за різними аспектами.

Для забезпечення інтерпретованості рішень моделі в системі реалізовано клас ExplainableModel, який надає уніфікований інтерфейс до методів SHAP, LIME та Integrated Gradients.

Метод SHAP ґрунтується на теорії кооперативних ігор та дає змогу оцінити внесок кожної ознаки у формування фінального передбачення. Для логістичної регресії використовується LinearExplainer, який аналізує вплив окремих компонентів вектора ембедінгів. Оскільки виміри ембедінгу не мають безпосереднього лінгвістичного значення, SHAP забезпечує радше глобальну інтерпретацію того, які латентні ознаки Sentence-BERT найбільш суттєво впливають на модель. Для трансформерної моделі DistilBERT застосовується shap.Explainer з текстовим маскером, який послідовно приховує або модифікує токени у вхідному тексті. Це дає змогу оцінити вплив окремих слів на зміну ймовірності класу. У процесі побудови пояснення модель багаторазово переобчислює оцінки ймовірностей із різними варіантами тексту, що робить SHAP одним із найточніших, але найбільш обчислювально затратних методів для трансформерів.

Метод LIME у системі застосовується для моделі DistilBERT та генерує локальне лінійне наближення поведінки моделі у мікрооколі вибраного текстового прикладу. Алгоритм працює за принципом стохастичного збурення: він створює сотні варіацій тексту шляхом випадкового вимикання або маскування слів, отримує відповідні передбачення моделі, а потім навчає просту лінійну модель, яка апроксимує рішення DistilBERT у цій локальній області. Отже, LIME дає інтуїтивно зрозумілі пояснення на рівні токенів, демонструючи, які слова найбільше вплинули на передбачення. Він є менш стабільним порівняно з SHAP, але значно швидшим та більш придатним для інтерактивних інтерфейсів, особливо при низьких обчислювальних ресурсах.

Метод IG належить до градієнтних підходів та застосовується виключно до диференційовних моделей, що робить його непридатним для класичних алгоритмів машинного навчання, таких як логістична регресія. У межах цієї роботи IG реалізовано для DistilBERT, який дає змогу обчислювати похідні відносно вхідних токенів та їхніх ембедінгів. Метод explain_ig() інтегрує градієнти виходу моделі вздовж траєкторії від базового стану (нульовий або «порожній» ембедінг) до реального вектора вхідного тексту. Він дає змогу коректно оцінити важливість токенів без проблем, властивих "звичайним" градієнтам, таких як залежність від локальних шумів або ефект зникання градієнтів.

Для візуалізації високорозмірних ембедінгів у двовимірному просторі застосовано методи зниження розмірності t-SNE (клас sklearn.manifold.TSNE) та UMAP (клас umap.UMAP). Обидва методи добре відображають локальну структуру даних та дають змогу виділити природні кластери, однак мають різні властивості. Метод t-SNE забезпечує якісну локальну диференціацію, проте може спотворювати глобальні зв’язки, є чутливим до параметра perplexity та працює повільніше на великих наборах даних. UMAP, навпаки, краще зберігає як локальну, так і глобальну структуру, але його поведінка залежить від параметрів n_neighbors і min_dist, які визначають компроміс між щільністю кластерів та розподілом далеких точок. Обчислені координати проєкцій (x, y) додатково зберігаються у базі даних, що дає змогу клієнтській частині швидко завантажувати візуалізацію без повторних обчислень. У разі появи нових даних проєкції потребують повного перерахунку, оскільки t-SNE та звичайний UMAP не підтримують коректне інкрементальне оновлення.

У сукупності наведені компоненти формують гнучку та розширювану архітектуру, придатну як для обробки новинних текстів, так і для подальшого підсилення моделі сучасними трансформерними підходами та оптимізованими методами інтерпретації.

3.4 Перевірка коректності роботи компонентів програмної реалізації методу та аналіз результатів тестування

У межах цього підрозділу розглядаються функціональні сценарії тестування, спрямовані на підтвердження коректності роботи основних компонентів інформаційної системи та відповідності реалізованого функціоналу поставленим вимогам. Особлива увага приділяється тестуванню процесу реєстрації користувача у системі, навчання моделей та стабільному створенні пояснень рішень моделі.

Перший тест оформлено у вигляді тест-кейсу, що спрямований на перевірку базового функціоналу інформаційної системи – можливості успішної реєстрації нового користувача. Реєстрація є початковим етапом взаємодії з системою, тому коректність її виконання напряму впливає на доступ до інших сервісів та загальну працездатність платформи. У межах тесту перевіряється правильність обробки введених даних, створення запису в системі, формування токена доступу та перенаправлення користувача після завершення процедури. Результати тестування оформлено у вигляді покрокового сценарію (таблиця 3.4).

Таблиця 3.4 – Тест-кейс ТС-01

<table>
<tr><td>Тест-кейс ID: ТС-01</td><td>Пріоритет: 3</td><td>Створено: 10.11.2025, Вовк С. В.</td></tr>
<tr><td colspan="3">Назва: Реєстрація нового користувача.
Вхідні дані: Сервери доступні, база даних доступна, ім’я – «Stesiia», пошта – «test@gmail.com», пароль – «197214m», роль – «Дослідник».</td></tr>
<tr><td colspan="2">Кроки</td><td>Очікуваний результат</td></tr>
<tr><td colspan="2">1. Зайти за посиланням «http://localhost:3000».
2. Перейти на сторінку реєстрації.
3. .Заповнити всі обов’язкові поля: ім’я, пошту, пароль та обрати роль «Дослідник».
4. Натиснути на кнопку «Зареєструватися».</td><td>Система створює нового користувача, повертає статус 201 та токен доступу; перенаправлення на сторінку входу.</td></tr>
<tr><td colspan="3">Результат виконання тест-кейсу: пройдено успішно</td></tr>
</table>

Після виконання кроків, визначених у тест-кейсі ТС-01, до бази даних було успішно додано нового користувача з роллю «Дослідник» (рисунок 3.5). Система коректно сформувала необхідні дані для авторизації та автоматично перенаправила клієнта на сторінку входу, що підтверджує відповідність функціоналу вимогам та стабільність роботи модуля реєстрації.

id [PK] integer	role character varying (50)	name character varying (100)
1	user	Stefa
2	researcher	Stesiia

Рисунок 3.5 – Новий користувач у системі

Наступне тестування продемонстроване в таблиці 3.5, спрямоване на перевірку коректності повного циклу навчання базової моделі логістичної регресії: від завантаження корпусу даних до формування основних метрик якості, побудови графічних результатів та збереження версії моделі. Метою тесту є підтвердження того, що інформаційна система здатна безпомилково обробляти дані, запускати навчання, генерувати діагностичні візуалізації та повертати користувачу всі необхідні результати. Під час виконання сценарію, зазначеного у тест-кейсі ТС-02, користувач завантажив датасет LIAR, обробив його та ініціював навчання обраної моделі LogReg із параметром тестової вибірки 30%.

Таблиця 3.5 – Тест-кейс ТС-02

Тест-кейс ID: ТС-02	**Пріоритет:** 1	**Створено:** 11.11.2025, Вовк С. В.
Назва: Перевірити коректність навчання, побудови метрик та графіків. **Вхідні дані:** Сервери доступні, користувач авторизований, модель не тренована.		
Кроки		**Очікуваний результат**
1. Натиснути на кнопку «Обрати файли» та завантажити набір даних «LIAR».		

Продовження Таблиці 3.5 – Тест-кейс ТС-02

Кроки	Очікуваний результат
2. Натиснути на кнопку «Обробити дані». 3. Дочекатися повідомлення «Файли успішно надіслані!». 4. Обрати базову модель – Logistic Regression (через BERT-ембедінги). 5. Обрати значення параметру Test-size у 30%. 6. Натиснути на кнопку «Навчити модель». 7. Очікувати завершення навчання.	Модель тренується без помилок; формується: accuracy, precision, recall, F1; матриця помилок та ROC-крива; виводяться візуалізації (t-SNE та UMAP методів); зберігається версія моделі.
Результат виконання тест-кейсу: пройдено успішно	

По завершені навчання, система успішно побудувала ключові метрики – accuracy, precision, recall, F1, а також матрицю помилок і ROC-криву. Додатково сформовано візуалізації t-SNE та UMAP, що підтверджують коректність проєкції високовимірних ембедінгів (рисунок 3.6).

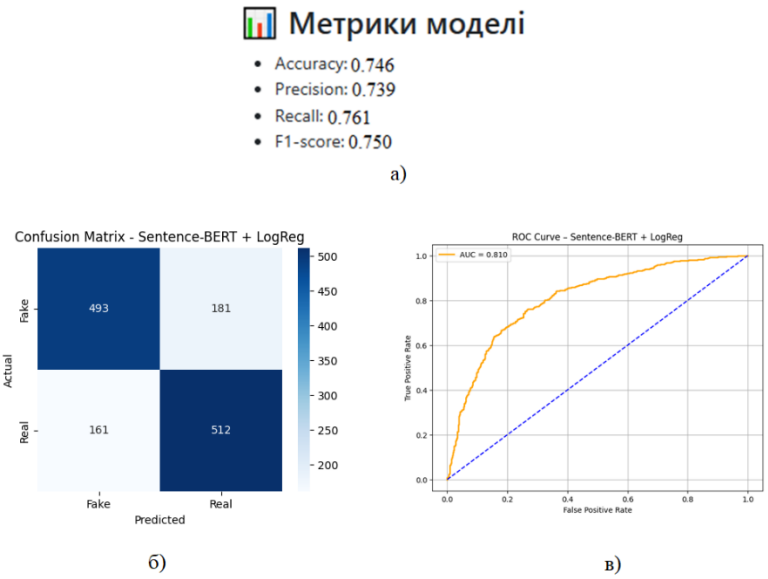


Рисунок 3.6 – Результати навчання базової моделі: а) метрики якості класифікації, б) матриця помилок та в) ROC-крива

Останнє тестування у форматі тест-кейс спрямоване на перевірку коректності роботи модуля інтерпретації, зокрема генерації локального пояснення за методом SHAP для вже навченої базової моделі (таблиця 3.6).

Таблиця 3.6 – Тест-кейс ТС-03

<table>
<tr><td>Тест-кейс ID: ТС-03</td><td>Пріоритет: 2</td><td>Створено: 13.11.2025, Вовк С. В.</td></tr>
<tr><td colspan="3">Назва: Генерація пояснення (SHAP) для базової моделі.
Вхідні дані: Сервери доступні, користувач авторизований, модель тренована.</td></tr>
<tr><td colspan="2">Кроки</td><td>Очікуваний результат</td></tr>
<tr><td colspan="2">1. Натиснути на кнопку «Рандомний прогноз».
2. Створюється пояснення для випадково обраної новини з бази даних.
3. Додавання створеного пояснення в базу даних.
4. Натиснути на кнопку «SHAP».</td><td>Повертається JSON з токенами та їх впливами; фронтенд відображає теплову підсвітку слів та графік топ-5 слів з їх впливом для обраної класу.</td></tr>
<tr><td colspan="3">Результат виконання тест-кейсу: пройдено успішно</td></tr>
</table>

У процесі виконання сценарію користувач ініціює створення випадкового прогнозу, після чого система автоматично обрала одну зі збережених новин та виконала для неї передбачення. Після цього модуль XAI сформував SHAP-пояснення у вигляді JSON-структури, що містить токени та числові значення їхнього впливу на вибір моделі.

Згенероване пояснення додано до бази даних, а фронтенд коректно відобразив отримані результати: теплову підсвітку слів у тексті та діаграму топ-5 найбільш впливових токенів для визначеного класу, що продемонстровано на рисунку 3.7.

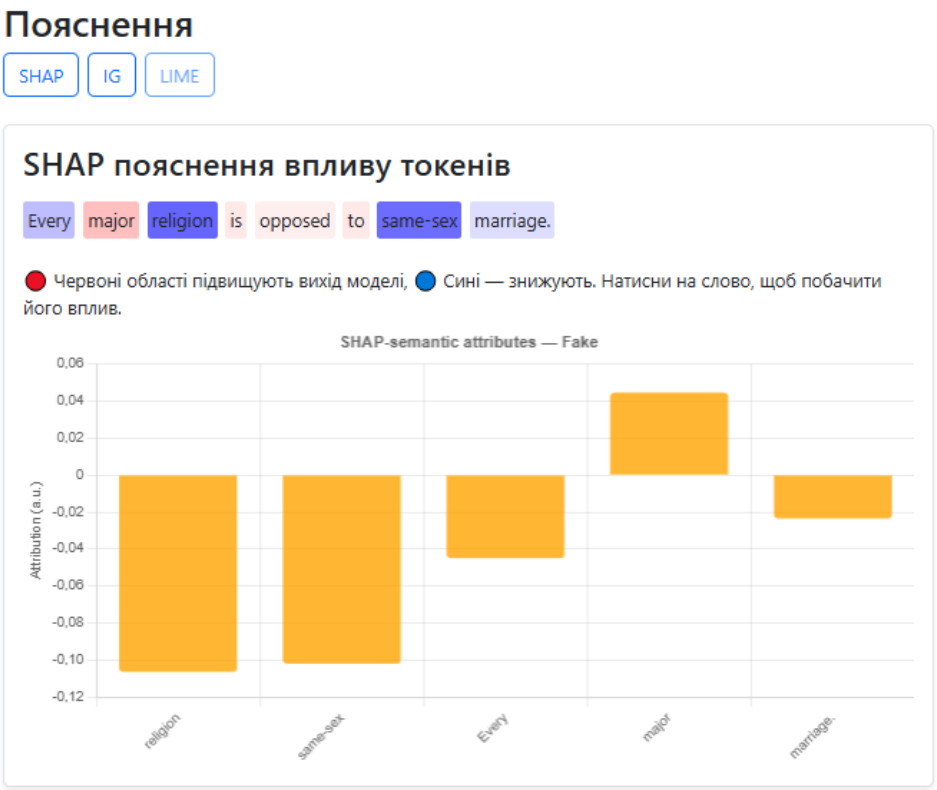


Рисунок 3.7 – Згенероване пояснення для базової моделі

Отже, проведене тестування основних функцій розробленої інформаційної системи, яка реалізує методи інтерпретації результатів класифікації фейкових новин, демонструє її функціональну повноту. Система надає користувачу можливість реєстрації та доступу до ключових модулів, включаючи обробку наборів даних, навчання моделей машинного навчання, створення пояснень для конкретних текстів новин, а також візуалізацію високорозмірних ембедінгів. Крім того, користувач може змінювати параметри моделей та здійснювати повторне навчання на нових даних, що забезпечує гнучкість та адаптивність системи під різні задачі аналізу новин.

Висновки до розділу 3

Отже, у роботі було обрано засоби для реалізації програмного продукту та продемонстровано прикладне використання розробленого методу інтерпретації результатів виявлення фейкових новин на основі великої мовної моделі у формі вебзастосунку. Через інтерфейс платформи користувач може зареєструватися,

обрати корпус даних, навчити модель, класифікувати окремі новини та аналізувати отримані результати. Система надає повний набір аналітичних інструментів: таблицю з метриками, матриці помилок, ROC-криві, а також локальні та глобальні інтерпретації роботи моделей у вигляді XAI-пояснень та візуалізацій векторного простору ознак, що допомагають краще зрозуміти процес прийняття рішень.

Архітектура рішення побудована на чотирьох мікросервісах, інтегрованих за допомогою Docker-оркестрації. Обчислювальний сервіс, що відповідає за роботу моделей і генерацію пояснень, реалізовано через низку класів, які забезпечують стабільність і коректне функціонування всіх компонентів. Його структура включає модулі для завантаження корпусів, управління моделями, обчислення ембедингів, формування локальних і глобальних інтерпретацій, а також API-шари для обміну даними з бекендом. Бекенд-сервіс, побудований на ASP.NET Core, виконує роль координатора між фронтендом та обчислювальним сервісом. Він обробляє запити користувачів, керує логікою авторизації та автентифікації, передає параметри для навчання та отримує результати моделювання. Фронтенд-сервіс, створений на основі React, забезпечує інтерфейс користувача. Він відображає метрики, таблиці, графіки, матриці помилок, інтерпретації моделей, а також надає можливість взаємодіяти з параметрами навчання, обирати корпуси даних, запускати класифікацію та переглядати пояснення для конкретних новин.

У вебзастосунку використано відповідні набори даних та попередньо навчені моделі, що дає змогу проводити повноцінну класифікацію новин і статей. Проведене прикладне тестування основних функцій системи підтвердило її працездатність: усі модулі працюють узгоджено, результати класифікації та інтерпретацій відповідають очікуваним.

РОЗДІЛ 4 Експериментальні дослідження та оцінювання методу

4.1 Характеристика наборів даних для проведення експериментів

Для тестування роботи реалізованого методу обрано два репрезентативні корпуси: LIAR [50] та FakeNewsNet [51] із підвибіркою GossipCop, яка містить значно більшу кількість повнотекстових новин для задач класифікації та подальшої кластеризації.

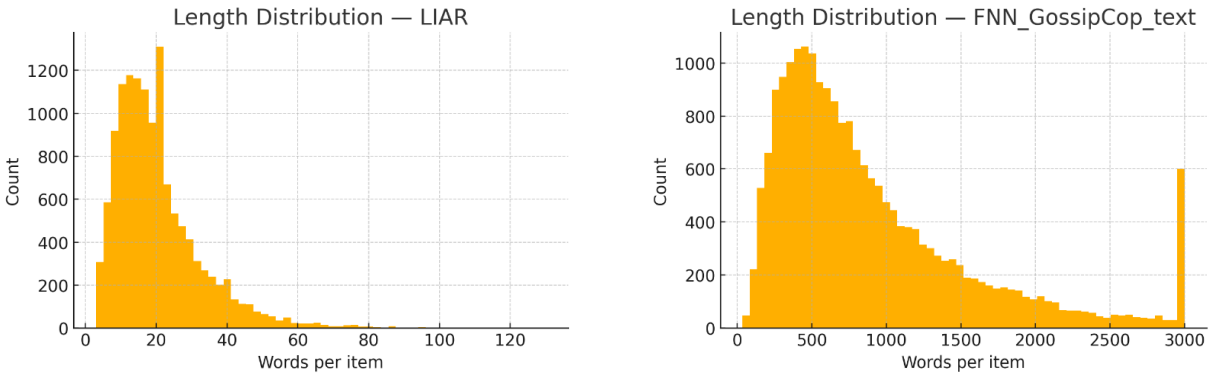
Корпус LIAR містить близько 12.8 тисяч коротких політичних висловлювань, зібраних із платформи PolitiFact. До кожного прикладу додаються супутні метадані – автор висловлювання, контекст, тематика та джерело інформації. Висловлювання оцінюються за шести класовою шкалою правдивості: pants-on-fire, false, barely-true, half-true, mostly-true, true. У межах дослідження застосовано саме бінаризацію, що дає змогу концентруватися на відмінності між правдивими та неправдивими твердженнями. Додатковою особливістю LIAR є надзвичайно короткі тексти: у середньому висловлювання містять 20–30 слів, рідше – до 50 слів, що зумовлює специфіку попередньої обробки та підходів до векторизації.

Іншим набором даних є FakeNewsNet, який складається з двох підвибірок – PolitiFact та GossipCop. Оскільки лише GossipCop містить достатню кількість повнотекстових новин для подальшого аналізу, у цьому дослідженні використано саме її. Кожен елемент даних подано у вигляді структури з чотирма полями: унікальний ідентифікатор новини, заголовок, основний текст та короткий опис. Хоча підвибірка не містить спеціального стовпця з мітками, сама структура директорій виконує роль джерела класів: новини розміщені у файлах HR (real) та HF (fake). У результаті даний набір новин складається із повнотекстових статей, середня довжина яких становить 300–500 слів, а окремі тексти можуть досягати кількох тисяч слів.

Для забезпечення коректності та відтворюваності експериментів обидва набори даних були поділені на тренувальну, валідаційну та тестову частини. Під час усіх запусків фіксується параметр random_state = 42, що гарантує сталість

розбиття та отриманих вибірок. Крім того, зберігаються всі артефакти: ембедінги, проміжні пояснення, координати низьковимірних проєкцій, що дає змогу відтворити повний хід експериментів.

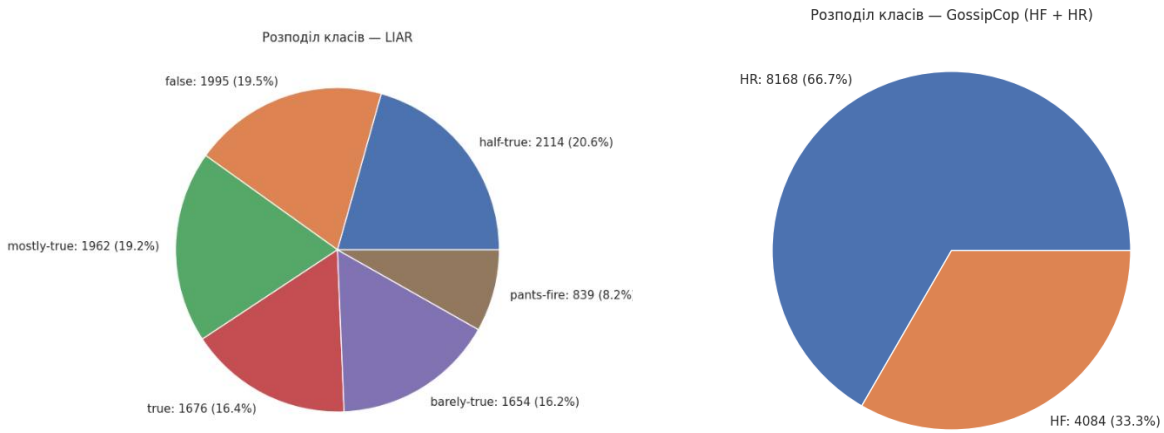
Додатково проведено аналіз довжин текстів. У наборі LIAR переважають висловлювання довжиною до 20 слів, однак трапляються і приклади на 5–50 слів. У GossipCop новини значно довші: середньо 300–500 слів, із максимальними значеннями до 3000 слів. Візуалізація цих відмінностей наведена на рисунку 4.1.



а) б)

Рисунок 4.1 – Порівняння розподілу довжин на різних корпусах: а) LIAR та б) FakeNewsNet (GossipCop)

Корпус LIAR характеризується нерівномірним розподілом класів: окремі категорії, такі як pants-on-fire або barely-true, менші за основні групи. У підвибірках FakeNewsNet, навпаки, спостерігається тенденція до переважання класу HR (real), що особливо помітно для GossipCop (Рисунок 4.2). Для коректної роботи моделей прийнято рішення виконати бінаризацію та балансування класів у LIAR.



