Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
Кафедра інформатики

Магістерська робота
освітній ступінь – магістр

на тему:

«ЗАСТОСУВАННЯ МЕТОДІВ DATA MINING ДЛЯ АНАЛІЗУ ПОВІДОМЛЕНЬ У СОЦІАЛЬНИХ МЕРЕЖАХ З МЕТОЮ ВИЯВЛЕННЯ ФЕЙКІВ ТА ДЕЗІНФОРМАЦІЇ»

Виконала: студентка 2-го року навчання,
Спеціальності
121 Інженерія програмного забезпечення
Живіцька Єлизавета Олексіївна

Керівник Олецький Олексій Віталійович, доцент кандидат наук

Магістерська робота захищена з оцінкою _____

Секретар ЕК _____
«____» _____ 20____ р.

Київ – 2026

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

Національний університет «Києво-Могилянська академія»

Факультет: Інформатики (ФІ)

Кафедра: Інформатики

Освітній ступінь: Магістр

Спеціальність: 121 Інженерія програмного забезпечення

(код і назва)

Освітня/освітньо-наукова програма: Інженерія програмного забезпечення

(назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри _____

_____

«___» _____ 20___ року

З А В Д А Н Н Я
ДЛЯ МАГІСТЕРСЬКОЇ РОБОТИ СТУДЕНТУ

Живіцька Єлизавета Олексіївна

(прізвище, ім'я, по батькові)

1. Тема роботи: Застосування методів Data Mining для аналізу повідомлень у соціальних мережах з метою виявлення фейків та дезінформації

керівник роботи Олецький Олексій Віталійович, доцент, кандидат наук

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «___» _____ 20___ року № ____

2. Строк подання студентом роботи 20 травня 2026 року

3. План роботи:

1) Здійснити огляд літератури та проаналізувати існуючі підходи до виявлення дезінформації.
2) Обґрунтувати вибір датасету для експериментів.
3) Спроектувати архітектуру програмного прототипу системи.
4) Реалізувати інженерію ознак для моделей машинного навчання.
5) Натренувати моделі різних парадигм та інтегрувати LLM-підхід.

6) Провести експериментальне дослідження у внутрішньодоменному та міждоменному сценаріях.
7) Виконати дослідження вкладу окремих груп ознак.
8) Здійснити порівняльний аналіз парадигм та сформулювати практичні рекомендації.
9) Оформити магістерську роботу та підготуватися до захисту.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	ПЕРЕЛІК РОБІТ	Термін виконання	Дата ознайомлення наукового керівника	Підпис наукового керівника	Примітки
1.	Вибір теми, затвердження її на засіданні кафедри та закріплення наукового керівника. Узгодження календарного графіка підготовки кваліфікаційної роботи. Ознайомлення студента з критеріями оцінювання кваліфікаційної роботи (п. 8.5).	жовтень			
2.	Вивчення джерел літератури, матеріалів архівів, періодичних видань, збір та узагальнення фактів, даних	жовтень – листопад			
3.	Складання плану кваліф. роботи та узгодження з науковим керівником	листопад			
4.	Написання розділів роботи	листопад – березень			

№ з/п	ПЕРЕЛІК РОБІТ	Термін виконання	Дата ознайомлення наукового керівника	Підпис наукового керівника	Примітки
5.	Проміжний контроль виконання роботи	31 січня			
6.	Написання кваліфікаційної роботи в цілому, ознайомлення з її першим варіантом наукового керівника	січень – березень			
	Розділ 1 *(постановка проблеми, теоретичні основи, огляд літературних джерел)*				
	Розділ 2 *(аналітично-дослідницька частина) (експериментальна частина для природничих і біологічних наук)*				
	Розділ 3 *(проєктно-рекомендаційна частина) (аналіз результатів експерименту для природничих і біологічних наук)*				

№ з/п	ПЕРЕЛІК РОБІТ	Термін виконання	Дата ознайомлення наукового керівника	Підпис наукового керівника	Примітки
7.	Повне завершення написання кваліфікаційної роботи, оформлення її згідно з вимогами й подання на відгук науковому керівнику	квітень – початок травня			
8.	Подання кваліфікаційної роботи для перевірки письмових робіт студентів НаУКМА на відповідність вимогам академічної доброчесності	середина травня			
9.	Подання на зовнішню рецензію	до 6 квітня			
10.	Підготовка до захисту кваліфікаційної роботи на засіданні кафедри: написання доповіді та виготовлення ілюстративного матеріалу	до 27 квітня			
11.	Попередній захист кваліфікаційної роботи на засіданні кафедри	27 квітня			
12.	Подання кваліфікаційної	до 20 травня			

№ з/п	ПЕРЕЛІК РОБІТ	Термін виконання	Дата ознайомлення наукового керівника	Підпис наукового керівника	Примітки
	роботи на кафедру з усіма супровідними документами				
13.	Публічний захист кваліфікаційної роботи перед екзаменаційною комісією	4-5 червня			

Графік узгоджено «___» _____ 2026 р.