а) б)

Рисунок 4.2 – Порівняння розподілу класів на різних корпусах: а) LIAR та б) FakeNewsNet (GossipCop)

Після балансування набору даних LIAR отримано рівні класи: 4488 “true” та 4488 “false”, загалом 8976 прикладів. Для підвибірки GossipCop також застосовано балансування, у результаті чого використовується 8168 текстів, порівну розподілених між класами HF та HR.

4.2 Порівняльний аналіз результативності моделей класифікації та дослідження впливу гіперпараметрів

Для оцінювання результативності запропонованого підходу обрано кілька моделей, що охоплюють як класичні методи текстової класифікації, так і сучасні трансформерні архітектури. Основна мета полягає у порівнянні їх продуктивності на однакових корпусах даних (LIAR та GossipCop) та у визначенні того, наскільки тонке налаштування трансформерних моделей може покращити результати класифікації у порівнянні з традиційними методами. Для цього розглянуто чотири моделі: логістичну регресію та Random Forest як базові підходи, а також DistilBERT і BERT-base як представників передових трансформерних систем.

Базові моделі працюють із TF-IDF векторизацією, яка перетворює текстові дані на числові ознаки, відображаючи важливість термів у документі та всьому корпусі. У випадку логістичної регресії використовується TF-IDF з n-грамами від

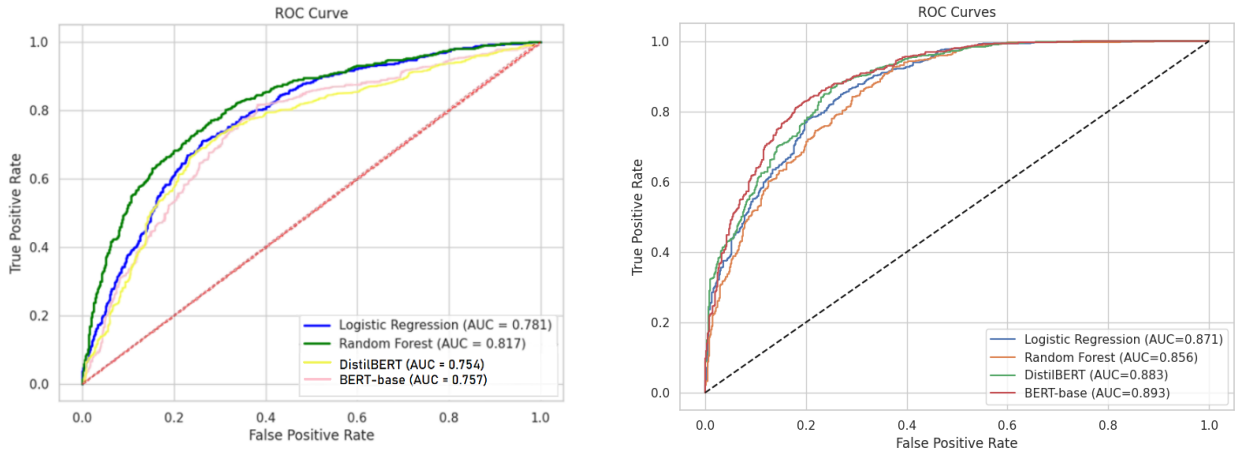
1 до 3, що дає змогу враховувати як окремі слова, так і короткі фрази. Застосовується L2-регуляризація зі стандартним параметром C = 1.0, а максимальна кількість ознак обмежена 12 тисячами. Для оптимізації використовується solver liblinear, який забезпечує стабільне навчання на малих та середніх корпусах.

Ще однією базовою моделлю є ансамбль дерев рішень (Random Forest). Ця модель поєднує велику кількість дерев, що дає змогу оцінювати важливість ознак та зменшувати ризик перенавчання окремого дерева. Для експериментів Random Forest навчається на тих же TF-IDF ознаках, а головні параметри, які контролюються для відтворюваності результатів, це кількість дерев (n_estimators) та максимальна глибина (max_depth) з початковими значеннями 200 та 20 відповідно. Процес навчання включає багаторазове створення дерев із випадковим підставлянням підмножини даних та ознак, після чого результати об’єднуються для остаточного прогнозу.

До трансформерних моделей включені DistilBERT та BERT-base. DistilBERT – це легка та швидка версія BERT, яка зберігає більшість точності при зменшенні розміру моделі та швидкості навчання. BERT-base використовується для порівняння впливу розміру трансформера на продуктивність класифікації. Процес навчання трансформерів включає попереднє токенізування текстів, для обох моделей встановлюється однакова базова конфігурація: learning rate 2e-5, batch size 16, десять епох навчання та фіксований random_state. Максимальна довжина токенізованої послідовності залежить від корпусу: для LIAR використовується значення 128, тоді як для довших текстів GossipCop – 512. Використання переднавчених моделей суттєво прискорює процес навчання та сприяє стабільному зближенню навіть на невеликих наборах даних. Усі параметри моделей та результати навчання зберігаються у конфігураційних файлах (model_config.json), що забезпечує відтворюваність експериментів.

Результати роботи моделей оцінювались за ROC-кривими на двох корпусах даних (Рисунок 4.3). Для корпусу LIAR найвищі значення AUC показала модель Random Forest (0.817), що підкреслює її перевагу над іншими підходами

на цьому наборі даних. Логістична регресія досягла AUC = 0.781, тоді як трансформери продемонстрували дещо нижчу продуктивність: DistilBERT – 0.754 та BERT-base – 0.757. Це пояснюється тим, що трансформери без спеціальної архітектури або додаткових шарів менш точно обробляють табличні чи метадані. Для корпусу GossipCop усі моделі продемонстрували вищу продуктивність, причому BERT-base лідирує з AUC 0.893, слідом йдуть DistilBERT 0.883, логістична регресія 0.871 та Random Forest 0.856. В цілому, ROC-криві демонструють, що сучасні трансформери краще розділяють класи на складних та об'ємних корпусах із довшими текстами, таких як GossipCop, тоді як прості моделі, особливо Random Forest, здатні досягати високої продуктивності на коротких наборах даних із багатими метаданими, таких як LIAR.



а) б)

Рисунок 4.3 – Порівняння ROC-кривих на різних корпусах: а) LIAR та б) FakeNewsNet (GossipCop)

Окрім побудови ROC-кривих на базових параметрах, проведено додаткове дослідження, щоб оцінити вплив зміни гіперпараметра random_state на точність класифікації моделей. У таблиці 4.1 наведено результати цього аналізу для чотирьох моделей на корпусі LIAR: Logistic Regression, Random Forest, DistilBERT та BERT-base.

Для кожної моделі наведено точність класифікації при трьох різних значеннях random_state (42, 8 та 123). Logistic Regression демонструє стабільні

результати в межах 0.718–0.726, що свідчить про невелику залежність від початкової ініціалізації. Random Forest показує трохи ширший діапазон точності (0.734–0.75), проте загальна продуктивність залишається високою, що підтверджує стійкість ансамблевого методу до змін випадкового насіння. DistilBERT виявляє більшу чутливість до random_state, з коливаннями точності від 0.659 до 0.71, що вказує на помітний вплив початкової ініціалізації на результати навчання трансформера. Модель BERT-base також демонструє варіації точності в межах 0.672–0.711, хоча загальні показники залишаються на конкурентному рівні.

Таблиця 4.1 – Матриця тестів для моделей зі зміною параметра «random_state» на наборі даних LIAR

Модель	**Значення параметра**	**Точність**
Logistic Regression	42/8/123	0.718/0.723/0.726
Random Forest	42/8/123	0.746/0.734/0.75
DistilBERT	42/8/123	0.701/0.659/0.71
BERT-base	42/8/123	0.711/0.672/0.675

Для оцінювання впливу гіперпараметрів на навчання трансформерних моделей проведено серію експериментів на корпусі даних LIAR. Для кожної моделі використовувався фіксований параметр random_state, обраний як оптимальний для початкових умов, що забезпечує відтворюваність результатів. Найкращі результати точності з параметрами до них виділено жирним стилем тексту.

Таблиця 4.2 демонструє результати тестування моделі DistilBERT при різних комбінаціях кількості епох (Epochs), розміру батчу (Batch) та швидкості навчання (learning rate). Найвищу точність (0.745) модель досягає при 10 епохах, батчі 32 та lr = 1e-3. Менші значення епох або батчу, а також низькі швидкості навчання призводять до зниження точності (найнижча – 0.7 при 5 епохах, батчі 32 та lr = 2e-5). Такі результати підкреслюють критичну роль підбору оптимальних

параметрів для досягнення максимальної продуктивності трансформерних моделей.

Таблиця 4.2 – Матриця тестів для DistilBERT зі зміною параметрів на наборі даних LIAR

Epochs	**Batch**	**Learning Rate**	**Точність**
10	16	2e-5	0.71
10	**32**	**1e-3**	**0.745**
5	16	1e-3	0.738
5	32	2e-5	0.7

Таблиця 4.3 відображає аналогічні результати для моделі BERT-base. Тут максимальна точність (0.749) досягається при двох комбінаціях параметрів: 10 епох, батч 32, lr = 1e-3 та 5 епох, батч 16, lr = 1e-3. Найнижча точність (0.608) спостерігається при 5 епохах, батч 32 та lr = 2e-5, що свідчить про високу чутливість BERT до параметрів навчання при обмеженій кількості епох та низькому learning rate. Отже, як оптимальна комбінація для BERT-base на корпусі LIAR обрана третя комбінація, оскільки вона забезпечує високу точність при зменшених обчислювальних витратах.

Таблиця 4.3 – Матриця тестів для BERT зі зміною параметрів на наборі даних LIAR

Epochs	**Batch**	**Learning Rate**	**Точність**
10	16	2e-5	0.711
10	32	1e-3	0.749
5	**16**	**1e-3**	**0.749**
5	32	2e-5	0.608

Також проведено дослідження впливу зміни гіперпараметрів на точність класифікації для базових моделей. Таблиці 4.4 та 4.5 демонструють результати

параметричних експериментів для алгоритмів Random Forest та Logistic Regression, виконаних на корпусі LIAR. Для обох моделей використовувався фіксований параметр random_state = 123, що гарантує відтворюваність отриманих результатів та коректність порівняння різних конфігурацій.

Таблиця 4.4 – Матриця тестів для Random Forest зі зміною параметрів на наборі даних LIAR

n_estimators	**max_depth**	**Точність**
200	20	0.75
200	**10**	**0.754**
200	30	0.75
500	20	0.753

У таблиці 4.4 наведено точність Random Forest за різних комбінацій параметрів n_estimators та max_depth. Отримані значення свідчать, що модель демонструє стабільну продуктивність у діапазоні 0.75–0.754, а зміна глибини дерев або збільшення кількості дерев до 500 не забезпечує суттєвого покращення якості. Найвищий результат (0.754) досягнуто для конфігурації n_estimators = 200 та max_depth = 10. Подальше збільшення складності ансамблю не призводить до помітного приросту точності, що може свідчити про досягнення межі підвищення точності для наявного простору ознак (ембедінгів).

Таблиця 4.5 подає результати експериментів із логістичною регресією за різних конфігурацій параметрів max_iter та solver. Усі протестовані варіанти демонструють близьку точність (0.724–0.731), причому найвищий результат отримано для комбінації max_iter = 5000 та solver = lbfgs. Це свідчить про те, що для логістичної регресії важливо забезпечити достатню кількість ітерацій оптимізації, а також правильно підібрати метод розв’язання, однак загальний вплив цих параметрів на кінцеву точність залишається помірним.

Таблиця 4.5 – Матриця тестів для Logistic Regression зі зміною параметрів на наборі даних LIAR

max_iter	**solver**	**Точність**
5000	liblinear	0.726
1000	liblinear	0.724
1000	lbfgs	0.73
5000	**lbfgs**	**0.731**

Загалом обидві моделі демонструють відносно стійку поведінку в межах протестованих параметрів, що свідчить про їхню стабільність на корпусі LIAR. Random Forest показує вищу точність, тоді як логістична регресія залишається простішим та більш інтерпретованим базовим алгоритмом.

Отже, аналіз показників точності та ROC-кривих демонструє, що точність моделей значно залежить від типу корпусу та наявності додаткових ознак. На короткому та насиченому метаданими наборі LIAR найкращі результати показує Random Forest, тоді як трансформери без спеціальної архітектури працюють трохи гірше. Для довгих текстових корпусів, таких як GossipCop, сучасні трансформери (BERT-base, DistilBERT) забезпечують кращу здатність відокремлювати класи, перевершуючи прості моделі. Коригування параметрів може вплинути на показники приблизно на 1–3%, а використання альтернативних наборів, наприклад CONSTRAINT-2021 (EN) [52], потенційно дозволяє отримати ще вищі результати.

4.3 Комплексний аналіз метрик якості та оцінка класифікаційних помилок

Для оцінювання результативності моделей, залучених у процесі виявлення фейкових новин, використовується комплекс узгоджених метрик, які дають змогу всебічно проаналізувати поведінку алгоритмів на різних типах представлення тексту – від класичних TF-IDF ознак до сучасних трансформерних ембедінгів.

Основними показниками виступають Accuracy, Precision, Recall, F1-score, а також ROC-AUC та матриці помилок, що дають змогу оцінити структуру класифікаційних рішень і виявити проблеми в моделі.

Усі чотири моделі – логістична регресія, випадковий ліс, DistilBERT і BERT-base – оцінюються за однаково визначеним набором метрик та на тих самих тестових вибірках, що забезпечує коректність порівняння. Базові моделі працюють на векторизованих TF-IDF ознаках, а трансформери безпосередньо опрацьовують текстові послідовності після токенізації, що дає змогу порівняти класичні та сучасні підходи до обробки природної мови.

Для оцінювання результативності моделей у процесі класифікації новин усі результати точності та якості роботи представлені у вигляді підсумкових таблиць. Кожна модель була протестована в конфігураціях з оптимальними гіперпараметрами, визначеними в підрозділі 4.2, що забезпечує коректність порівняння та надійність отриманих показників.

У таблиці 4.6 наведено підсумкові метрики для корпусу LIAR, який характеризується великою кількістю класів та невеликою довжиною текстових повідомлень. Модель з найкращими результатами виділено жирним стилем тексту.

Таблиця 4.6 – Підсумок обчислених метрик на наборі даних LIAR

Модель	**Accuracy**	**Precision**	**Recall**	**F1-score**
Логістична регресія	0.731	0.73	0.725	0.761
Випадковий ліс	0.754	0.76	0.75	0.777
DistilBERT	0.745	0.779	0.775	0.777
BERT-base	**0.749**	**0.767**	**0.798**	**0.782**

З результатів в таблиці видно, що логістична регресія показує найнижчі значення точності та повноти, що свідчить про обмеженість її здатності враховувати складні мовні патерни, характерні для завдання визначення фейкових висловлювань. Випадковий ліс забезпечує помітно кращу збалансованість між

точністю та повнотою, демонструючи більш стійку роботу на різноманітних прикладах.

Архітектури DistilBERT і BERT-base, натреновані на мовних представленнях трансформерного типу, суттєво підвищують якість класифікації. DistilBERT досягає найвищої точності та майже максимальної збалансованості між precision і recall, тоді як BERT-base забезпечує найвищий F1-score та найкращий показник recall, тобто здатен виявляти найбільшу кількість фейкових тверджень, зберігаючи Водночас стабільну точність.

Також проведено аналіз результативності моделей на складнішому корпусі даних FakeNewsNet (GossipCop). У таблиці 4.7 наведено метрики для цього корпусу, який відзначається значно більшою середньою кількістю слів у кожному тексті та менш збалансованим розподілом класів. Модель з найкращими результатами виділено жирним стилем тексту.

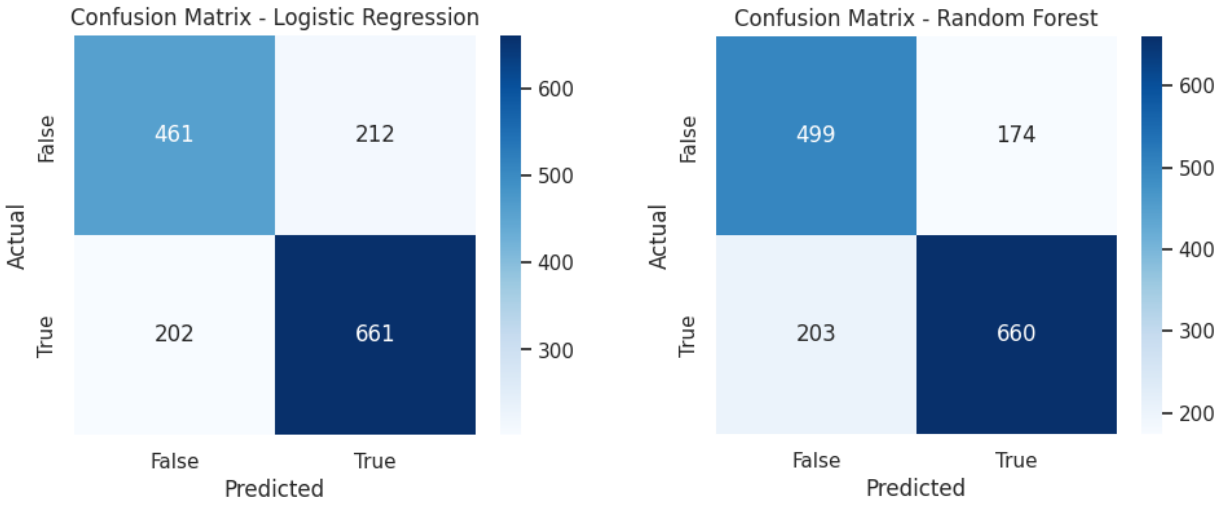
Таблиця 4.7 – Підсумок обчислених метрик на наборі даних GossipCop

Модель	**Accuracy**	**Precision**	**Recall**	**F1-score**
Логістична регресія	0.788	0.751	0.863	0.8
Випадковий ліс	0.765	0.716	0.877	0.788
DistilBERT	**0.805**	**0.762**	**0.885**	**0.819**
BERT-base	0.789	0.729	0.92	0.813

Результати показують, що на більших корпусах трансформерні моделі демонструють суттєву перевагу. DistilBERT досягає найвищого F1-score (0.819), а також кращих значень Recall, що свідчить про здатність архітектури виявляти більшість фейкових новин. BERT-base показує подібну точність, але ще вищий Recall (0.920), що є очікуваним для моделі з більшою кількістю параметрів.

Базові методи залишаються точними (особливо логістична регресія з F1 = 0.8), однак поступаються трансформерам на складніших або багатовимірних текстових просторах.

Окрім узагальнених метрик точності, для детальнішого аналізу поведінки кожної моделі побудовано матриці помилок для відповідних корпусів даних. Матриця помилок є важливим інструментом оцінювання класифікаційних моделей, оскільки дає змогу не лише визначити загальну точність, а й побачити структуру допущених помилок: які саме класи частіше плутаються та у яких пропорціях виникають хибнопозитивні та хибнонегативні рішення. Нижче, на рисунку 4.4 наведено матриці помилок для логістичної регресії та випадкового лісу, отримані під час оцінювання на корпусі LIAR.

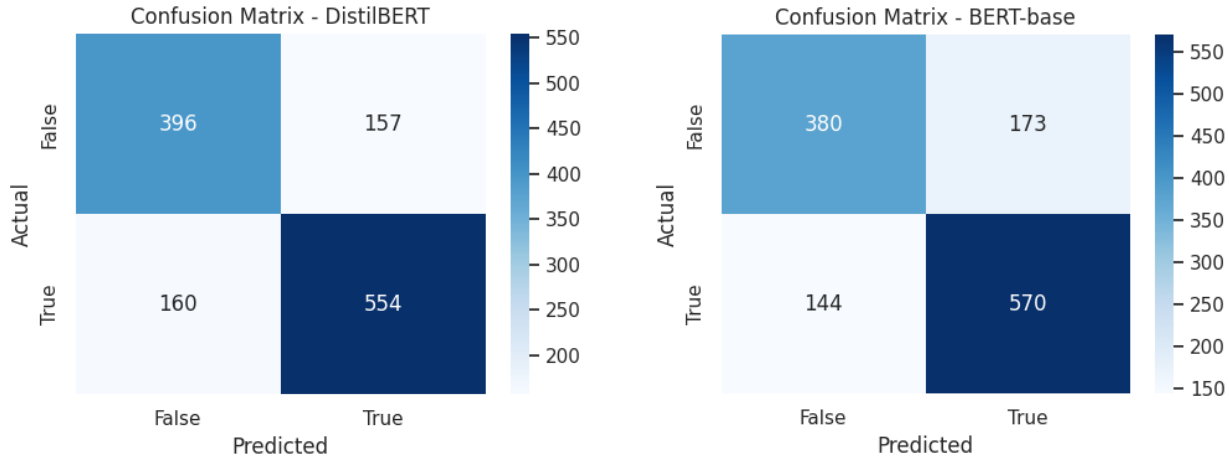


а) б)

Рисунок 4.4 – Порівняння матриць помилок для корпусу LIAR на різних моделях: а) Логістична регресія та б) Випадковий ліс

Основні спостереження свідчать про те, що логістична регресія дещо краще розпізнає правдиві новини, ніж фейкові. Зокрема, кількість хибнопозитивних класифікацій (212) перевищує кількість хибнонегативних (202), що вказує на тенденцію моделі частіше «довіряти» сумнівним новинам та класифікувати їх як правдиві. Випадковий ліс натомість, демонструє краще виявлення класу «False» та загалом забезпечує більш збалансований розподіл між вірно та помилково класифікованими прикладами обох класів. Водночас показники для класу «True» залишаються майже ідентичними до результатів логістичної регресії.

На іншому Рисунку 4.5, подано порівняння матриць помилок для моделей DistilBERT і BERT-base на корпусі LIAR, що дає змогу оцінити характер їхніх помилок. Обидві моделі демонструють схожу структуру класифікації, проте BERT-base загалом точніше визначає позитивний клас: у неї менше хибних від'ємних спрацювань (144 проти 160 у DistilBERT) та більше правильно класифікованих позитивних прикладів (570 проти 554). DistilBERT, натомість, трохи краще розпізнає негативний клас, маючи більше правильних негативних передбачень (396 проти 380). У цілому обидві моделі показують збалансовану роботу, але BERT-base демонструє кращу здатність виявляти фейкові твердження.



а) б)

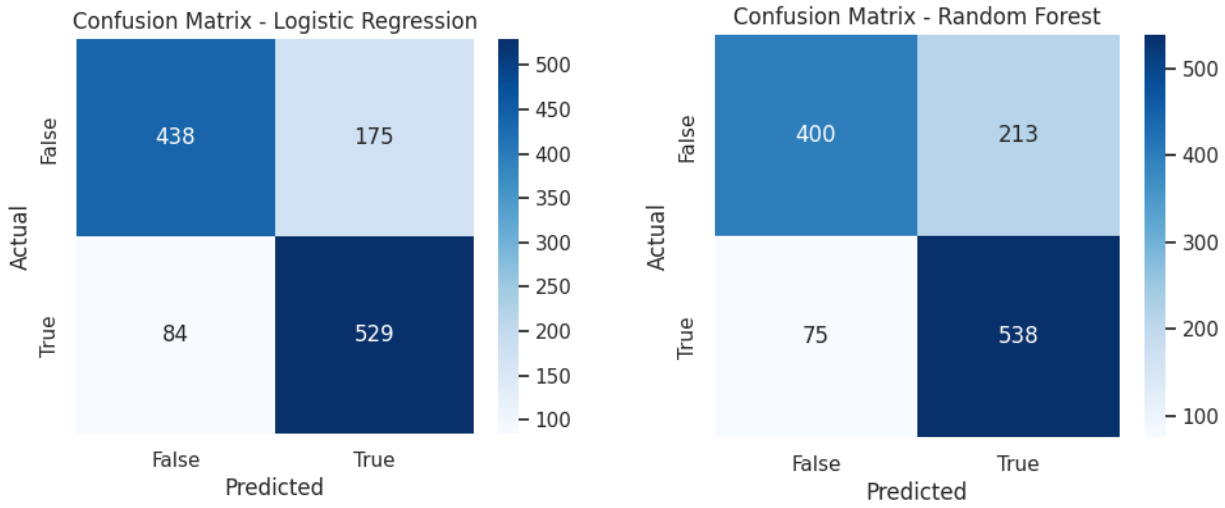
Рисунок 4.5 – Порівняння матриць помилок для корпусу LIAR на різних моделях: а) DistilBERT та б) BERT

Узагальнюючи результати, логістична регресія демонструє помірну схильність частіше класифікувати сумнівні новини як правдиві, тоді як випадковий ліс забезпечує більш збалансоване розпізнавання обох класів. Моделі DistilBERT і BERT-base показують подібну структуру помилок, проте BERT-base точніше виявляє фейкові твердження завдяки меншій кількості пропущених випадків класу «False». DistilBERT трохи краще виокремлює правдиві повідомлення, але загалом поступається BERT-base у здатності чітко

розмежовувати класи. У підсумку найвищу якість класифікації на корпусі LIAR демонструє BERT-base.

Окрім корпусу LIAR, сформовано та проаналізовано матриці помилок для корпусу GossipCop. Зазначений набір даних вирізняється значно більшими текстовими обсягами та певною нерівномірністю розподілу класів, що дає змогу оцінити стійкість моделей у складніших умовах. Матриці, наведені на рисунку 4.6, відображають продуктивність класичних алгоритмів.

Для логістичної регресії спостерігається достатньо стабільна робота в обох класах, однак модель краще ідентифікує правдиві новини. Коректно класифіковано 529 прикладів класу «True» та 438 прикладів класу «False». Водночас кількість хибнопозитивних рішень перевищує кількість хибнонегативних, що свідчить про схильність моделі частіше помилково позначати неправдиві новини як правдиві.



а) б)

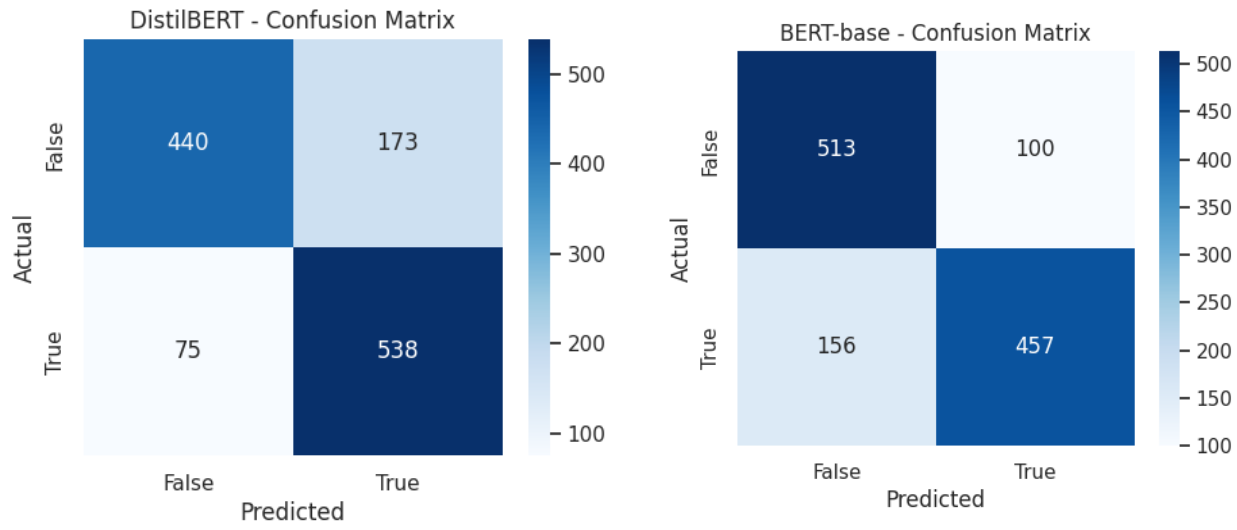
Рисунок 4.6 – Порівняння матриць помилок для корпусу GossipCop на різних моделях: а) Логістична регресія та б) Випадковий ліс

Випадковий ліс демонструє більш збалансовану поведінку. Хоча кількість правильних класифікацій класу «False» дещо нижча порівняно з логістичною регресією, модель забезпечує вищу точність для класу «True» та зменшує кількість хибнонегативних випадків. Разом із тим вона допускає більше

хибнопозитивних помилок (213), що вказує на підвищену чутливість до класу «True» і певну схильність до «довірливих» рішень.

Окремо були створені матриці помилок для моделей-трансформерів. Як показано на рисунку 4.7, трансформерні підходи демонструють відчутну перевагу над класичними моделями. DistilBERT забезпечує покращені результати для обох класів: 440 правильних класифікацій для «False» та 538 для «True». Модель демонструє зменшену кількість хибнонегативних рішень (75), що свідчить про кращу здатність правильно виявляти правдиві новини. Кількість хибнопозитивних випадків (173) також нижча, ніж у класичних алгоритмів, що підкреслює загальну збалансованість підходу.

BERT-base демонструє найвищу точність поміж усіх розглянутих моделей за певними показниками. Він досягає найбільшої кількості правильних класифікацій класу «False» (513) та значно зменшує хибнопозитивні помилки (100), що свідчить про точну ідентифікацію неправдивих новин.



а) б)

Рисунок 4.7 – Порівняння матриць помилок для корпусу GossipCop на різних моделях: а) DistilBERT та б) BERT

Водночас для класу «True» BERT-base показує 457 вірних класифікацій, хоча кількість хибнонегативних випадків (156) дещо більша порівняно з DistilBERT. Ймовірною причиною цього є підвищена чутливість BERT до

семантичних нюансів тексту, що іноді призводить до надмірної впевненості у прогнозах на користь класу «True».

У підсумку, порівняння матриць помилок класичних алгоритмів (логістична регресія, випадковий ліс) і трансформерних моделей (DistilBERT, BERT-base) свідчить про суттєву перевагу останніх. Трансформери демонструють нижчі рівні як хибнопозитивних, так і хибнонегативних класифікацій, забезпечуючи більш точне та збалансоване розмежування класів. Класичні підходи, хоча й залишаються працездатними, поступаються трансформерам у здатності адекватно обробляти складніші й довші текстові послідовності, характерні для корпусу GossipCop.

4.4 Аналіз візуалізацій простору ознак та інтерпретацій рішень моделей методами пояснюваного штучного інтелекту

Для глибшого аналізу роботи класифікаційної моделі, окрім оцінки через метрики, матриці помилок та ROC-криві, користувачу надаються візуалізації простору ознак та пояснення методами XAI як для окремих новин, так і для всього корпусу даних. Нижче подано результати такого аналізу.

На рисунку 4.8 показано проєкцію багатовимірного вектора ознак у двовимірний простір для набору даних LIAR. Вектор ознак кожного зразка формувався шляхом конкатенації текстових ембедінгів, отриманих із моделі SentenceBERT, та закодованих метаданих (інформація про спікера, партію, контекст тощо). Класифікація виконувалася моделлю логістичної регресії.

На рисунку а, на відміну від суто текстових наборів даних, де точки зазвичай формують одну щільну хмару, спостерігається чітке формування окремих кластерів. Така структура зумовлена впливом метаданих, причому окремі групи точок можуть відповідати конкретним спікерам або тематичним підмножинам, що мають схожі характеристики в просторі ознак. В межах цих кластерів помітне змішування червоних та зелених точок, що вказує на наявність як правдивих, так і неправдивих заяв від одного й того самого спікера або у межах

однієї теми. На рисунку б подано розподіл прогнозованих класів, який візуально дуже схожий на істинний (рисунок а), що свідчить про здатність лінійної моделі коректно відображати структуру простору, утвореного комбінацією SentenceBERT та метаданих.

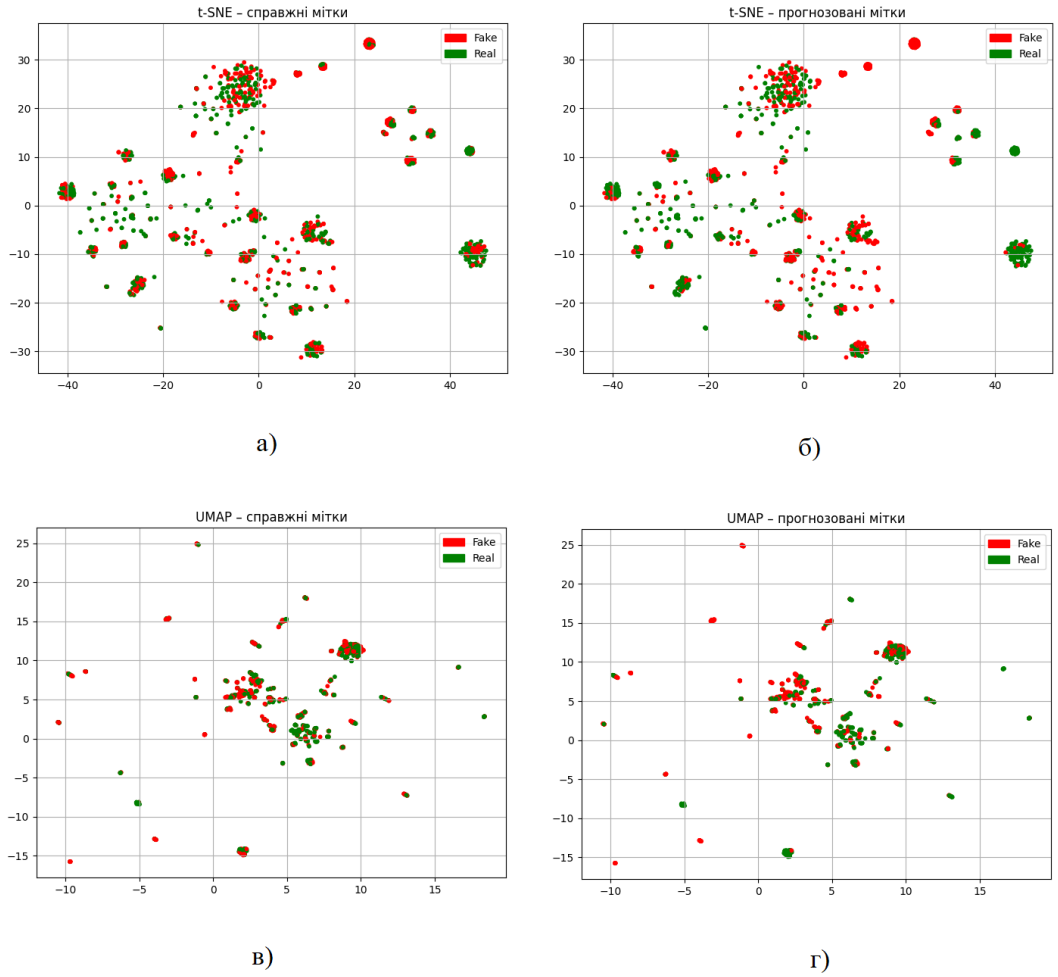


Рисунок 4.8 – Візуалізації для корпусу LIAR методами зниження розмірності: а), б) t-SNE та в), г) UMAP

Проєкція UMAP (рисунок в) демонструє локальні групування даних, де окремі кластери мають домінування одного кольору, що спрощує класифікацію цих підмножин. На рисунку г відображено прогнозовані мітки у просторі UMAP. Як і у випадку з t-SNE, модель успішно відтворює більшість локальних закономірностей, хоча у зонах сильного змішування класів можливі похибки.

Окрім візуалізації простору ознак для корпусу LIAR, виконано проєкцію багатовимірного простору ознак тестового набору GossipCop, отриманих за

допомогою векторизації моделі-трансформера DistilBERT, у двовимірний простір. Результати цих проєкцій, представлені на рисунку 4.9, дають змогу оцінити розподіл зразків новин за класами та виявити потенційні кластери, що відображають схожість між текстами. Використання двох алгоритмів зниження розмірності – t-SNE та UMAP – надає можливість порівняти локальну та глобальну структуру даних у візуалізації. Кожна точка на графіку відповідає окремому зразку новин.

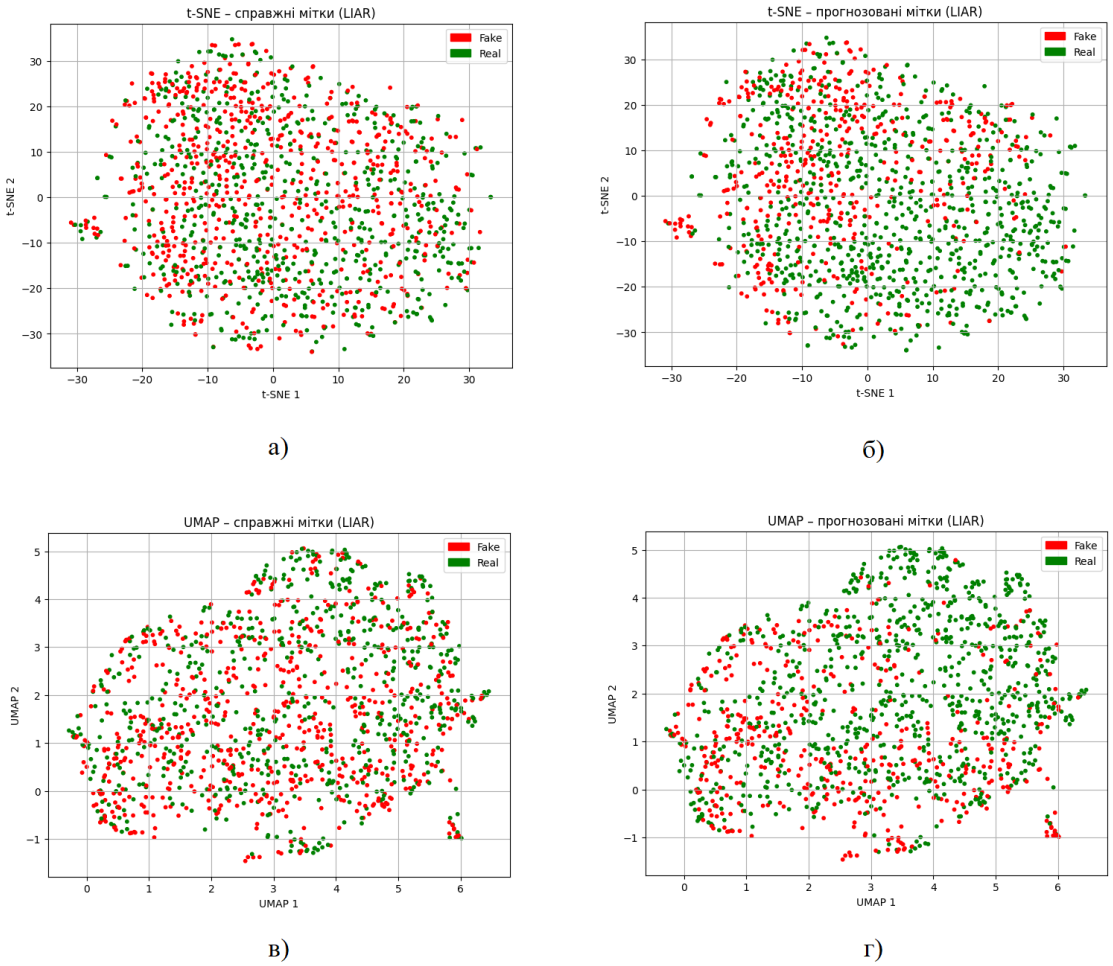


Рисунок 4.9 – Візуалізації для корпусу GossipCop методами зниження розмірності: а), б) t-SNE та в), г) UMAP

На графіку t-SNE (рисунок а) спостерігається значне перекриття між кластерами справжніх та фейкових новин. Чіткої границі поділу немає, що підкреслює складність класифікації даних лише на основі текстових ознак. На рисунку б відображено розподіл прогнозованих класів, який візуально повторює

структуру істинних класів (рисунок а), що свідчить про здатність моделі відтворювати загальні закономірності даних, проте не ідеально розділяти змішані області.

Алгоритм UMAP (рисунок в), який краще зберігає глобальну структуру даних, також показує сильне перемішування точок класів «Real» та «Fake». Видно кілька віддалених невеликих кластерів (справа вгорі), проте основна маса даних концентрується в щільному скупченні. Результати прогнозу моделі у просторі UMAP (рисунок г), аналогічно до t-SNE, відображають загальну структуру даних і корелюють із розподілом істинних класів, підтверджуючи адекватність моделі у відтворенні глобальних закономірностей.

Для інтерпретації рішень моделей створено низку графіків пояснень на прикладі набору даних LIAR. Один із графіків демонструє глобальні пояснення за допомогою методу SHAP, а два інших – локальні пояснення для окремої новини, побудовані з використанням SHAP та Integrated Gradients.

На рисунку 4.10 подано глобальне пояснення для базової моделі з логістичною регресією. Графік відображає внесок різних ознак (features) у прогноз моделі, тобто ймовірність того, що новина є правдивою. Ознаки відсортовані за спаданням їхньої глобальної важливості. Найважливішою ознакою виявилось закодоване ім’я спікера, що видно з великого розкиду точок по горизонталі. Це свідчить про те, що модель переважно покладається на історичну правдивість конкретного політика чи організації, а не лише на зміст тексту. Наступними за важливістю є метадані про попередні заяви спікера, що підкреслює роль його «послужного списку» у визначенні правдивості поточної заяви. Всі інші слова, представлені через TF-IDF, мають менший, але все ж помітний вплив. Отже, модель засвоїла, що контекст – хто робить заяву та яка історія його попередніх висловлювань – є більш надійним предиктором правдивості, ніж окремі слова в тексті. Це типова поведінка для корпусу LIAR, де стиль мовлення може не сильно відрізнятися між правдою та брехнею, але певні політики брешуть систематично частіше за інших.

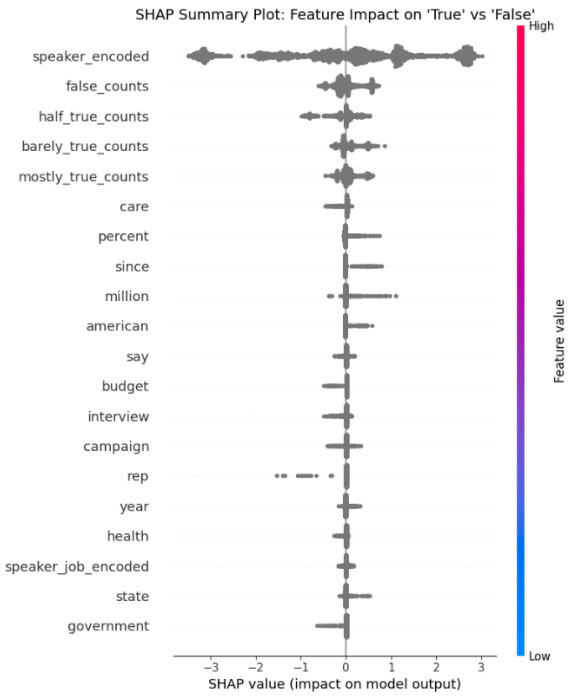
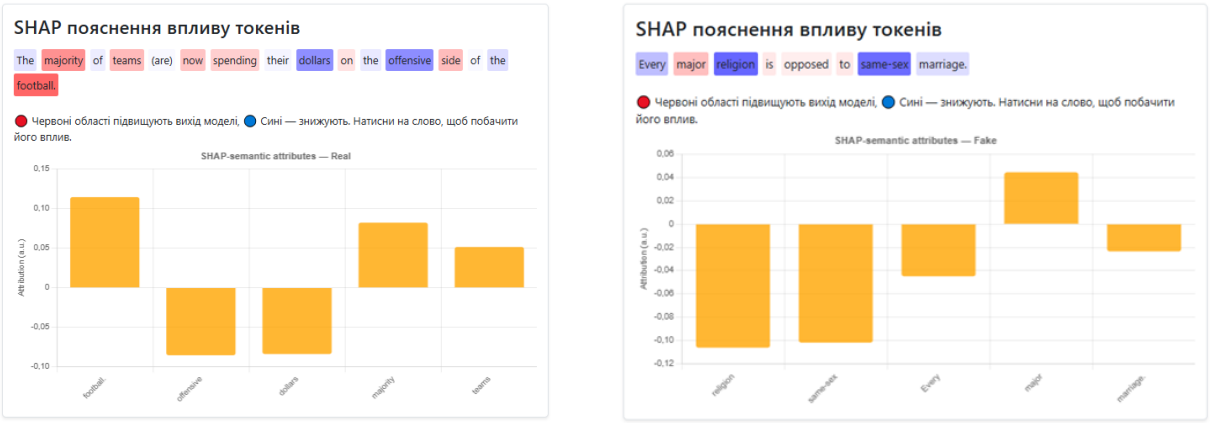


Рисунок 4.10 – Важливість ознак за SHAP методом для базової моделі класифікації правдивості новин за корпусом даних LIAR

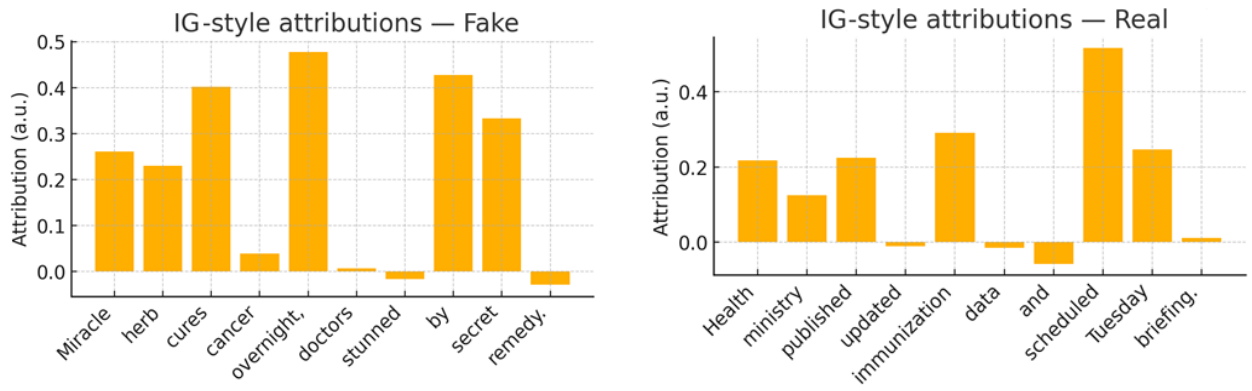
На наступних рисунках 4.11–4.12 наведено локальні атрибуції ознак для конкретних прикладів класів fake та real, побудовані за допомогою SHAP та Integrated Gradients. Для фейкових текстів спостерігається концентрація високих вагових впливів у словах із сенсаційним або емоційним забарвленням, що відповідає типовим маркерам фейкових повідомлень.



а) б)

Рисунок 4.11 – Порівняння SHAP-поясненя для окремого тексту з набору даних LIAR: а) класу «Real» та б) класу «Fake»

Для правдивих текстів домінують нейтральні або фактологічні лексеми. Такі результати демонструють, що система не лише здійснює класифікацію текстів, але й надає інформативні пояснення щодо факторів, що впливають на прийняття рішення, забезпечуючи прозорість роботи алгоритму.



а) б)

Рисунок 4.12 – Порівняння IG-пояснення для окремого тексту з набору даних LIAR: а) класу «Fake» та б) класу «Real»

Отже, реалізований підхід до інтерпретування результатів виявлення фейкових новин забезпечує користувачу доступ до різних форм пояснення роботи моделі: від класичних метрик класифікації, матриць помилок та ROC-кривих – до візуалізацій простору ознак і пояснень, згенерованих XAI-методами. Такий комплексний підхід дає змогу глибше зрозуміти, на основі яких ознак модель ухвалює свої рішення та які компоненти тексту або метаданих найбільше впливають на класифікацію.

Отримані результати демонструють, що у випадках, коли набір даних містить додаткові інформативні ознаки, окрім тексту (наприклад, метадані про спікера чи контекст), модель формує більш структурований простір ознак, що полегшує як побудову якісних візуалізацій, так і створення змістовних пояснень.

Висновки до розділу 4

Таким чином, для перевірки роботи шостого кроку запропонованого методу проведено комплексне дослідження на двох корпусах – LIAR та FakeNewsNet (підвибірка GossipCop). Аналіз охоплював чотири моделі: дві прості (логістична регресія та випадковий ліс) і дві трансформерні (DistilBERT та BERT-base). Отримані результати засвідчили, що на великих та змістовних текстах моделі-трансформери демонструють вищу загальну точність завдяки здатності глибше опрацьовувати контекст і семантичні залежності. Водночас прості моделі показали непогані результати, забезпечуючи стабільну роботу на обох корпусах.

На корпусі LIAR, який є складним через короткі та лаконічні тексти, прості моделі продемонстрували вищу ефективність: вони краще працюють із табличними ознаками та малими обсягами даних. Натомість трансформери без спеціальної обробки метаданих дають нижчу точність. Для корпусу GossipCop ситуація протилежна – трансформерні моделі забезпечують глибший аналіз довших і складніших новин, хоча потребують більше ресурсів.

Окрім метрик класифікації, дослідження продемонструвало можливості моделей формувати інформативні візуалізації простору ознак та пояснення XAI-методами. Для LIAR спостерігається виражена кластеризація завдяки додатковим метаданим, тоді як у GossipCop точки формують менш структуровану «хмару», що пояснюється переважно текстовою природою ознак. Аналіз пояснень XAI показав, що фейкові новини характеризуються наявністю емоційно забарвлених або маніпулятивних слів, тоді як правдиві – нейтральною та фактологічною лексикою.

Дослідження впливу зміни параметрів моделей засвідчило, що оптимізація гіперпараметрів може підвищити точність у середньому на 2–4%. Використання інших корпусів, зокрема CONSTRAINT-2021 (EN), у поєднанні з розробленим методом потенційно може забезпечити ще вищі показники якості класифікації.

Висновки

Кваліфікаційна робота магістра спрямована на підвищення рівня інтерпретованості та точності систем виявлення фейкових новин через створення методу інтерпретування результатів виявлення фейкових новин із використанням великих мовних моделей та технологій пояснюваного штучного інтелекту.

Результатом роботи є створений метод інтерпретування результатів виявлення фейкових новин на базі великої мовної моделі, який забезпечує обробку та класифікацію новинних текстів, а також отримання інтерпретацій «хід думки моделі» через аналіз метрик, простору текстових ембедінгів та пояснення, сформовані за допомогою XAI-методів. Для дослідження результативності запропонованого підходу було створено інформаційну систему у вигляді вебзастосунку, що дає можливість класифікувати новини, інтерпретувати роботу моделей та змінювати параметри й використовувані датасети. Проведені дослідження показали, що застосування розробленого методу дає змогу підвищити точність класифікації на 2–4%, що підтверджує успішне досягнення мети кваліфікаційної роботи магістра.

Проведені експерименти на двох різних корпусах показали, що короткі тексти без додаткових метаданих істотно знижують точність класифікації, як у простих моделей на кшталт логістичної регресії, так і у трансформерних архітектур. За таких умов показники точності зазвичай не перевищують 59–64%. Аналіз набору LIAR продемонстрував, що для класифікації коротких та малозмістовних висловлювань моделі потребують значно більшого обсягу інформації, ніж сам лише текст. Водночас набір GossipCop, який хоч і не містить додаткових ознак окрім тексту та мітки класу, забезпечує вищу якість класифікації завдяки наявності довших і більш насичених змістом новинних статей.

Застосування методів інтерпретації рішень моделі у запропонованому методі дало можливість глибше зрозуміти як структуру даних, що подаються на вхід, так і принципи формування прогнозів. Візуалізації простору ознак, зокрема двовимірні проекції та графічні схеми важливості ознак, допомагають виявити

приховані закономірності: кластери, кореляції між групами текстів або дисбаланси між класами. Подібні структури особливо добре простежуються в корпусі LIAR, де наявність метаданих створює виразну кластеризацію. Методи XAI, такі як SHAP, Integrated Gradients та LIME, забезпечують детальні пояснення роботи моделей і особливо важливі для інтерпретації трансформерних архітектур. Завдяки їх використанню виявлено, що у корпусі LIAR фейкові висловлювання часто характеризуються підвищеною концентрацією емоційно забарвлених та сенсаційних лексем, тоді як правдиві висловлювання переважно містять нейтральні або стримані за тоном слова.

Для подальшого підвищення результативності роботи методу можна замінити використану модель-трансформер DistilBERT на більшу за розміром архітектуру з більшою кількістю параметрів або на модель, попередньо натреновану саме на новинних корпусах.

За темою кваліфікаційної роботи автором виконано дві наукові публікації. Основні наукові й практичні результати роботи доповідались у статті «Метод інтерпретування результатів виявлення фейкових новин за великою мовною моделлю» у журналі Вісник Хмельницького національного університету 2025 року та у доповіді «Метод інтерпретування результатів виявлення фейкових новин за великою мовною моделлю» на XVII Всеукраїнській науково-практичній конференції «Актуальні проблеми комп’ютерних наук АПКН-2025» (м. Хмельницький) 14–15 листопада 2025 року.

Перелік посилань

1. Вовк С. В., Радюк П. М., Скрипник Т. К. Метод інтерпретування результатів виявлення фейкових новин за великою мовною моделлю. *Актуальні проблеми комп'ютерних наук АПКН-2025* : матеріали XVII Всеукр. науково-практ. конф., м. Хмельницький, 14–15 листоп. 2025 р. Хмельницький, 2025. С. 68–71. URL: https://elar.khmnu.edu.ua/handle/123456789/19862 (дата звернення: 03.12.2025)

2. Вовк С. В., Радюк П. М., Скрипник Т. К. Метод інтерпретування результатів виявлення фейкових новин за великою мовною моделлю. *Вісник Хмельницького національного університету. Технічні науки*. 2025. Т. 359, № 6(2). (Довідка з редакції).

3. Interpreting Black-Box Models: A Review on Explainable Artificial Intelligence / V. Hassija et al. Cognitive Computation. 2023. URL: https://doi.org/10.1007/s12559-023-10179-8 (date of access: 30.11.2025).

4. Human-in-the-loop approach based on MRI and ECG for healthcare diagnosis / P. Radiuk et al. *Proceedings of the 5th international conference on informatics & data-driven medicine* : CEUR-Workshop Proceedings, Lyon, 18–20 November 2022. Aachen, 2022. P. 9–20. URL: https://ceur-ws.org/Vol-3302/paper1.pdf (date of access: 28.11.2025).

5. A survey on hallucination in large language models: principles, taxonomy, challenges, and open questions / L. Huang et al. *ACM transactions on information systems*. 2024. Vol. 43, no. 2. P. 1–55. URL: https://doi.org/10.1145/3703155 (date of access: 29.11.2025).

6. Bjerring J. C., Mainz J., Munch L. Deep learning models and the limits of explainable artificial intelligence. Asian Journal of Philosophy. 2025. Vol. 4, no. 1. URL: https://doi.org/10.1007/s44204-024-00238-8 (date of access: 30.11.2025).

7. Ribeiro M. T., Singh S., Guestrin C. "Why Should I Trust You?". KDD '16: The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data

Mining, San Francisco California USA. New York, NY, USA, 2016. URL: https://doi.org/10.1145/2939672.2939778 (date of access: 30.11.2025).

8. Toward explainable deep learning in healthcare through transition matrix and user-friendly features / O. Barmak et al. *Frontiers in Artificial Intelligence*. 2024. Vol. 7. P. 1482141. URL: https://doi.org/10.3389/frai.2024.1482141 (date of access: 28.11.2025).

9. What Does BERT Look at? An Analysis of BERT's Attention / K. Clark et al. Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Florence, Italy. Stroudsburg, PA, USA, 2019. URL: https://doi.org/10.18653/v1/w19-4828 (date of access: 30.11.2025).

10. Attention-Based Bi-LSTM Model for Anomalous HTTP Traffic Detection / Y. Yu et al. 2018 15th International Conference on Service Systems and Service Management (ICSSSM), Hangzhou, China, 21–22 July 2018. 2018. URL: https://doi.org/10.1109/icsssm.2018.8465034 (date of access: 30.11.2025).

11. Information system for public places and institutions visualization with opportunities of inclusive access and optimal routing / O. Pavlova et al. *Computer systems and information technologies*. 2022. Vol. 1, no. 6. P. 62–68. URL: https://doi.org/10.31891/CSIT-2022-1-8 (date of access: 30.11.2025).

12. Schneider J. Explainable Generative AI (GenXAI): a survey, conceptualization, and research agenda. Artificial Intelligence Review. 2024. Vol. 57, no. 11. URL: https://doi.org/10.1007/s10462-024-10916-x (date of access: 30.11.2025).

13. Integrated gradients | TensorFlow Core. *TensorFlow*. URL: https://www.tensorflow.org/tutorials/interpretability/integrated_gradients (date of access: 30.11.2025).

14. A Survey of Methods for Explaining Black Box Models / R. Guidotti et al. ACM Computing Surveys. 2019. Vol. 51, no. 5. P. 1–42. URL: https://doi.org/10.1145/3236009 (date of access: 30.11.2025).

15. Barmak O., Radiuk P. Web-based information technology for classifying and interpreting early pneumonia based on fine-tuned convolutional neural network.

Computer systems and information technologies. 2021. Vol. 3, no. 1. P. 12–18. URL: https://doi.org/10.31891/CSIT-2021-3-2 (date of access: 28.11.2025).

16. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI / A. Barredo Arrieta et al. Information Fusion. 2020. Vol. 58. P. 82–115. URL: https://doi.org/10.1016/j.inffus.2019.12.012 (date of access: 30.11.2025).

17. Radiuk P.M. Application of a genetic algorithm to search for the optimal convolutional neural network architecture with weight distribution. *Herald of Khmelnytskyi National University. Technical sciences*. 2020. Vol. 281, no. 1. P. 7–11. URL: https://doi.org/10.31891/2307-5732-2020-281-1-7-11 (date of access: 30.11.2025).

18. Wang D., Hu H., Chen D. Transformer with sparse self-attention mechanism for image captioning. Electronics Letters. 2020. Vol. 56, no. 15. P. 764–766. URL: https://doi.org/10.1049/el.2020.0635 (date of access: 30.11.2025).

19. Singh G., Yow K.-C. These do not Look Like Those: An Interpretable Deep Learning Model for Image Recognition. IEEE Access. 2021. Vol. 9. P. 41482–41493. URL: https://doi.org/10.1109/access.2021.3064838 (date of access: 30.11.2025).

20. Huber L. Concept Learning: Making your network interpretable. Towards Data Science. URL: https://towardsdatascience.com/concept-learning-making-your-network-interpretable-735a0698fc03/ (date of access: 30.11.2025).

21. Shupta A., Radiuk P., Krak I. Feature computation procedure for fake news detection: An LLM-based extraction approach. *Proceedings of the 6th international workshop on intelligent information technologies & systems of information security (intelitsis 2025)*: CEUR-Workshop Proceedings, Khmelnytskyi, 4 April 2025. Aachen, 2025. P. 112–124. URL: https://ceur-ws.org/Vol-3963/paper10.pdf (date of access: 28.11.2025).

22. Rajesh M. Explainability for Text Data: 3D Visualization of Token Embeddings using PCA, t-SNE, and UMAP. Medium. URL: https://medium.com/@madhugraj/explainability-for-text-data-3d-visualization-of-

token-embeddings-using-pca-t-sne-and-umap-8da33602615b (date of access: 30.11.2025).

23. Guadagnolo L. Mastering Text Similarity: combining embedding techniques and distance metrics. Medium. URL: https://medium.com/eni-digitalks/mastering-text-similarity-combining-embedding-techniques-and-distance-metrics-98d3bb80b1b6 (date of access: 30.11.2025).

24. Belinkov Y. Probing Classifiers: Promises, Shortcomings, and Advances. Computational Linguistics. 2022. Vol. 48, no. 1. P. 207–219. URL: https://doi.org/10.1162/coli_a_00422 (date of access: 30.11.2025).

25. Maaten L. V. D., Hinton G. Visualizing Data using t-SNE. *Journal of machine learning research*. 2008. Vol. 9, no. 86. P. 2579–2605. URL: http://jmlr.org/papers/v9/vandermaaten08a.html (date of access: 29.11.2025).

26. UMAP: Uniform Manifold Approximation and Projection / L. McInnes et al. Journal of Open Source Software. 2018. Vol. 3, no. 29. P. 861. URL: https://doi.org/10.21105/joss.00861 (date of access: 30.11.2025).

27. Uniform Manifold Approximation and Projection (UMAP) / B. Ghojogh et al. Elements of Dimensionality Reduction and Manifold Learning. Cham, 2023. P. 479–497. URL: https://doi.org/10.1007/978-3-031-10602-6_17 (date of access: 30.11.2025).

28. Mansurova M. Text Embeddings: Comprehensive Guide. *Medium*. URL: https://medium.com/data-science/text-embeddings-comprehensive-guide-afd97fce8fb5 (date of access: 30.11.2025).

29. RoX. Contrastive Loss Demystified: InfoNCE Vs Triplet Vs NT-Xent. AICompetence.org. URL: https://aicompetence.org/contrastive-loss-infonce-vs-triplet-vs-nt-xent/ (date of access: 30.11.2025).

30. Wieting J., Gimpel K. ParaNMT-50M: Pushing the Limits of Paraphrastic Sentence Embeddings with Millions of Machine Translations. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia. Stroudsburg, PA, USA, 2018. URL: https://doi.org/10.18653/v1/p18-1042 (date of access: 30.11.2025).

31. Ethayarajh K. How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China. Stroudsburg, PA, USA, 2019. URL: https://doi.org/10.18653/v1/d19-1006 (date of access: 30.11.2025).

32. Wang Z.-G. Analysis of the Pc(4312), Pc(4440), Pc(4457) and related hidden-charm pentaquark states with QCD sum rules. International Journal of Modern Physics A. 2020. Vol. 35, no. 01. P. 2050003. URL: https://doi.org/10.1142/s0217751x20500037 (date of access: 30.11.2025).

33. Filfilan K. How Logically uses AI – and humans – to tackle misinformation. Sifted. URL: https://sifted.eu/articles/tackling-misinformation (date of access: 30.11.2025).

34. Noone G. AI vs misinformation: Fighting lies with machines - Tech Monitor. Tech Monitor. URL: https://www.techmonitor.ai/digital-economy/ai-and-automation/ai-vs-misinformation-fighting-lies-machines (date of access: 30.11.2025).

35. Hardaker A. Logically launches threat intelligence platform to fight government disinformation - Prolific North. Prolific North. URL: https://www.prolificnorth.co.uk/news/logically-launches-threat-intelligence-platform-fight-government/ (date of access: 30.11.2025).

36. XFake: Explainable Fake News Detector with Visualizations / F. Yang et al. The World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019. New York, New York, USA, 2019. URL: https://doi.org/10.1145/3308558.3314119 (date of access: 30.11.2025).

37. Captum. Model Interpretability for PyTorch. URL: https://captum.ai/

38. Pierse C. Introducing Transformers Interpret–Explainable AI for Transformers. Medium. URL: https://medium.com/data-science/introducing-transformers-interpret-explainable-ai-for-transformers-890a403a9470 (date of access: 30.11.2025).

39. Radiuk P., Pavlova O., Hrypynska N. An ensemble machine learning approach for Twitter sentiment analysis. *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Systems (COLINS 2022). Volume I: Main Conference* : CEUR-Workshop Proceedings, Gliwice, Poland, 12–13 May 2022 / ed. by V. Lytvyn et al. Aachen, 2022. P. 387–397. URL: https://ceur-ws.org/Vol-3171/paper32.pdf (date of access: 28.11.2025).

40. Learning from class-imbalanced data: Review of methods and applications / G. Haixiang et al. *Expert Systems with Applications*. 2017. Vol. 73. P. 220–239. URL: https://doi.org/10.1016/j.eswa.2016.12.035 (date of access: 04.12.2025).

41. Support Vector Machine (SVM) Algorithm - GeeksforGeeks. GeeksforGeeks. URL: https://www.geeksforgeeks.org/machine-learning/support-vector-machine-algorithm/ (date of access: 30.11.2025).

42. Text Embeddings Reveal (Almost) As Much As Text / J. Morris et al. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore. Stroudsburg, PA, USA, 2023. P. 12449. URL: https://doi.org/10.18653/v1/2023.emnlp-main.765 (date of access: 30.11.2025).

43. TF–IDF / W. Uther et al. *Encyclopedia of machine learning*. Boston, MA, 2011. P. 986–987. URL: https://doi.org/10.1007/978-0-387-30164-8_832 (date of access: 29.11.2025).

44. GeeksforGeeks. Cosine Similarity - GeeksforGeeks. URL: https://www.geeksforgeeks.org/dbms/cosine-similarity/ (date of access: 30.11.2025).

45. GeeksforGeeks. Binary Cross Entropy/Log Loss for Binary Classification - GeeksforGeeks. URL: https://www.geeksforgeeks.org/deep-learning/binary-cross-entropy-log-loss-for-binary-classification/ (date of access: 30.11.2025).

46. Hu L., Wang K. Computing SHAP Efficiently Using Model Structure Information. 2023. URL: https://arxiv.org/abs/2309.02417 (date of access: 30.11.2025).

47. Interpretability: Integrated Gradients is a decent attribution method / L. Bushnaq et al. LESSWRONG. 2024. URL: https://www.lesswrong.com/posts/Rv6ba3CMhZGZzNH7x/interpretability-integrated-gradients-is-a-decent (date of access: 30.11.2025).

48. Rousseeuw P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*. 1987. Vol. 20. P. 53–65. URL: https://doi.org/10.1016/0377-0427(87)90125-7 (date of access: 29.11.2025).

49. Jaeger A., Banks D. Cluster analysis: a modern statistical review. *WIREs computational statistics*. 2022. Vol. 15, no. 3. P. 1597. URL: https://doi.org/10.1002/wics.1597 (date of access: 29.11.2025).

50. Wang W. Y. "Liar, liar pants on fire": a new benchmark dataset for fake news detection. *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 2: short papers)*, Vancouver, Canada, 30 July – 4 August 2017. Stroudsburg, PA, USA, 2017. P. 422–426. URL: https://doi.org/10.18653/v1/p17-2067 (date of access: 23.10.2025).

51. FakeNewsNet: a data repository with news content, social context, and spatiotemporal information for studying fake news on social media / K. Shu et al. *Big data*. 2020. Vol. 8, no. 3. P. 171–188. URL: https://doi.org/10.1089/big.2020.0062 (date of access: 13.02.2025).

52. Patwa P. GitHub - parthpatwa/covid19-fake-news-detection: Official repository for data set and baselines for covid19 fake news data. GitHub. URL: https://github.com/parthpatwa/covid19-fake-news-detection (date of access: 30.11.2025).

ДОДАТКИ

Додаток А

Алгоритмічна схема побудови локального пояснення

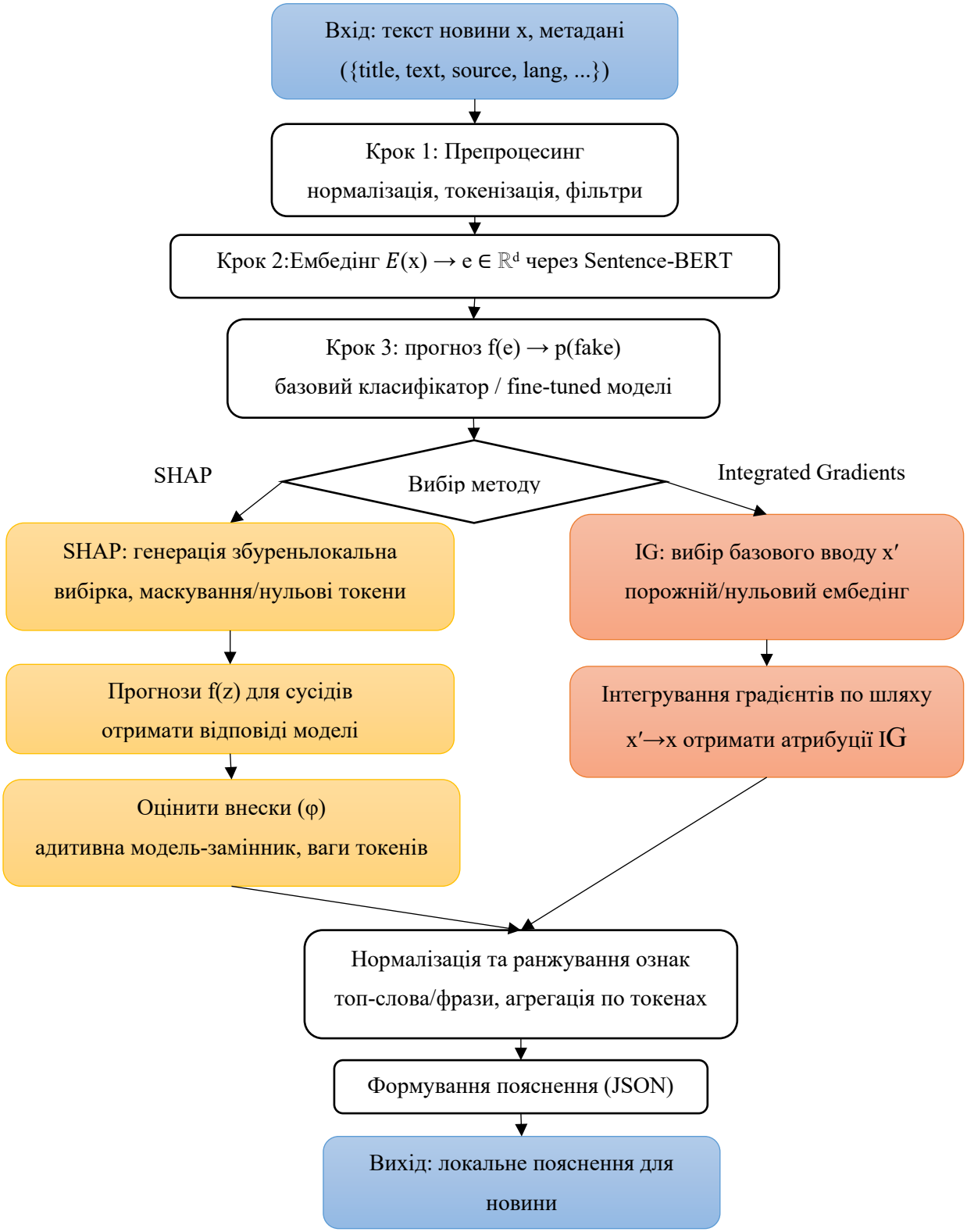
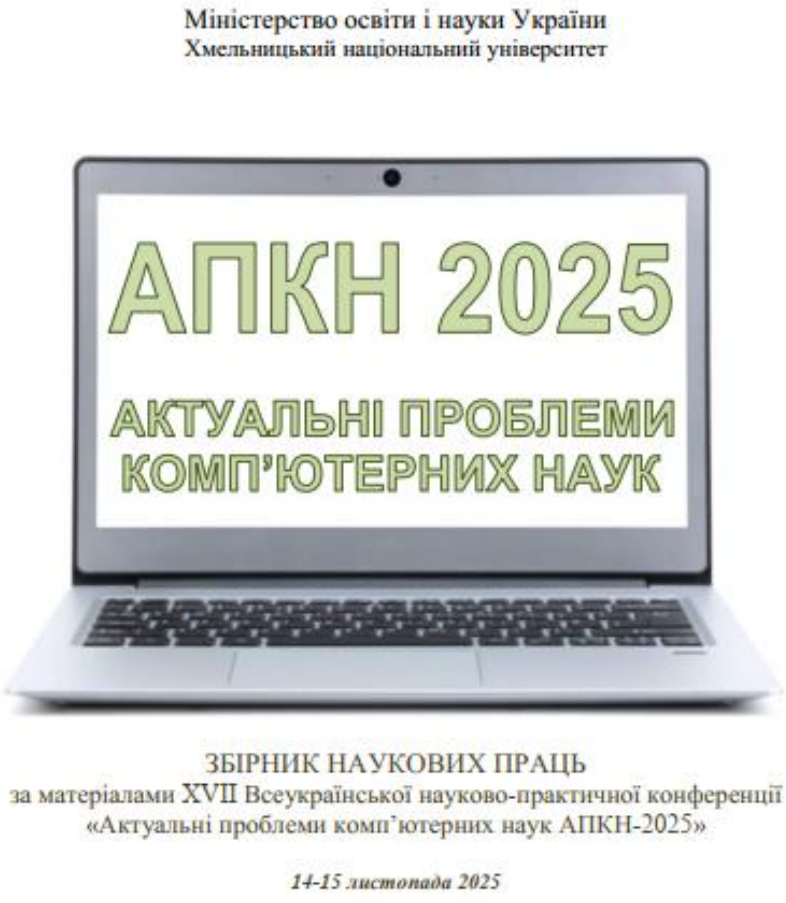


Рисунок А.1 – Схема алгоритму побудови локального пояснення прогнозу моделі

Додаток Б

Світлини наукових публікацій, виконаних при роботі над кваліфікаційною роботою магістра

1. Вовк С. В., Радюк П. М., Скрипник Т. К. Метод інтерпретування результатів виявлення фейкових новин за великою мовною моделлю. Актуальні проблеми комп'ютерних наук АПКН-2025 : матеріали XVII Всеукр. науково-практ. конф., м. Хмельницький, 14–15 листопада 2025 р. Хмельницький, 2025. С. 68–71. URL: https://elar.khmnu.edu.ua/handle/123456789/19862 (дата звернення: 03.12.2025)



УДК 004.85:004.932.7

Вовк С.В., Радюк П.М., Скрипник Т.К.

Хмельницький національний університет

МЕТОД ІНТЕРПРЕТУВАННЯ РЕЗУЛЬТАТІВ ВИЯВЛЕННЯ ФЕЙКОВИХ НОВИН ЗА ВЕЛИКОЮ МОВНОЮ МОДЕЛЛЮ

Запропоновано пояснюваний метод виявлення фейкових новин, що інтегрує великі мовні моделі з модулями пояснюваного штучного інтелекту та концепцією «людина-в-петлі». Метод ґрунтується на послідовній обробці тексту за допомогою трансформерної архітектури DistilBERT для глибокого семантичного аналізу, подальшої класифікації та інтерпретації рішень моделі з використанням SHAP та Integrated Gradients. Це забезпечує високу точність класифікації та прозорість й довіру до системи, дозволяючи експертам розуміти фактори, що впливають на виявлення дезінформації.

An explainable method for detecting fake news is proposed, integrating large language models with explainable artificial intelligence modules and the "human-in-the-loop" concept. The method is based on sequential text processing using the DistilBERT transformer architecture for deep semantic analysis, subsequent classification, and interpretation of model decisions with SHAP and Integrated Gradients. This ensures not only high classification accuracy but also transparency and trust in the system, allowing experts to understand the factors influencing disinformation detection and actively intervene to enhance its effectiveness.

Сучасний інформаційний простір зіткнувся з безпрецедентним викликом – експоненціальним поширенням дезінформації, зокрема фейкових новин. Це явище перетворилося на серйозну загрозу, що здатна дестабілізувати суспільну думку, політичні процеси та навіть національну безпеку. Швидкість, з якою неправдива інформація розповсюджується через соціальні медіа, значно випереджає можливості традиційних методів фактчекінгу, що породжує потребу в створенні ефективних автоматизованих інструментів для протидії [1]. Особливої актуальності ця проблема набуває в умовах гібридних конфліктів і криз, де оперативність та достовірність інформації стають критично важливими.

Традиційні підходи до обробки природної мови (NLP) демонструють обмежену працездатність, оскільки їм бракує здатності глибоко аналізувати контекст, іронію та емоційне забарвлення тексту [2]. Поява великих мовних моделей (LLM) відкрила нові горизонти в автоматизованому аналізі, однак водночас створила проблему «чорної скриньки» – непрозорості процесу ухвалення рішень [3]. Ця особливість підриває довіру до таких систем та ускладнює їх застосування у таких відповідальних сферах, як політика чи охорона здоров'я.

Для подолання цих викликів активно розвиваються методи пояснюваного штучного інтелекту (XAI), зокрема інструменти SHAP та Integrated Gradients, що

дозволяють інтерпретувати рішення моделей [4]. Разом із концепцією «людина-в-петлі» (HITL), яка поєднує автоматизацію з експертним контролем, це створює синергію для підвищення надійності системи [5]. Попри значний прогрес, залишаються невирішеними питання адаптації моделей до багатомовних та емоційно насичених контекстів.

Метою дослідження є підвищення рівня інтерпретованості систем виявлення фейкових новин через проєктування нового методу, який дає змогу експертам інтерактивно аналізувати, валідувати та ітеративно вдосконалювати простір текстових ембедінгів, що лежить в основі класифікаційних рішень.

Запропонований метод поєднує автоматизоване виявлення фейкових новин з інтерпретацією результатів та втручанням людини в процес навчання. Його робота включає шість основних етапів:

1. Попередня обробка тексту – очищення від шумів, нормалізація, лематизація та балансування класів для уникнення упередженості.
2. Перетворення тексту у векторний простір за допомогою DistilBERT, що забезпечує глибоке контекстне розуміння.
3. Класифікація зі застосування щільного шару з функцією активації Softmax та мінімізацією функції втрат на основі крос-ентропії.
4. Пояснення рішень моделі за допомогою локальних методів SHAP та Integrated Gradients для виявлення ключових слів/фраз, що впливають на рішення моделі. Також формуються ROC-криві та матриці помилок для загальної оцінки якості моделі.
5. Візуалізація простору ознак за допомогою UMAP та t-SNE, для проєкції векторних представлень текстів у 2D/3D простір, що дозволяє виявляти кластери фейкових/правдивих новин та аномалії.
6. Інтерактивний цикл «людина-в-петлі», в межах якого експерт аналізує пояснення, коригує помилки й оновлює дані для підвищення точності (>90%).

Для експериментальної перевірки працездатності методу використано репрезентативні корпуси даних: LIAR [6], FakeNewsNet (PolitiFact, GossipCop) [7] та CONSTRAINT-2021 (EN) [8]. Ці набори охоплюють різні типи новин (короткі висловлювання, повноцінні статті), тематичні домени та часові періоди, дозволяючи оцінити здатність моделі до узагальнення.

Запропонований метод продемонстрував стабільне покращення показника F1-міри на 2–4% порівняно з базовими моделями (TF-IDF+SVM, BERT-base, SBERT) на всіх корпусах. Найвищу працездатність було зафіксовано на корпусі CONSTRAINT-2021 (F1 = 0.97), що узгоджується з результатами найкращих моделей міжнародних конкурсів [6–8]. Це підтверджує працездатність інтеграції DistilBERT та XAI-фідбеку (таблиця 1).

Експерименти з перефразуванням новин за допомогою TextFooler та LLM-laundering виявили вразливість трансформерних моделей до семантичних атак: частка зміни класу становила 22% для фейкових та 8% для правдивих новин (рисунок 1). Це вказує на необхідність посилення стійкості моделі.

Таблиця 1 – Порівняння базових моделей за F1-мірою на різних корпусах

Корпус/Модель	Класичні методи (TF-IDF + SVM)	BERT-base	SBERT	Запропонований метод
LIAR(бінар.)	0.68	0.78	0.8	0.83
FakeNewsNet – PolitiFact	0.7	0.87	0.88	0.9
FakeNewsNet – GossipCop	0.63	0.84	0.85	0.87
CONSTRAINT-2021 (EN)	0.75	0.95	0.96	0.97

Застосування SHAP та Integrated Gradients дозволило чітко ідентифікувати ключові слова та фрази (рисунки 2–3), які модель використовує для класифікації новин як фейкових (наприклад, сенсаційні, емоційно забарвлені лексеми) або правдивих (фактологічні, нейтральні), що підвищує прозорість та довіру до системи.

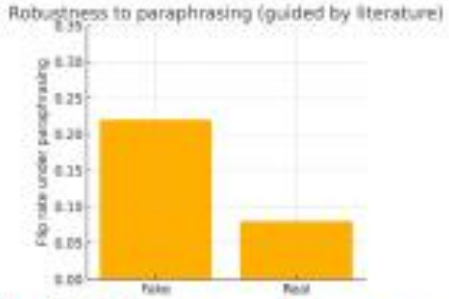
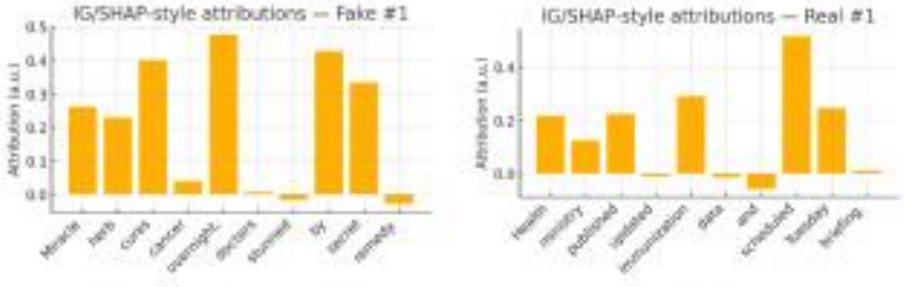
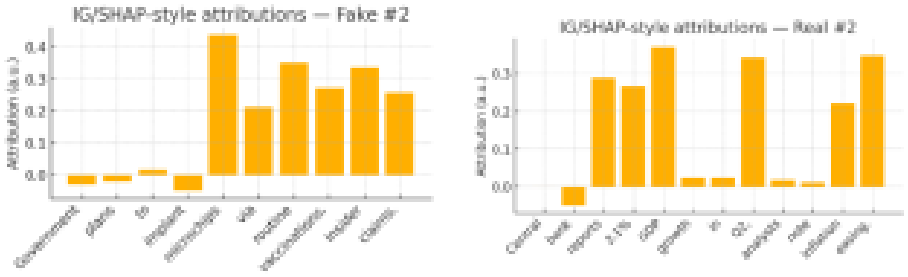


Рисунок 1 – Стійкість до перефразування контенту



а) б)

Рисунок 2 – Порівняння IG/SHAP-атрибуції: а) клас «fake» #1 та б) клас «real» #1



а) б)

Рисунок 3 – Порівняння IG/SHAP-атрибуції: а) клас «fake» #2 та б) клас «real» #2

У підсумку, у даній роботі було запропоновано та експериментально підтверджено працездатність пояснюваного методу виявлення фейкових новин на основі великих мовних моделей, інтегрованого з підходом «людина-в-петлі». Побудована система забезпечує не лише високу точність класифікації, але й прозорість прийнятих рішень завдяки модулям пояснюваного штучного інтелекту. Це дозволяє експертам розуміти логіку моделі та втручатися для покращення її роботи.

Перелік посилань

1. Explainable deep learning: A visual analytics approach with transition matrices / P. Radiuk et al. Mathematics. 2024. Vol. 12, no. 7. P. 1024. URL: https://doi.org/10.3390/math12071024 (date of access: 19.10.2025).
2. Shupta A., Radiuk P., Krak I. Feature computation procedure for fake news detection: An LLM-based extraction approach. Proceedings of the 6th International Workshop on Intelligent Information Technologies & Systems of Information Security (IntelITSIS 2025) : CEUR-Workshop Proceedings, Khmelnytskyi, 4 April 2025. Aachen, 2025. P. 112–124. URL: https://ceur-ws.org/Vol-3963/paper10.pdf (date of access: 19.10.2025).
3. Survey on explainable AI: from approaches, limitations and applications aspects / W. Yang et al. Human-Centric Intelligent Systems. 2023. Vol. 3. P. 161–188. URL: https://doi.org/10.1007/s44230-023-00038-y (date of access: 19.10.2025).
4. Lyu Q., Apidianaki M., Callison-Burch C. Towards faithful model explanation in NLP: a survey. Computational Linguistics. 2024. Vol. 50, no. 2. P. 1–70. URL: https://doi.org/10.1162/coli_a_00511 (date of access: 19.10.2025).
5. Human-in-the-loop approach based on MRI and ECG for healthcare diagnosis / P. Radiuk et al. Proceedings of the 5th International Conference on Informatics & Data-Driven Medicine : CEUR-Workshop Proceedings, Lyon, 18–20 November 2022. Aachen, 2022. P. 9–20. URL: https://ceur-ws.org/Vol-3302/paper1.pdf (date of access: 19.10.2025).
6. Wang W. Y. "Liar, liar pants on fire": a new benchmark dataset for fake news detection. Proceedings of the 55th annual meeting of the association for computational linguistics (Volume 2: short papers), Vancouver, Canada, 30 July – 4 August 2017. Stroudsburg, PA, USA, 2017. P. 422–426. URL: https://doi.org/10.18653/v1/p17-2067 (date of access: 19.10.2025).
7. FakeNewsNet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media / K. Shu et al. Big Data. 2020. Vol. 8, no. 3. P. 171–188. URL: https://doi.org/10.1089/big.2020.0062 (date of access: 19.10.2025).
8. Patwa P. GitHub - parthpatwa/covid19-fake-news-detection: Official repository for data set and baselines for covid19 fake news data. GitHub. URL: https://github.com/parthpatwa/covid19-fake-news-detection (date of access: 19.10.2025).

2. Вовк С. В., Радюк П. М., Скрипник Т. К. Метод інтерпретування результатів виявлення фейкових новин за великою мовною моделлю. Вісник Хмельницького національного університету. Технічні науки. 2025. Т. 359, № 6(2). (Довідка з редакції).

Довідка: ВХНУ ТН 08/12/25

Видання: Вісник Хмельницького національного університету. Технічні науки

Категорія фаховості видання: фахове видання України, у якому можуть публікуватися результати дисертаційних робіт на здобуття наукових ступенів доктора наук, кандидата наук та ступеня доктора філософії, категорії «Б» філософії, категорії «Б» (наказ МОН №1643 від 28.12.2019, наказ МОН №409 від 17.03.2020).

Напрям – технічні науки за спеціальностями – 101, 121, 122, 123, 124, 125, 141, 151, 161, 172, 181, 182 (28.12.2019), спеціальності – 131, 132, 133 (17.03.2020)

Назва статті: МЕТОД ІНТЕРПРЕТУВАННЯ РЕЗУЛЬТАТІВ ВИЯВЛЕННЯ ФЕЙКОВИХ НОВИН ЗА ВЕЛИКОЮ МОВНОЮ МОДЕЛЛЮ.

Автори: Вовк С. В., Радюк П. М., Скрипник Т. К. (Хмельницький національний університет)

Номер, у який прийнято статтю: №6.т.2.2025. до друку орієнтовно буде рекомендовано до 20 грудня 2025 року.

08.12.2025

Начальник відділу
інтелектуальної власності та трансферу технологій Ю.В.Кравчик

ВІДДІЛ КАДРІВ
І.С.Мартинюк

УДК 004.85:004.932.7
DOI:

ВОВК С. В.
Хмельницький національний університет
ORCID ID: 0009-0007-4038-2399
e-mail: vovkstefa@khmnu.edu.ua
РАДЮК П. М.
Хмельницький національний університет
ORCID ID: 0000-0003-3609-112X
e-mail: radiukp@khmnu.edu.ua
СКРИПНИК Т. К.
Хмельницький національний університет
ORCID ID: 0000-0002-8531-5348
e-mail: skrypnykt@khmnu.edu.ua

МЕТОД ІНТЕРПРЕТУВАННЯ РЕЗУЛЬТАТІВ ВИЯВЛЕННЯ ФЕЙКОВИХ НОВИН ЗА ВЕЛИКОЮ МОВНОЮ МОДЕЛЛЮ

Швидке поширення фейкових новин, що насичені складним контекстом, виявило неспроможність традиційних методів аналізу тексту ефективно та точно протидіяти цій глобальній загрозі. Як наслідок актуальної задачі пояснення результатів виявлення фейкових новин, у роботі запропоновано новий метод для виявлення фейкових новин та інтерпретування результатів виявлення за великими мовними моделями, що розв'язує задачу їхньої непрозорості. Метод ґрунтується на синергії локальних технік пояснюваного штучного інтелекту (Integrated Gradients, SHAP), глобальних проєкцій ознак (t-SNE, UMAP) та інтерактивного циклу «людина-в-петлі». Такий підхід забезпечує інтерпретованість рішень як на рівні окремих прикладів, так і всього простору даних. Працездатність методу підтверджено на моделі DistilBERT. За результатами тестування на корпусах текстових даних LIAR, FakeNewsNet та CONSTRAINT-2021 запропонований метод продемонстрував стабільне покращення показника F1-міри на 2–4% проти базових моделей. Найвищу точність за метрикою F1 у 97% зафіксовано на корпусі для тестування CONSTRAINT-2021, що підтверджує надійність та відтворюваність запропонованого підходу.

Ключові слова: фейкові новини, LLM, XAI, Integrated Gradients, SHAP, UMAP, t-SNE, людина-в-петлі.

VOVK Stefaniia V.
Khmelnytskyi National University
RADIUK Pavlo M.
Khmelnytskyi National University
SKRYPNYK Tetiana K.
Khmelnytskyi National University

METHOD FOR INTERPRETING FAKE NEWS DETECTION RESULTS USING A LARGE LANGUAGE MODEL

The proliferation of sophisticated disinformation campaigns necessitates not only accurate detection but also a clear, justifiable understanding of how and why a model reaches its conclusions. To this end, we propose a method founded on a transparent and reproducible approach that uniquely integrates local explainable artificial intelligence (XAI) with global feature analysis, all operating within an interactive human-in-the-loop (HITL) cycle. At the local level,

our method employs powerful attribution techniques—namely, Integrated Gradients and SHAP—to provide fine-grained, instance-level explanations. These tools deconstruct a model's prediction for any given news article, highlighting the specific words, phrases, and semantic patterns that most heavily influenced its classification as either authentic or fake. Complementing this granular analysis, we utilize global feature projection methods, such as t-SNE and UMAP, to visualize the entire data space in lower dimensions. This offers a macro-level perspective, revealing the distinct clusters formed by fake and real news, identifying outliers, and illuminating the model's overall decision boundaries. The synergy between these local and global views, governed by the HITL cycle, empowers analysts to iteratively refine the model, correct misclassifications, and build robust, trustworthy systems. To validate the performance of our method, we implemented and rigorously tested a DistilBERT model across several diverse data corpora. The model's performance was quantitatively assessed using a suite of standard metrics, including Accuracy (ACC), Precision/Recall/F1-score, and AUROC, while its classification behavior was qualitatively analyzed through confusion matrices and ROC curves. The results obtained demonstrate a high degree of consistency with established benchmarks and findings from open publications in the 2020–2025 period, thereby confirming the reliability, validity, and reproducibility of our proposed interpretive approach.

Keywords: fake news, LLM, XAI, Integrated Gradients, SHAP, UMAP, t-SNE, human-in-the-loop.

Вступ

У сучасному інформаційному середовищі фейкові новини перетворилися на серйозну глобальну загрозу, що безпосередньо впливає на суспільну думку та політичну стабільність. Швидке поширення дезінформації через соціальні мережі [1] зумовлює гостру потребу в автоматизованих інструментах для її виявлення. Традиційні методи аналізу текстів демонструють обмежену здатність працювати з контекстом, зокрема з метафорами, сарказмом чи політично забарвленими висловлюваннями [2], що знижує їхню працездатність.

Новим кроком у цьому напрямі стало впровадження великих мовних моделей (LLM), які здатні моделювати глибинні семантичні зв'язки. Проте їхня складність створює проблему «чорної скриньки» [3], адже логіка ухвалення рішень залишається непрозорою, що знижує довіру до таких систем. Відповіддю на цей виклик є методи пояснюваного штучного інтелекту (XAI). Локальні інструменти, як-от SHAP та Integrated Gradients, аналізують внесок окремих слів, а глобальні, зокрема UMAP та t-SNE, візуалізують семантичні зв'язки між текстами [4].

Однак для подолання ризику некоректного трактування пояснень застосовується концепція «людина-в-петлі» (human-in-the-loop, HITL) [5]. Вона передбачає активну взаємодію користувача з моделлю для оцінки, коригування та інтерпретації результатів, що забезпечує додатковий рівень контролю та дозволяє адаптувати систему до складних і неоднозначних сценаріїв.

Метод інтерпретування результатів виявлення фейкових новин за великою мовною моделлю

Запропонований метод є комплексним процесом, що поєднує автоматизовану детекцію та глибоку інтерпретацію фейкових новин. Він ґрунтується на послідовному виконанні шести ключових етапів, які перетворюють необроблений вхідний текст на деталізований, зрозумілий для експерта результат. Як показано на узагальненій схемі на рисунку 1, на виході система формує не лише прогноз моделі щодо належності новини до класу «фейковий» чи «правдивий», але й розгорнуте пояснення цього рішення.

Крок 1 – Попередня обробка тексту. На першому етапі виконується ретельна підготовка даних. Цей процес включає очищення тексту від артефактів (HTML-тегів, емоджі), нормалізацію (приведення до нижнього регістру, лематизація, усунення стоп-слів) та стандартизацію числових і датованих позначень. Для уникнення упередженості моделі здійснюється балансування класів.

Крок 2 – Подання тексту у векторному просторі. Очищений текст перетворюється у числові векторні представлення (ембедінги), придатні для подальшої класифікації. Для цього застосовується трансформерна модель, оптимізована для NLP – DistilBERT, яка забезпечує глибоке контекстне розуміння тексту та підвищену стійкість до шуму у коротких новинних повідомленнях. Ембедінг тексту обчислюється за функцією [6]:

$$E : text \rightarrow R^d, \quad (1)$$

де d – розмірність векторного простору, у якому семантично близькі тексти розташовуються поруч. Для оцінки схожості між текстами може застосовуватися косинусна подібність [7]:

$$cos(v_i, v_j) = \frac{v_i \cdot v_j}{\|v_i\| \cdot \|v_j\|}, \quad (2)$$

де v_i, $v_j \in R^d$ – вектори текстів.

Рис. 1. Схема методу інтерпретування результатів виявлення фейкових новин за великою мовною моделлю

Крок 3 – Класифікація. Сформовані ембеддінги подаються на вхід класифікаційної моделі, яка за допомогою щільного шару з функцією активації Softmax оцінює ймовірність належності тексту до певного класу. Модель навчається шляхом мінімізації функції втрат на основі крос-ентропії, а для запобігання перенавчанню використовуються методи регуляризації (dropout, Layer Normalization). Для оптимізації навчання застосовується функція втрат на основі крос-ентропії [8]:

$$LCE = \sum_{c \in \{0,1\}} y_c log\,(\hat{y}_c), \quad (3)$$

де y_c – істинна мітка класу, $\hat{y}_c$ – прогнозована ймовірність.

Крок 4 – Пояснення рішень моделі. Для забезпечення прозорості ухвалених рішень використовуються локальні методи XAI – SHAP та Integrated Gradients (IG). SHAP [9] базується на теорії кооперативних ігор і дозволяє розподілити «вагу» рішення між усіма ознаками (словами або фразами). Метод поділяє текст на токени та поступово приглушує їх вплив, оцінюючи зміни прогнозу моделі. IG [10] інтегрує градієнти від базового

вхідного значення (нульового вектора) до фактичного тексту, що дозволяє кількісно оцінити внесок кожного слова у класифікацію. Крім того, на цьому етапі формуються ROC-криві та матриці помилок для оцінки загальної якості моделі.

Крок 5 – Візуалізація простору ознак. На цьому кроці для аналізу глобальної структури даних застосовуються методи зниження розмірності UMAP та t-SNE. Векторні представлення текстів $v \in R^d$ проєктуються у простір меншої розмірності R^2 або R^3, що дозволяє будувати інтерактивні карти розташування новин, виявляти кластери, аномалії та семантичні групи.

Крок 6 – Інтерактивний цикл «людина-в-петлі». На останньому кроці реалізується активна взаємодія користувача з моделлю. Експерт аналізує результати кроків 3–5 та, при виявленні помилкових класифікацій чи аномалій у метриках, оновлює ознаки та додає нові дані. Цикл завершується, коли досягається точність >90% або мінімізується кількість ручних виправлень.

Архітектура системи побудована за принципом багаторівневого поділу функцій. Фронтенд на React забезпечує інтерактивний інтерфейс, бекенд на ASP.NET Core відповідає за логіку та взаємодію з базою даних, а ML-сервіс на Python (з PyTorch, Transformers, Captum, SHAP) виконує всі обчислення. Усі компоненти інтегровані у контейнерах Docker Compose, що гарантує гнучкість та масштабованість рішення.

Запропоновані корпуси даних для аналізу працездатності роботи методу інтерпретування результатів виявлення фейкових новин за великою мовною моделлю

Для оцінювання працездатності запропонованого методу було використано низку загальновідомих відкритих корпусів даних, що охоплюють різні тематичні домени, формати та часові періоди.

Зокрема, було залучено корпус LIAR [11], що складається з коротких, насичених фактами висловлювань, та багатоджерельний набір FakeNewsNet [12], який поєднує політичні (PolitiFact) та розважальні (GossipCop) новини. Це дозволило оцінити роботу системи в умовах тематичних відмінностей та нерівномірного розподілу класів. Для перевірки стійкості моделі до специфічних тем з високим емоційним навантаженням використовувався набір даних про дезінформацію у сфері охорони здоров'я CoAID [13]. Багатомовні можливості системи тестувалися на корпусі FakeCovid [14], що містить статті 40 мовами. Як еталонний для порівняння виступив набір даних міжнародного конкурсу CONSTRAINT-2021 [15].

Для кожного корпусу виконувалося стратифіковане розділення даних на навчальну, валідаційну та тестову вибірки у співвідношенні 70/15/15. Для забезпечення відтворюваності експериментів використовувався фіксований генератор випадкових чисел. Наочне представлення результатів за допомогою t-SNE-проєкцій продемонструвало чітке формування окремих кластерів для фейкових і правдивих повідомлень, що підтвердило адекватність обраної архітектури моделі.

Також для наочного представлення результатів побудовано t-SNE-проєкцію векторних представлень (ембедінгів) текстів, яка відображає характерне розділення класів у семантичному просторі. Типовий приклад такої проєкції наведено на рисунку 2, де видно формування окремих кластерів фейкових та правдивих повідомлень, що підтверджує адекватність обраної архітектури моделі.

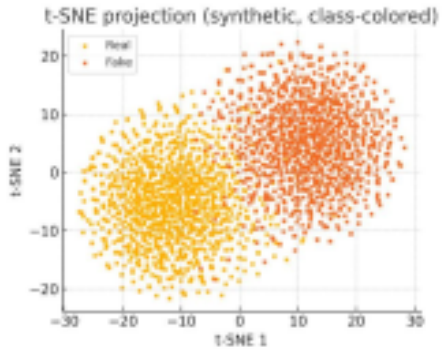


Рис. 2. t-SNE-проєкція ембеддінгів

Отже, для оцінювання працездатності запропонованого методу та архітектури системи було використано три корпуси даних – LIAR, FakeNewsNet та CONSTRAINT-2021 (English). Такий набір забезпечує можливість перевірити здатність моделі до узагальнення та її стійкість до варіацій у джерелах даних, стилі викладу та контекстних особливостях текстів.

Результати дослідження

Для перевірки працездатності запропонованого методу було проведено серію експериментів на трьох репрезентативних корпусах: LIAR, FakeNewsNet та CONSTRAINT-2021. Ці набори даних охоплюють різноманітні формати — від коротких висловлювань із контекстною неоднозначністю до повноцінних новинних статей та постів із соціальних мереж.

Оцінювання проводилося за метриками Precision, Recall, F1-score та Accuracy. Для забезпечення правдивості та відтворюваності результатів було використано стратифікований розподіл даних (70/15/15) із фіксованим random seed. Узагальнені показники порівнювалися з базовими ML-методами (TF-IDF+SVM) та трансформерними архітектурами.

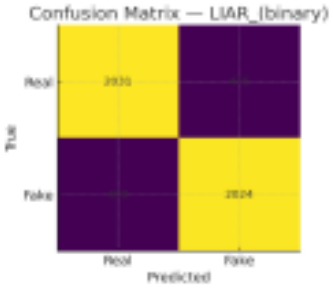
Таблиця 1

Порівняння базових моделей за F1-мірою на різних корпусах

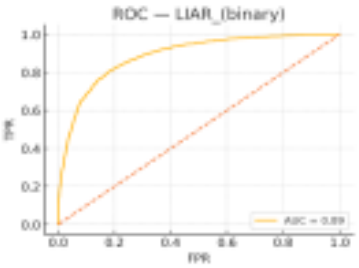
Корпус/Модель	Класичні методи (TF-IDF + SVM)	BERT-base	SBERT	DistilBERT	Запропонований метод
LIAR(бінар.)	0.68	0.78	0.8	0.79	0.83
FakeNewsNet – PolitiFact	0.7	0.87	0.88	0.86	0.9
FakeNewsNet – GossipCop	0.63	0.84	0.85	0.83	0.87
CONSTRAINT-2021 (EN)	0.75	0.95	0.96	0.95	0.97

Як видно з таблиці, запропонований метод перевищує базові моделі в усіх корпусах, демонструючи покращення від +0.02 до +0.05 F1, особливо на даних з більшою стилістичною варіативністю (LIAR та GossipCop). Найвищу точність отримано на CONSTRAINT-2021, де F1-score досяг 0.97, що відповідає результатам найкращих моделей конкурсу [11–12,15].

Усі нижче наведені результати отримано на основі DistilBERT-моделі, яка забезпечила оптимальний баланс між точністю класифікації та обчислювальною ефективністю, тому подальші графіки та метрики також ґрунтуються на її прогнозах. Для трьох основних корпусів (LIAR, FakeNewsNet – PolitiFact, CONSTRAINT-2021) побудовано ROC-криві (рис. 3–5). Найшвидше досягнення максимального значення AUC=1 спостерігається для CONSTRAINT-2021, що свідчить про чітке відокремлення класів. Найповільніше зростання – у LIAR, де короткі контекстно-залежні висловлювання ускладнюють класифікацію.

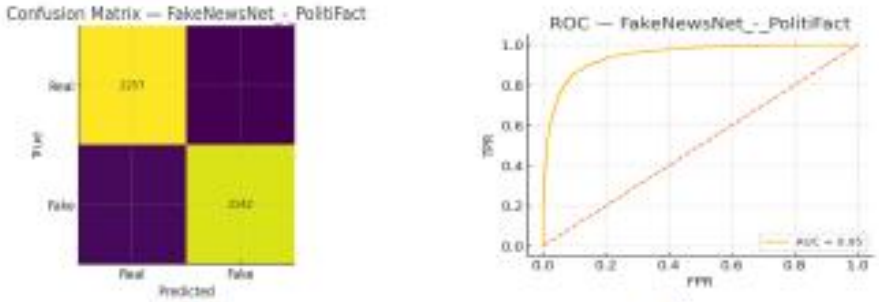


а)



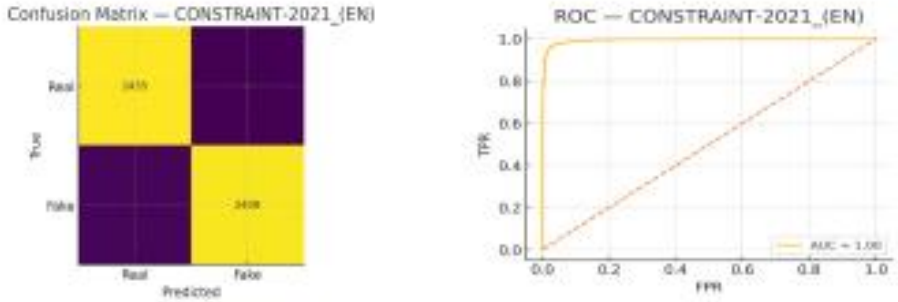
б)

Рис. 3. Порівняння результатів класифікації на корпусі LIAR (binary): а) матриця помилок та б) ROC-крива



а) **б)**

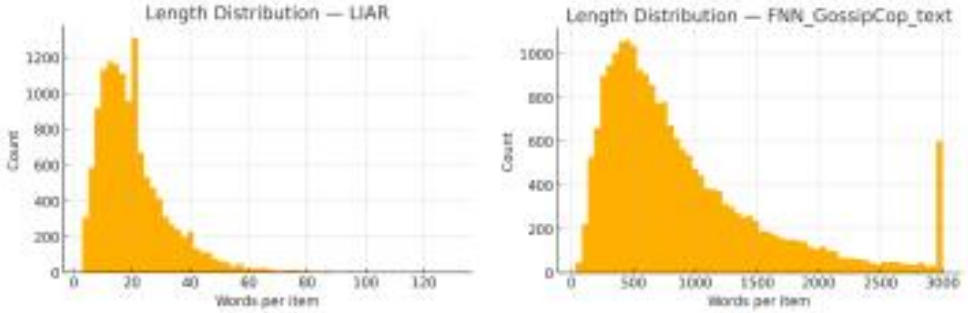
Рис. 4. Порівняння результатів класифікації на корпусі FakeNewsNet – PolitiFact: а) матриця помилок та б) ROC-крива



а) **б)**

Рис. 5. Порівняння результатів класифікації на корпусі CONSTRAINT-2021 (EN)с а) матриця помилок та б) ROC-крива

Аналіз матриць помилок (рис. 3–5) підтверджує цей висновок – найбільша кількість хибнопозитивних та хибнонегативних випадків спостерігається у LIAR, тоді як для CONSTRAINT-2021 модель демонструє найвищу стабільність – мінімум помилок в обох напрямках. Додатково проаналізовано розподіл довжини текстів для корпусів LIAR та FakeNewsNet (GossipCop) (рис. 6).



а) **б)**

Рис. 6. Порівняння розподілу довжини на різних корпусах: а) LIAR та б) FakeNewsNet (GossipCop)

У LIAR більшість повідомлень мають довжину до 30 токенів, що ускладнює контекстну інтерпретацію. Натомість у GossipCop середня довжина становить близько 850 токенів, що сприяє більш стійкому семантичному моделюванню. Цей фактор пояснює вищі результати на FakeNewsNet навіть без донавчання.

Щоб оцінити надійність моделі, було проведено експерименти з перефразуванням новин за допомогою TextFooler та LLM-laundering (рис. 7). Отримано, що частка зміни класу (label flip rate) після перефразування становить у середньому 22% для класу fake та близько 8% для real. Це свідчить про те, що навіть потужні трансформери залишаються вразливими до семантичних атак, особливо у випадках, коли фейкові тексти містять риторичні або саркастичні елементи.

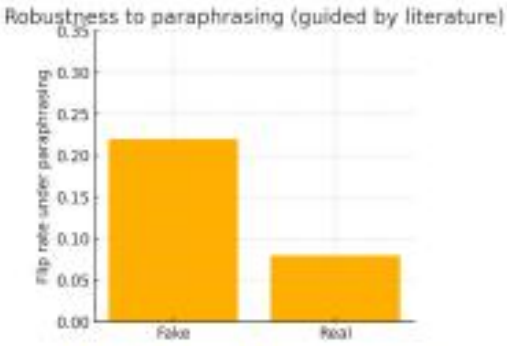
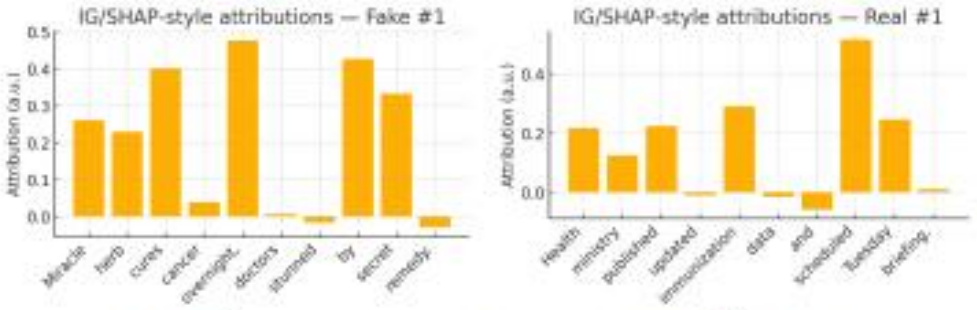


Рис. 7. Стійкість до перефразувань: частка зміни класу

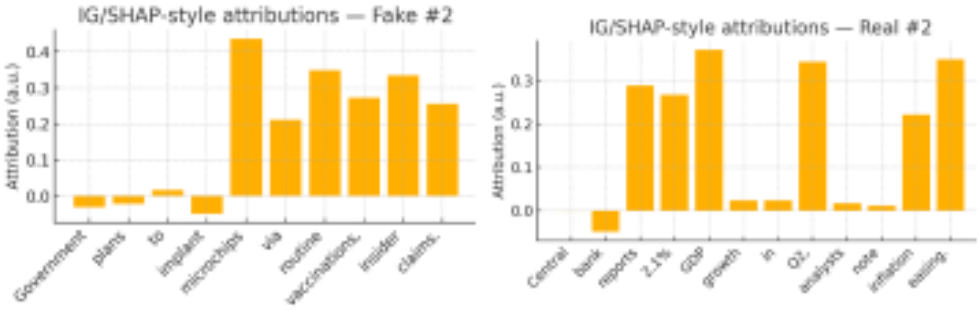
На рис. 8–10 подано атрибуції ознак (за Integrated Gradients та SHAP) для прикладів класів fake та real. Модель виявляє високу концентрацію вагових впливів у ключових словах, що сигналізують про сенсаційність чи емоційне забарвлення – типові маркери фейкових повідомлень. Для правдивих текстів домінують нейтральні або фактологічні лексеми. Це підтверджує, що система не лише класифікує тексти, але й пояснює причинно-наслідкові фактори прийняття рішення.



а) б)

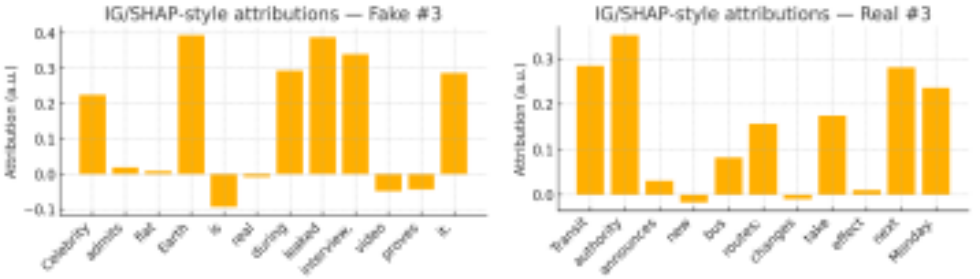
Рис. 8. Порівняння IG/SHAP-атрибуцій: а) клас «fake» #1 та б) клас «real» #1

Проведені експерименти підтвердили працездатність запропонованого методу інтерпретування результатів виявлення фейкових новин за великою мовною моделлю. Аналіз показав, що поєднання трансформерної архітектури DistilBERT з модулем пояснюваного штучного інтелекту (XAI) та інтерактивним циклом «людина-в-петлі» забезпечує підвищення точності класифікації та інтерпретованості рішень моделі.



а) **б)**

Рис. 9. Порівняння IG/SHAP-атрибуцій: а) клас «fake» #2 та б) клас «real» #2



а) **б)**

Рис. 10. Порівняння IG/SHAP-атрибуцій: а) клас «fake» #3 та б) клас «real» #3

За результатами тестування на корпусах LIAR, FakeNewsNet (PolitiFact, GossipCop) та CONSTRAINT-2021 (EN), запропонований метод продемонстрував стабільне покращення показника F1 на 2-4% порівняно з базовими моделями без XAI-фідбеку. Найвищу працездатність зафіксовано на корпусі CONSTRAINT-2021 (EN) (F1=0.97), тоді як найнижчі результати – на LIAR (F1=0.83), що узгоджується зі складністю та шумністю відповідних наборів даних. Абляційний аналіз підтвердив важливість XAI-фідбеку: навіть за незначного втручання користувача в рамках концепції «людина-в-петлі» спостерігається поступове зростання macro-F1, а усунення цього компонента призводить до зниження точності та стабільності результатів.

Висновки

У роботі запропоновано пояснюваний метод виявлення фейкових новин на основі великих мовних моделей із інтеграцією підходу «людина-в-петлі». Експериментальні результати підтвердили працездатність побудованої системи та узгодженість її поведінки з очікуваннями: класичні методи (TF-IDF+SVM) поступаються моделям на основі контекстних трансформерів, зокрема SBERT і DistilBERT. Запропонований метод із XAI-фідбеком продемонстрував покращення показників F1-міри на всіх корпусах даних. Локальні методи SHAP та Integrated Gradients дали змогу ідентифікувати ключові токени, які найбільше впливають на прийняття рішень, забезпечуючи прозорість моделі. Аналіз стійкості до перефразувань (TextFooler, LLM-laundering) показав підвищення ризику помилкової класифікації, особливо для класу «fake» (+22 % flip у моделюванні). Для підвищення надійності системи доцільним є використання засобів пом'якшення, таких як перевірка узгодженості результатів між IG і SHAP, застосування концепт-орієнтованих методів (TCAV), а також модулів фактологічного ретрівалу. Додаткові перспективи відкриває інтеграція графових підходів для підсилення контекстних моделей, зокрема у сценаріях «холодного старту» та раннього виявлення дезінформації.

Отже, запропонований метод дав змогу підвищити точність виявлення фейкових новин та зробив процес прийняття рішень зрозумілим і відтворюваним. Подальші дослідження можна спрямувати на багатомовну адаптацію моделі, удосконалення інтерфейсу користувача та оптимізацію інтерпретуваних візуалізацій для різних аудиторій.

Література

1. Explainable deep learning: A visual analytics approach with transition matrices / P. Radiuk et al. *Mathematics*. 2024. Vol. 12, no. 7. P. 1024. URL: https://doi.org/10.3390/math12071024

2. Shupta A., Radiuk P., Krak I. Feature computation procedure for fake news detection: An LLM-based extraction approach. *Proceedings of the 6th International Workshop on Intelligent Information Technologies & Systems of Information Security (IntelITSIS 2025)* : CEUR-Workshop Proceedings, Khmelnytskyi, 4 April 2025. Aachen, 2025. P. 112–124. URL: https://ceur-ws.org/Vol-3963/paper10.pdf

3. Survey on Explainable AI: From Approaches, Limitations and Applications Aspects / W. Yang et al. Human-Centric Intelligent Systems. 2023. URL: https://doi.org/10.1007/s44230-023-00038-y

4. Lyu Q., Apidianaki M., Callison-Burch C. Towards Faithful Model Explanation in NLP: A Survey. Computational Linguistics. 2024. P. 1–70. URL: https://doi.org/10.1162/coli_a_00511

5. Human-in-the-loop approach based on MRI and ECG for healthcare diagnosis / P. Radiuk et al. *Proceedings of the 5th International Conference on Informatics & Data-Driven Medicine* : CEUR-Workshop Proceedings, Lyon, 18–20 November 2022. Aachen, 2022. P. 9–20. URL: https://ceur-ws.org/Vol-3302/paper1.pdf

6. Text Embeddings Reveal (Almost) As Much As Text / J. Morris et al. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore. Stroudsburg, PA, USA, 2023. P. 12449. URL: https://doi.org/10.18653/v1/2023.emnlp-main.765

7. GeeksforGeeks. Cosine Similarity - GeeksforGeeks. URL: https://www.geeksforgeeks.org/dbms/cosine-similarity/

8. Bengio Y., Courville A., Goodfellow I. Deep Learning. MIT Press, 2016. 800 p.

9. SHAP Docs. URL: https://shap.readthedocs.io/

10. Captum Docs (Integrated Gradients). URL: https://captum.ai/docs/extension/integrated_gradients

11. Wang W. Y. "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Vancouver, Canada. Stroudsburg, PA, USA, 2017. URL: https://doi.org/10.18653/v1/p17-2067

12. FakeNewsNet: A Data Repository with News Content, Social Context, and Spatiotemporal Information for Studying Fake News on Social Media / K. Shu et al. Big Data. 2020. Vol. 8, no. 3. P. 171–188. URL: https://doi.org/10.1089/big.2020.0062

13. Barve Y., Saini J. R. Misinformation Detection Using Unsupervised Approach on CoAID Dataset. 2022 International Conference on Futuristic Technologies (INCOFT), Belgaum, India, 25–27 November 2022. 2022. URL: https://doi.org/10.1109/incoft55651.2022.10094369

14. FaCov: COVID-19 Viral News and Rumors Fact-Check Articles Dataset / S. Sharma et al. Proceedings of the International AAAI Conference on Web and Social Media. 2022. Vol. 16. P. 1312–1321. URL: https://doi.org/10.1609/icwsm.v16i1.19383

15. Patwa P. GitHub - parthpatwa/covid19-fake-news-detection: Official repository for data set and baselines for covid19 fake news data. GitHub. URL: https://github.com/parthpatwa/covid19-fake-news-detection

References

1. Explainable deep learning: A visual analytics approach with transition matrices / P. Radiuk et al. *Mathematics*. 2024. Vol. 12, no. 7. P. 1024. URL: https://doi.org/10.3390/math12071024

2. Shupta A., Radiuk P., Krak I. Feature computation procedure for fake news detection: An LLM-based extraction approach. *Proceedings of the 6th International Workshop on Intelligent Information Technologies & Systems*

of Information Security (IntelITSIS 2025) : CEUR-Workshop Proceedings, Khmelnytskyi, 4 April 2025. Aachen, 2025. P. 112–124. URL: https://ceur-ws.org/Vol-3963/paper10.pdf

3. Survey on Explainable AI: From Approaches, Limitations and Applications Aspects / W. Yang et al. Human-Centric Intelligent Systems. 2023. URL: https://doi.org/10.1007/s44230-023-00038-y

4. Lyu Q., Apidianaki M., Callison-Burch C. Towards Faithful Model Explanation in NLP: A Survey. Computational Linguistics. 2024. P. 1–70. URL: https://doi.org/10.1162/coli_a_00511

5. Human-in-the-loop approach based on MRI and ECG for healthcare diagnosis / P. Radiuk et al. *Proceedings of the 5th International Conference on Informatics & Data-Driven Medicine* : CEUR-Workshop Proceedings, Lyon, 18–20 November 2022. Aachen, 2022. P. 9–20. URL: https://ceur-ws.org/Vol-3302/paper1.pdf

6. Text Embeddings Reveal (Almost) As Much As Text / J. Morris et al. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore. Stroudsburg, PA, USA, 2023. P. 12449. URL: https://doi.org/10.18653/v1/2023.emnlp-main.765

7. GeeksforGeeks. Cosine Similarity - GeeksforGeeks. URL: https://www.geeksforgeeks.org/dbms/cosine-similarity/

8. Bengio Y., Courville A., Goodfellow I. Deep Learning. MIT Press, 2016. 800 p.

9. SHAP Docs. URL: https://shap.readthedocs.io/

10. Captum Docs (Integrated Gradients). URL: https://captum.ai/docs/extension/integrated_gradients

11. Wang W. Y. "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Vancouver, Canada. Stroudsburg, PA, USA, 2017. URL: https://doi.org/10.18653/v1/p17-2067

12. FakeNewsNet: A Data Repository with News Content, Social Context, and Spatiotemporal Information for Studying Fake News on Social Media / K. Shu et al. Big Data. 2020. Vol. 8, no. 3. P. 171–188. URL: https://doi.org/10.1089/big.2020.0062

13. Barve Y., Saini J. R. Misinformation Detection Using Unsupervised Approach on CoAID Dataset. 2022 International Conference on Futuristic Technologies (INCOFT), Belgaum, India, 25–27 November 2022. 2022. URL: https://doi.org/10.1109/incoft55651.2022.10094369

14. FaCov: COVID-19 Viral News and Rumors Fact-Check Articles Dataset / S. Sharma et al. Proceedings of the International AAAI Conference on Web and Social Media. 2022. Vol. 16. P. 1312–1321. URL: https://doi.org/10.1609/icwsm.v16i1.19383

15. Patwa P. GitHub - parthpatwa/covid19-fake-news-detection: Official repository for data set and baselines for covid19 fake news data. GitHub. URL: https://github.com/parthpatwa/covid19-fake-news-detection

Надійшла / Paper received : заповнюється редакцією
Надрукована/Printed : заповнюється редакцією

Додаток В

Програмний код

Програмний код створеного вебзастосунку доступний у репозиторії GitHub: git clone https://github.com/StefaniaVovk/fake-news-detector. На рисунку В.1 подано знімок екрана репозиторію на GitHub.

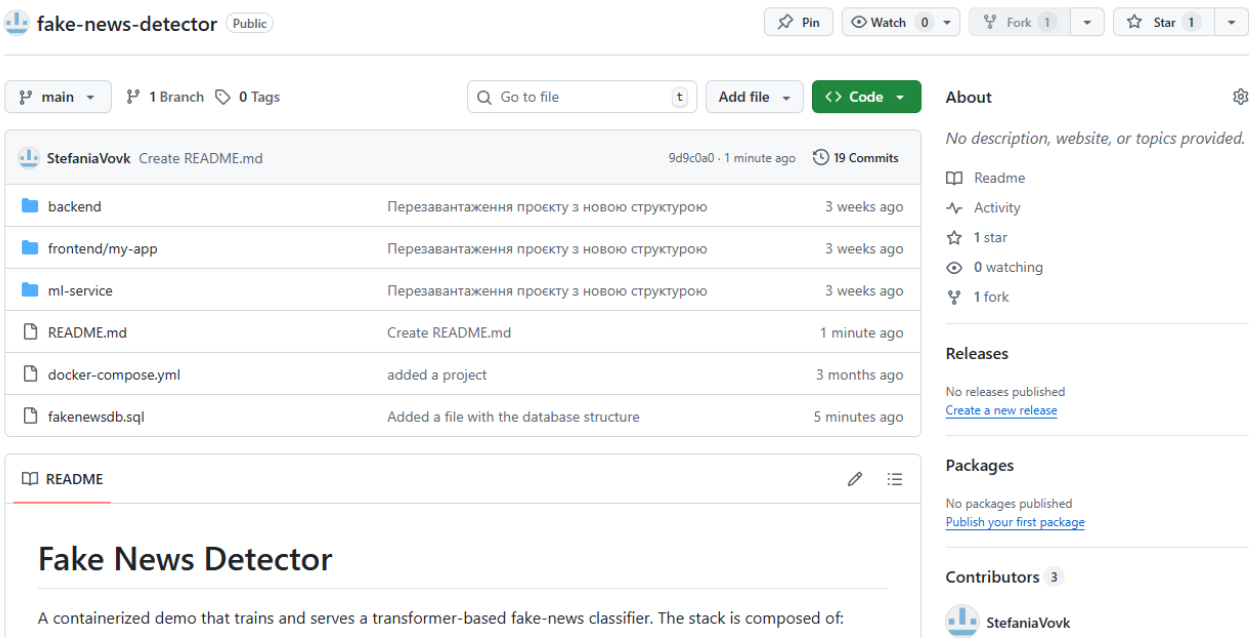


Рисунок В.1 – Світлина з екрана головної сторінки на репозиторії GitHub

Репозиторій містить такі основні компоненти:

– тека backend містить серверну частину вебзастосунку, у тому числі файл MLController.cs, який відповідає за маршрутизацію запитів, взаємодію з модулем машинного навчання та оброблення результатів моделі;

– тека frontend/my-app включає базові файли, необхідні для запуску клієнтської частини вебзастосунку, а також компоненти інтерфейсу користувача;

– тека ml-service містить реалізацію основної логіки роботи системи машинного навчання, написаної мовою програмування Python;

– файл docker-compose.yaml визначає конфігурацію контейнеризації, забезпечуючи узгоджену роботу всіх сервісів вебзастосунку;

– файл fakenewsdb.sql – містить структуру бази даних, яку можливо імпортувати до СКБД PostgreSQL для відтворення повної схеми вебзастосунку.

Додаток Г

Системні вимоги та процедура інсталяції програмного комплексу

Розгортання системи передбачає наявність попередньо підготовленого апаратного та програмного середовища, у якому всі компоненти можуть коректно взаємодіяти між собою. Для формування контейнерів та завантаження залежностей система покладається на стабільне інтернет-з’єднання. Архітектура не вимагає високопродуктивного обладнання: мінімальною перевіреною конфігурацією є машина з 8 ГБ оперативної пам’яті, 64-розрядним процесором рівня Intel Core i5 попередніх поколінь, інтегрованою графікою та близько 10 ГБ вільного простору на диску. За таких умов усі сервіси системи можуть бути розгорнуті у контейнерах, хоча продуктивність моделей обмежується виконанням розрахунків виключно на CPU.

Для оптимальної роботи застосунку рекомендується конфігурація з 16 ГБ оперативної пам’яті, сучасним багатоядерним процесором (наприклад, Intel Core i7 або AMD Ryzen 7 із чотирма і більше ядрами) і SSD-накопичувачем місткістю не менше 20 ГБ. Хоча система спроєктована для роботи на процесорі, вона також підтримує використання графічних прискорювачів через CUDA, що дає змогу суттєво зменшити час обробки моделей глибинного навчання.

З погляду програмного забезпечення система є кросплатформенною та підтримує сучасні операційні системи сімейств Windows, Linux і macOS. Клієнтська частина функціонує у браузері, який забезпечує підтримку актуальних веб-стандартів, тоді як фронтенд побудовано з використанням середовища Node.js, версії не нижче 22. Серверні компоненти та модуль машинного навчання можуть працювати як у вигляді контейнерів, так і у локальних середовищах, що потребує встановлених Python 3.12+ та .NET 8.0. Всі, інші, ключові версії бібліотек як-от PyTorch, Transformers, Scikit-learn тощо, інтегруються відповідно до обраної версії Python. Windows система інтегрується з WSL2, який забезпечує коректну роботу інструментів контейнеризації.

За зберігання даних відповідає реляційна СУБД PostgreSQL версії 15 або новішої, що гарантує сумісність зі схемою БД та ORM-рівнем. Усі модулі системи контейнеризовані та взаємодіють між собою за допомогою Docker-оркестрації, яка забезпечує автоматизоване збирання образів, запуск сервісів і підтримку стабільної роботи всієї інфраструктури.

Для розгортання та використання вебзастосунку необхідно відкрити термінал, клонувати репозиторій командою та перейти до каталогу проєкту:

git clone https://github.com/StefaniaVovk/fake-news-detector

cd <reposity-directory>

Запуск контейнерів виконується з кореневої директорії, де розташований файл docker-compose.yml. Достатньо виконати команду:

docker compose up --build -d

Під час першого запуску Docker автоматично збирає всі образи, встановлює залежності та запускає кожну службу у фоні. Після успішного розгортання фронтенд доступний за адресою:

- Frontend: http://localhost:3000 (для взаємодії з користувачем);
- Backend: http://localhost:5000 (API-шлюз);
- ML Service: http://localhost:8000 (API для ML-логіки).

Після успішного розгортання інфраструктури на стартовому екрані (Рисунок Г.1) відображається інтерфейс автентифікації, який дозволяє створювати облікові записи з різними рівнями доступу. У системі передбачено дві категорії користувачів: звичайні користувачі та дослідники. Звичайні користувачі мають доступ до основного функціоналу та інструментів аналітики, тоді як дослідники отримують розширені можливості – зміну конфігурацій моделей, управління навчальними даними та доступ до інструментів оптимізації та інтерпретації.

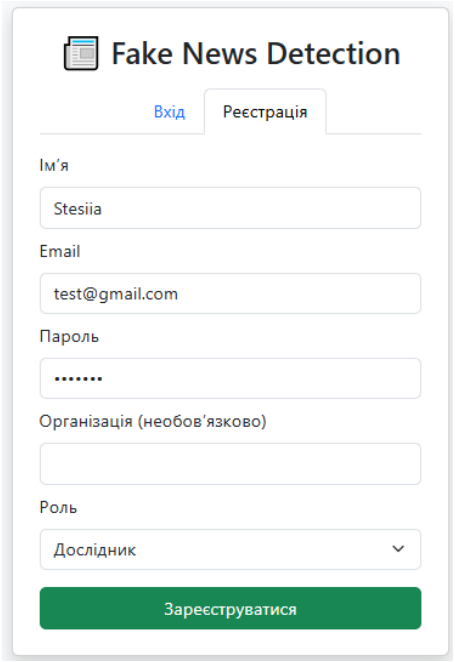


Рисунок Г.1 – Реєстрація користувача в системі

Після проходження авторизації користувач потрапляє на головну сторінку системи (Рисунок Г.2), де зібрано доступ до ключових модулів: завантаження текстів, аналізу, тренування моделей, роботи з конфігураціями та управління даними. Користувач із роллю «Дослідник» може взаємодіяти з усіма інструментами системи, що забезпечує повний цикл експериментування та покращення моделей.

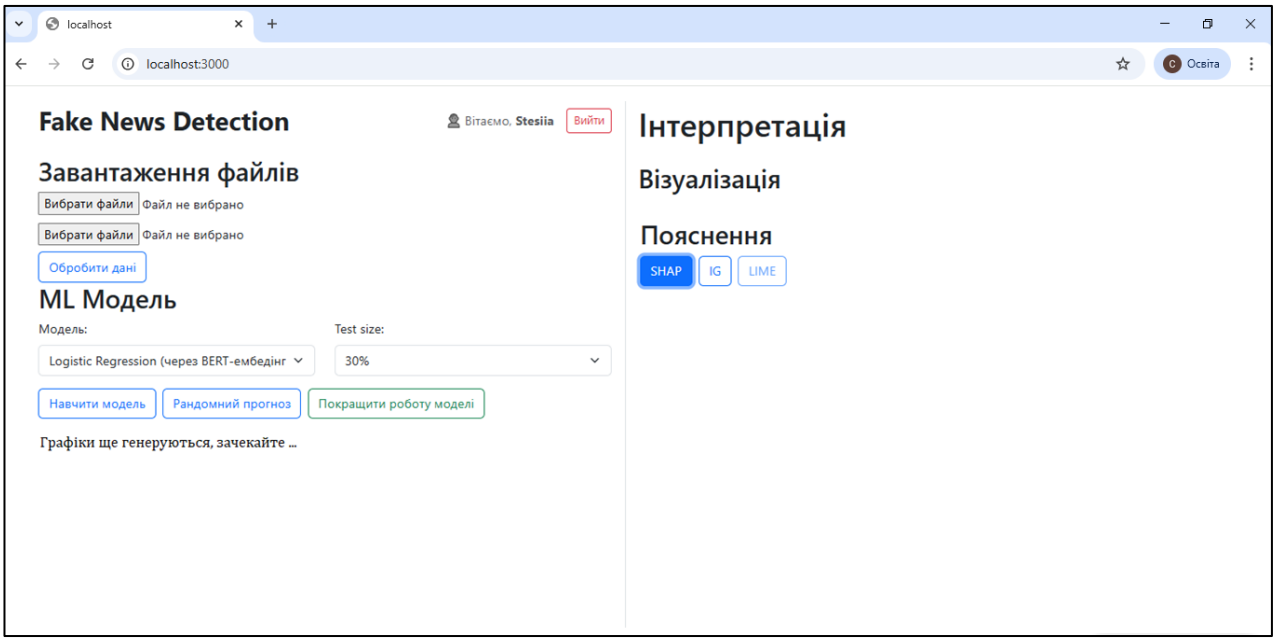


Рисунок Г.2 – Головна сторінка вебсистеми

Додаток Д

Презентаційний матеріал

Кваліфікаційна робота магістра

МЕТОД ІНТЕРПРЕТУВАННЯ РЕЗУЛЬТАТІВ ВИЯВЛЕННЯ ФЕЙКОВИХ НОВИН ЗА ВЕЛИКОЮ МОВНОЮ МОДЕЛЛЮ

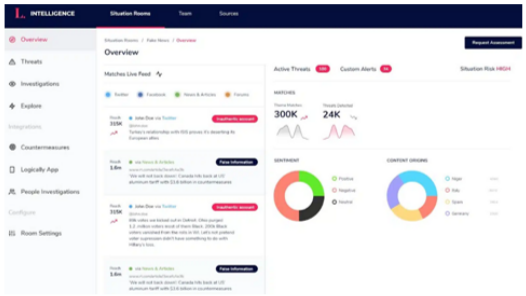
Виконала: студентка 2 курсу, групи КНм-24-1, Вовк Стефанія Віталіївна
Науковий керівник: доцент кафедри КН, Радюк Павло Михайлович

Актуальність

Сучасний інформаційний простір перебуває в стані безперервної «гонки озброєнь» між творцями дезінформації та системами її виявлення. Поява потужних великих мовних моделей кардинально змінила цей ландшафт: вони здатні генерувати надзвичайно переконливий і важко-ідентифікований фейковий контент. Водночас ці ж моделі використовуються для створення складних детекторів. Однак, попри високі показники точності, переважна більшість таких систем функціонує як «чорна скринька». Без розуміння логіки роботи моделі неможливо сформувати ефективні стратегії протидії дезінформації.

Ключ до розв'язання цієї проблеми лежить у дослідженні внутрішніх представлень даних моделі – текстових ембедінгів. Саме в цих багатовимірних векторах закодовані семантичні та стилістичні знання, на які спирається модель. Тому розробка методу, що дає змогу експерту інтерактивно досліджувати цей простір, є актуальною задачею для створення нового покоління інтерпретованих систем.

Наявні програмні засоби

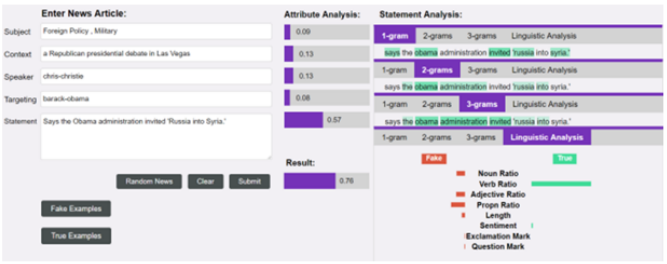


Logical Intelligence

- Комерційний проект
- Великий аналітичний асортимент
- Закрита система

XFake

- Для наукових досліджень
- Пояснення: лінгвістика та n-грами
- Відкрита система



Мета роботи

Метою роботи є підвищення рівня інтерпретованості систем виявлення фейкових новин через проєктування методу, який забезпечує інтерактивний аналіз, валідування та ітеративне вдосконалення простору текстових ембедінгів із залученням експерта.

Задачі роботи

Провести аналіз методів забезпечення прозорості LLM, підходів до візуалізації текстових ембедінгів та наявних програмних рішень для виявлення дезінформації в текстових даних.

Спроєктувати метод інтерпретування результатів виявлення фейкових новин, що полягає в генеруванні високоінформативних текстових ембедінгів для новинних статей із використанням архітектур великих мовних моделей.

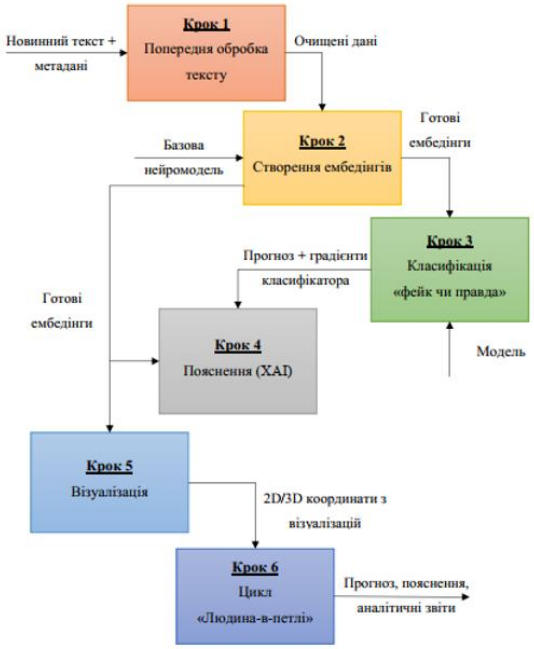
Розробити підсистему зниження розмірності та візуалізації багатовимірних ембедінгів у двовимірному просторі за спроєктованим методом.

Розробити архітектуру інтерактивної системи та виконати її програмну реалізацію, що надає інструменти для дослідження проєкцій, пояснення рішень та реалізує підхід “людина-у-петлі” для виявлення фейкових новин.

Провести експериментальне дослідження спроєктованого методу та його програмної реалізації за еталонними наборами даних та оцінити результативність запропонованих підходів.

Схема методу

Спроєктований метод інтерпретування результатів виявлення фейкових новин за великою мовною моделлю структуровано у шість послідовних кроків – від надходження сирих текстових даних до формування інтерпретованого прогнозу.



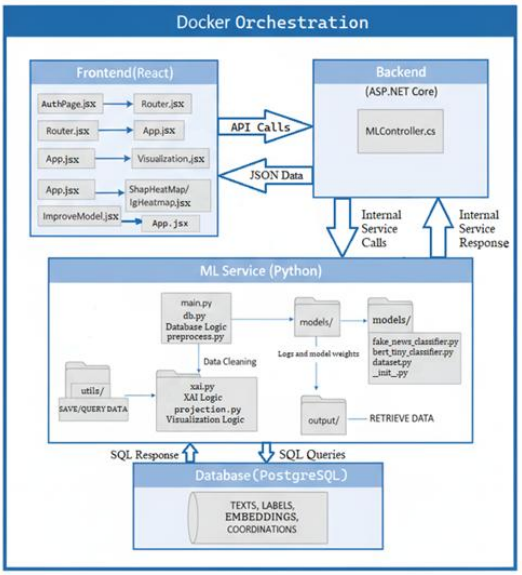
Архітектура системи

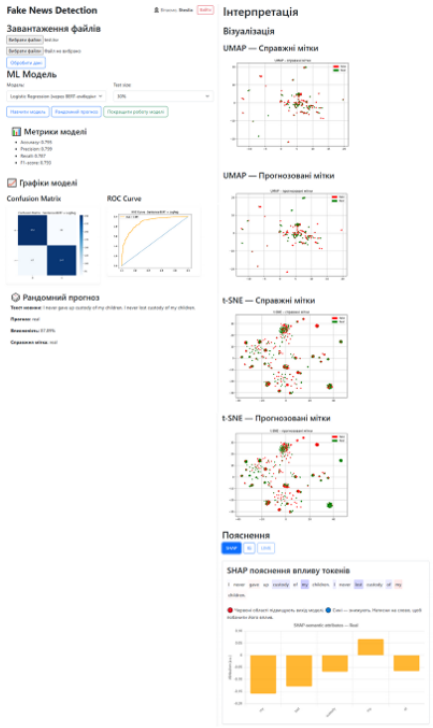
Структура інформаційної системи складається з трьох взаємопов'язаних сервісів та бази даних, об'єднаних у єдине середовище за допомогою Docker-оркестрації.

Перша підсистема — фронтенд, реалізована на основі шаблону React, відповідає за інтерфейс користувача, відображення результатів обчислень, взаємодію з формами введення та передачу запитів до бекенду.

Друга підсистема — бекенд, розроблена з використанням ASP.NET Core, забезпечує маршрутизацію запитів, обробку бізнес-логіки та координацію взаємодії між інтерфейсом користувача та обчислювальним модулем.

Третя підсистема — ML-сервіс виконує основні обчислення: навчання моделей, класифікацію текстів, формування метрик та генерацію пояснень, включно з глобальними та локальними інтерпретаціями.





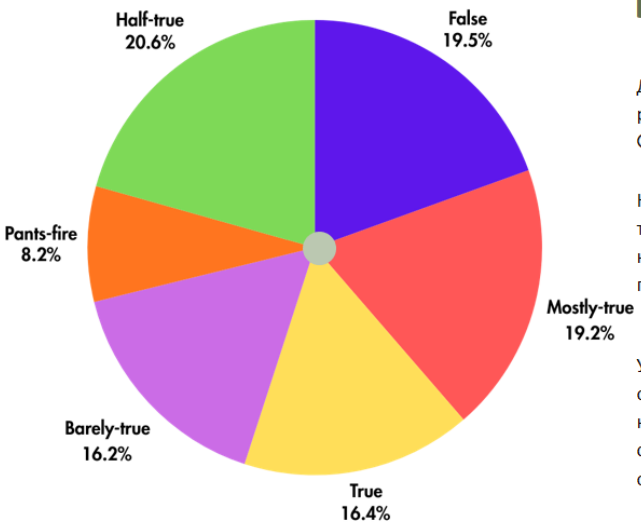
Скріншот роботи системи

Ліва панель системи

Розподіл роботи вебзастосунку організовано на дві основні частини. Ліва панель відповідає за навчання моделей на вибраному корпусі даних та подання результатів їхньої роботи, включно з метриками, матрицею помилок і ROC-кривою. У цьому ж розділі можна переглянути випадково вибрану класифіковану новину та сформувати для неї пояснення. Крім того, панель надає можливість користувачу покращувати модель, змінюючи параметри навчання або обраний набір даних.

Права панель системи

Права панель системи зосереджена на інтерпретації роботи моделей: вона відображає візуалізації векторного простору ознак, згенеровані за допомогою t-SNE та UMAP, а також подає пояснення для випадково вибраної новини.



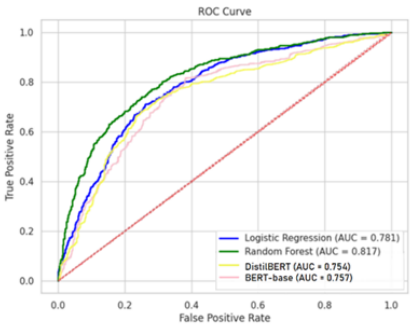
Корпус даних

Для тестування роботи реалізованого методу було обрано два репрезентативні корпуси: LIAR та FakeNewsNet із підвибіркою GossipCop.

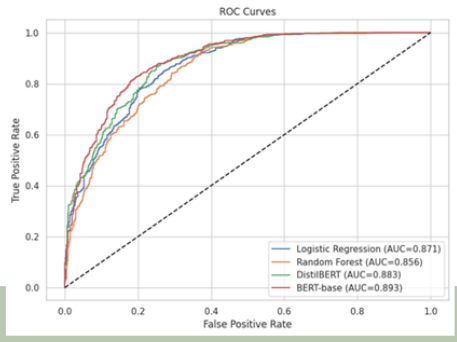
Корпус LIAR містить приблизно 12,8 тисячі коротких політичних тверджень із середньою довжиною близько 20 слів; хоча вихідний набір поділено на шість категорій, у межах цього дослідження він був приведений до бінарної класифікації.

У випадку FakeNewsNet було обрано саме підвибірку GossipCop, оскільки вона єдина містить достатній обсяг повнотекстових новин, необхідних для коректного аналізу. Хоча набір має два класи, у ньому спостерігається значний дисбаланс на користь класу True, тому перед обчисленнями дані було приведено до збалансованого вигляду.

Дослідження ефективності методу

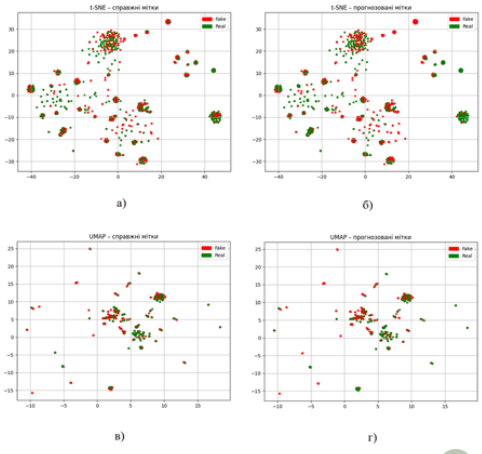


Для корпусу LIAR найвищий AUC показала модель Random Forest (0.817), що свідчить про її перевагу над іншими підходами. Логістична регресія отримала AUC = 0.781, тоді як трансформери продемонстрували нижчі значення (DistilBERT – 0.754, BERT-base – 0.757), оскільки без спеціальної архітектури вони менш ефективні для роботи з короткими текстами та табличними метаданими.

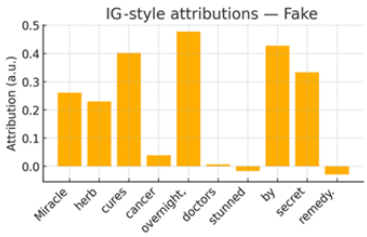


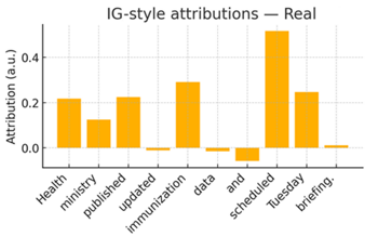
Для корпусу GossipCop усі моделі продемонстрували вищу продуктивність, однак BERT-base лідирує з AUC 0.893, слідом йдуть DistilBERT 0.883, логістична регресія 0.871 та Random Forest 0.856.

Дослідження ефективності методу



Для глибшого аналізу роботи класифікаційної моделі, окрім оцінки через метрики, матриці помилок та ROC-криві, користувачу надаються візуалізації простору ознак та пояснення методами XAI.





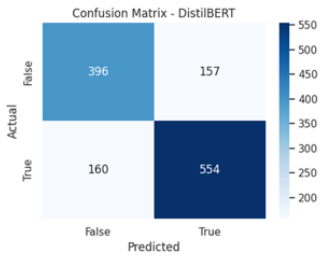
Дослідження ефективності методу

Дослідження ефективності роботи методу здійснювалось шляхом оцінки процесу навчання моделей. Для цього було проведено 6 циклів навчання для кожної моделі зі зміною початкових параметрів. Таким чином було обрано опитимальні початкові параметри для кожної моделі.

Epochs	Batch	Learning Rate	Точність
10	16	2e-5	0.71
10	32	1e-3	0.745
5	16	1e-3	0.738
5	32	2e-5	0.7

Визначення опитимальні значення початкових параметрів для моделі DistilBERT на бінаризованому корпусі даних LIAR. Найбільшої точності у 74,5% досягнуто на другому циклі навчання.

Матриця помилок DistilBERT моделі при класифікації новин на корпусі LIAR. Як видно з матриці, модель краще розпізнає позитивний клас, маючи більше правильних позитивних передбачень.



Висновки

Результатом роботи є розроблений метод інтерпретування результатів виявлення фейкових новин на базі великої мовної моделі, який забезпечує обробку та класифікацію новинних текстів, а також отримання інтерпретацій «хід думки моделі» через аналіз метрик, простору текстових ембедінгів та пояснення, сформовані за допомогою XAI-методів.

Для подальшого підвищення ефективності роботи методу можна замінити використану модель-трансформер DistilBERT на більш потужну архітектуру з більшою кількістю параметрів або на модель, попередньо натреновану саме на новинних корпусах. Крім того, потенційно кращих результатів можна досягти, застосувавши метод на інших наборах даних, зокрема CONSTRAINT-2021 (EN).

Проведені дослідження показали, що застосування розробленого методу дає змогу підвищити точність класифікації всередньому на 2–4%.

Mon Dec 08 19:51:43 EET 2025, Петровський Сергій Степанович, Хмельницький національний університет, ХНУ

Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 1.0%

Dictionaries check: en_US, ru_RU, ua_UA. **Errors in the documents: 14%**

	Document		Sum coincidence on the DB	
ID: 252060 Title: КВАЛІФІКАЦІЙНА РОБОТА на тему Метод інтерпретування результатів виявлення фейкових новин за великою мовною моделлю Added in a DB: 2025-12-08 Authors: Стефанія ВОВК Heads: Павло РАДЮК Consultants: Opponents:	Symbols	Lexemes	Symbols	Lexemes
	121042	1816	3210 (3%)	59 (3%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Стефанія ВОВК

Співавтор:

Назва: КВАЛІФІКАЦІЙНА РОБОТА на тему Метод інтерпретування результатів виявлення фейкових новин за великою мовною моделлю

Науковий керівник: Павло РАДЮК, док. філ., доцент кафедри

Підрозділ: Кафедра комп'ютерних наук

Коефіцієнт подібності 1:1.8%

Коефіцієнт подібності 2:0.3%

Мікропробіли: 0

Заміна букв: 9

Інтервали: 0

Білі знаки: 1

Дата створення звіту: 2025-12-10 15:03:49.0

Після аналізу Звіту подібності констатую наступне:

☑ **Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.**

☐ **Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.**

☐ **Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.**

Обгрунтування:

Дата 10.12.25 експерт Федровський С.Р.

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: *Метод інтерпретування результатів виявлення фейкових новин за великою мовною моделлю*

Автор: *студентка групи КНм-24-1 Вовк Стефанія Віталіївна*

Спеціальність: *122 Комп'ютерні науки*

Освітня програма: *освітньо-професійна*

Науковий керівник: *док. філ., доц. каф. КН Радюк П.М.*

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	***відповідає***
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, що виявлені в роботі Стефанії ВОВК, не є плагіатом, оскільки: запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; до запозичень входять фрагменти, що не мають авторства і містять поширені конструкції; поміж запозичень є загальновідомі терміни та скорочення.

Обсяг запозичень, що визначений системами виявлення збігів/ ідентичності/схожості, складає:

– за системою Anti-Plagiarism: 1.0%;

– за системою StrikePlagiarism: КП 1 – 1.8%, КП 2 – 0.3%.

Отже, знайдені запозичення є допустимими та відносяться до описаних вище і адресуються до першоджерел, що, з урахуванням наведених обґрунтувань, свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ *Павло РАДЮК*

Гарант ОП _____ *Руслан БАГРІЙ*

Завідувач кафедри КН _____ *Олександр БАРМАК*

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ

Кафедра комп'ютерних наук

ВІДГУК НАУКОВОГО КЕРІВНИКА

на кваліфікаційну роботу магістра

студентки *гр. КНм-24-1 Вовк Стефанії Віталіївни*
за темою *Метод інтерпретування результатів виявлення фейкових новин за великою мовною моделлю*

1. Актуальність теми

У сучасних умовах інформаційних атак та стрімкого поширення дезінформації автоматизовані системи виявлення фейків стають необхідним інструментом медіаграмотності та інформаційної безпеки. Сучасні великі мовні моделі демонструють високу ефективність у задачах класифікації текстів, однак часто функціонують як "чорні скриньки", що ускладнює розуміння логіки прийняття ними рішень. Брак прозорості знижує довіру користувачів до результатів роботи таких систем. Тому актуальним завданням є створення методів, які не лише точно класифікують новини на фейковість, а й надають зрозумілі пояснення (інтерпретації) власних рішень, а також дають змогу експертам впливати на процес навчання моделі для підвищення її точності.

2. Відповідність роботи предметній області 122 Комп'ютерні науки та загальним вимогам наукових робіт

Відповідно до стандарту вищої освіти України спеціальності 122 Комп'ютерні науки, описом предметної галузі, об'єктом та предметом вивчення є математичні, інформаційні та імітаційні моделі реальних явищ, об'єктів, систем і процесів та методи й технології отримання, зберігання, обробки, передачі та використання інформації. Метою поданої роботи є підвищення рівня інтерпретованості та точності систем виявлення фейкових новин через проєктування методу, який забезпечує інтерактивний аналіз, валідування та ітеративне вдосконалення простору текстових ембедінгів із залученням експерта. Мету роботи досягнуто внаслідок використання методів, способів та алгоритмів розв'язання теоретичних і прикладних задач, що виникають у процесі проєктування вказаного методу. Отже, кваліфікаційна робота магістра повністю відповідає спеціальності 122 Комп'ютерні науки.

3. Професійні та особистісні якості магістранта

Під час виконання кваліфікаційної роботи магістра студентка Стефанія Вовк проявила себе кваліфікованою фахівчинею та дисциплінованою студенткою, вчасно виконуючи поставлені завдання. Як у процесі розроблення прикладного програмного забезпечення, так і під час написання пояснювальної записки студентка успішно засвоїла компетентності та результати навчання. Стефанія Вовк опанувала професійні навички та компетентності, що повністю відповідають виконанню освітньо-професійної програми другого рівня вищої освіти «Магістр» за спеціальністю 122 Комп'ютерні науки.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Результати роботи та їхня обґрунтована практична значущість одержані та обумовлені студенткою особисто, як наслідок виконання нею усіх поставлених завдань.

5. Наукова новизна та оригінальність запропонованих підходів

Отримані результати відзначаються оригінальністю застосованого підходу до інтерпретування модельних рішень, прийнятих великою мовною моделлю для класифікації фейкових новин. Запропонований Стефанією Вовк метод доповнює наявні рішення до виявлення фейкових новин з використанням моделей глибокого навчання та забезпечує зрозуміле пояснення рішень моделі.

6. Ступінь оволодіння методами дослідження

У процесі проєктування та програмної реалізації методу інтерпретування результатів виявлення фейкових новин за великою мовною моделлю студентка Стефанія Вовк продемонструвала відмінний рівень компетентностей, умінь і навичок використання інструментарію інформаційних технологій.

7. Повнота та якість розкриття теми роботи

Тема роботи повністю обґрунтована й розкрита; актуальність предметної галузі та відомі дослідження щодо обраної тематики проаналізовані повно та вичерпно. Усі завдання, що були поставлені перед студенткою, виконані повно та успішно. Розроблений вебзастосунок для експериментального тестування над спроєктованим методом відповідає технічним вимогам спеціальності 122 Комп'ютерні науки.

8. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу

Матеріал кваліфікаційної роботи магістра Стефанії Вовк подано логічно, послідовно, аргументовано та є таким, що відповідає поставленій меті. Мова та стиль викладення роботи відповідають стандартам, що забезпечує доступність сприймання матеріалу й відповідає вимогам до сучасних кваліфікаційних робіт.

9. Можливість практичного застосування кваліфікаційної роботи, окремих її частин

Створений в роботі метод та його програмна реалізація можуть використовуватися для автоматизованого фактчекінгу та моніторингу медіапростору, надаючи експертам інструменти для інтерпретації рішень штучного інтелекту та інтерактивного донавчання моделей для підвищення точності виявлення дезінформації.

10. Висновок про можливість допуску кваліфікаційної роботи до захисту, на яку оцінку заслуговує робота

З огляду на високий рівень виконання та забезпечення всіх необхідних вимог, вважаю, що кваліфікаційна робота магістра Стефанії Вовк може бути допущена до захисту. Рекомендована оцінка – «відмінно».

Керівник _____ *док. філ., доц. каф. КН Павло РАДЮК*

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ

Кафедра комп'ютерних наук

ВІДГУК ОПОНЕНТА

на кваліфікаційну роботу магістра

студентки *гр. КНм-24-1 Вовк Стефанії Віталіївни*

за темою: *Метод інтерпретування результатів виявлення фейкових новин за великою мовною моделлю*

1. Актуальність обраної теми

У наш час стрімкого зростання обсягів інформації та активізації дезінформаційних кампаній системи автоматизованого виявлення фейкових новин набувають особливої важливості. Використання сучасних моделей штучного інтелекту відкриває можливості для точного аналізу текстів. Разом з тим штучний інтелект породжує проблему браку прозорості, оскільки багато моделей залишаються так званими «чорними скриньками». Це знижує довіру користувачів і ускладнює оцінювання коректності результатів. Тому актуальним завданням є створення рішень, які поєднують високу точність класифікації з можливістю інтерпретації рішень моделі та забезпечують інструменти для глибшого аналізу і подальшого вдосконалення процесу виявлення фейків у текстових даних.

2. Відповідність роботи предметній області 122 Комп'ютерні науки та загальним вимовам до наукових робіт

Кваліфікаційна робота магістра відповідає предметній області спеціальності 122 Комп'ютерні науки, оскільки досліджує актуальну проблему створення методу інтерпретування результатів виявлення фейкових новин за великою мовною моделлю. Усі етапи виконання, від аналізу проблеми до експериментальної перевірки, відповідають стандартам підготовки магістерських робіт.

3. Повнота розкриття мети та завдань дослідження

У результаті виконання кваліфікаційної роботи магістра мету та завдання роботи розкрито повністю. Проведено глибокий аналіз предметної галузі, розглянуто та проаналізовано п'ятдесят два літературні джерела щодо обраної тематики. Чітко визначено структуру спроєктованого методу та реалізовано прикладне програмне забезпечення у вигляді вебзастосунку для проведення експериментального тестування.

4. Наявність наукової новизни

Наукова новизна кваліфікаційної роботи полягає в удосконаленні методу інтерпретування результатів виявлення фейкових новин унаслідок інтеграції великих мовних моделей із модулями пояснюваного штучного інтелекту та концепцією "людина-у-петлі". Запропонований метод відрізняється від наявних рішень поєднанням послідовного оброблення тексту (через трансформери), глибокого семантичного аналізу, візуалізації простору ознак та наданням можливості експерту інтерактивно впливати на параметри моделі та навчальну вибірку, що дало змогу підвищити точність класифікації на 2–4 % та забезпечити прозорість прийняття рішень.

5. Зміст кожного розділу роботи

У першому розділі кваліфікаційної роботи проведено аналіз проблем та рішень у сфері інтерпретації великих мовних моделей у задачах виявлення фейкових новин. Другий

розділ присвячений проєктуванню методу інтерпретування результатів виявлення фейкових новин за великою мовною моделлю. У третьому розділі наведено та описано програмну реалізацію поданого методу у вигляді вебзастосунку У четвертому розділі проведено експериментальні дослідження для оцінювання точності та рівня інтерпретованості методу. Виконані завдання та одержані результати роботи підсумовано у загальних висновках.

6. Ступінь розкриття теми роботи

У кваліфікаційній роботі магістра повністю розкрито заявлену тему через аналіз, проєктування, програмну реалізацію та експериментальне тестування методу інтерпретування результатів виявлення фейкових новин. Магістеркою подано детальний та вичерпний опис спроєктованого методу. Робота містить опис програмної реалізації методу у вигляді вебзастосунку. Авторкою проведено дослідження роботи вебзастосунку та здійснено експериментальне тестування за еталонними корпусами текстових даних.

7. Якість оформлення кваліфікаційної роботи

Робота відповідає вимогам науково-технічного стилю написання, має чітку структуру та включає всі необхідні розділи, як от, перелік скорочень, вступ, огляд літератури, методологія, результати експериментів, висновки та перелік посилань. Кожен розділ повністю виконує своє функціональне призначення і має чіткий зміст. Використані джерела належно процитовані й включені до списку використаної літератури, відповідно до наукових стандартів цитування. Використана авторкою структура та оформлення сприяють легкому сприйняттю та розумінню матеріалу.

8. Недоліки оформлення кваліфікаційної роботи

Робота також містить несуттєві недоліки. Подекуди оформлення графічних матеріалів відрізняється за розмірами та шрифтами, тому їхня уніфікація могла б покращити загальне сприйняття роботи.

9. Недоліки кваліфікаційної роботи

Також до недоліків можна віднести обмеження експериментальної бази лише англомовними наборами даних (LIAR та GossipCop) та брак оцінювання запропонованого методу при обробці україномовного контенту. Це потребує подальшої адаптації системи для використання у вітчизняному медіапросторі.

10. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота

*З огляду на високий рівень виконання та забезпечення всіх необхідних вимог, вважаю, що подана кваліфікаційна робота магістра Стефанії Вовк є оригінальним та завершеним науковим дослідженням. Тому кваліфікаційна робота може бути допущена до захисту. Рекомендована оцінка – «**відмінно**».*

Опонент (прізвище, ім'я, по батькові, посада, місце роботи)

д.т.н., професор Тетяна ГОВОРУЩЕНКО

«11» 12 2024 р. _____

(підпис)