Науковий керівник Олецький Олексій Віталійович (ПІБ)

Виконавець кваліфікаційної роботи Живіцька Єлизавета Олексіївна (ПІБ)

ЗМІСТ

АНОТАЦІЯ

Магістерська робота присвячена дослідженню ефективності методів машинного навчання для виявлення дезінформації у соціальних мережах – однієї з найактуальніших проблем сучасного цифрового середовища, де неправдиві новини поширюються швидше та ширше за достовірні, ставлячи під загрозу інформаційну безпеку суспільства. Увага зосереджена на порівнянні чотирьох парадигм машинного навчання – статистичної, нейромережевої, графової та підходу на основі великих мовних моделей – у внутрішньодоменному та міждоменному сценаріях класифікації. У роботі проаналізовано обмеження існуючих підходів до автоматизованого виявлення дезінформаційного контенту, а також розроблено програмний прототип, що інтегрує усі чотири парадигми у єдину дослідницьку платформу з підтримкою ансамблевих стратегій та зовнішніх фактчекер-сервісів. Основою прототипу є гнучка архітектура з розподілом між фронтендом, бекендом та ML-сервером, що дозволяє проводити порівняльне оцінювання моделей у межах єдиного експериментального середовища. Експериментальне дослідження виявило радикальну зміну ієрархії ефективності моделей при перенесенні між тематичними доменами: парадигми з найвищою внутрішньодоменною якістю демонструють найбільшу деградацію, тоді як соціальні ознаки та LLM-підхід забезпечують стійкість до зміни домену. Практичне значення роботи полягає у можливості застосування отриманих результатів та сформульованих рекомендацій у системах медіа-моніторингу, інформаційно-аналітичних платформах та інструментах підтримки журналістської роботи, де якість виявлення дезінформації безпосередньо впливає на достовірність інформаційного простору.

Ключові слова: виявлення дезінформації, фейкові новини, машинне навчання, графові нейронні мережі, великі мовні моделі, FakeNewsNet, порівняльний аналіз моделей.

ВСТУП

Актуальність теми дослідження. Феномен дезінформації у цифровому середовищі за останні роки перетворився з технічної проблеми на серйозний виклик для демократичних суспільств. Масштабне емпіричне дослідження поширення інформації у Twitter [3] переконливо продемонструвало, що неправдиві новини поширюються значно швидше та ширше за правдиві: серед найбільш вірусних публікацій фейкові охоплювали до 100 тисяч користувачів, тоді як правдиві рідко долали поріг тисячі. Такий масштаб поширення унеможливлює ручну верифікацію та вимагає автоматизованих систем виявлення дезінформації.

Сучасна дослідницька спільнота напрацювала кілька технологічних парадигм для розв'язання цієї задачі – від класичних статистичних моделей (наївний баєсівський класифікатор, SVM) через трансформерні архітектури (BERT, DistilBERT) до графових нейронних мереж (GraphSAGE, GIN) та підходів на основі великих мовних моделей (GPT, Claude). Кожна з цих парадигм демонструє свої сильні та слабкі сторони, проте у дослідницькій літературі бракує систематичного порівняння їх ефективності у межах єдиного експериментального середовища – з однаковим датасетом, єдиним розбиттям даних та узгодженим набором метрик. Більшість досліджень зосереджуються на одному підході або порівнюють обмежений набір методів у різних умовах, що ускладнює перенесення висновків на практичні задачі. Саме цей пробіл і обумовлює актуальність даної магістерської роботи, метою якої є проведення комплексного порівняльного аналізу чотирьох парадигм машинного навчання у задачі виявлення дезінформації.

Мета роботи полягає у проведенні порівняльного експериментального дослідження ефективності чотирьох парадигм машинного навчання – статистичної, нейромережевої, графової та LLM-парадигми – у задачі виявлення дезінформації у внутрішньодоменному та міждоменному сценаріях, а також у розробці інтегрованого програмного прототипу системи, що об'єднує ці парадигми у єдиному дослідницькому середовищі.

Для досягнення поставленої мети сформульовано такі завдання: здійснити огляд літератури з виявлення дезінформації, проаналізувати існуючі підходи та методи, виявити їхні переваги та обмеження; провести аналіз доступних датасетів для задачі виявлення дезінформації; спроектувати архітектуру програмного прототипу системи виявлення дезінформації; провести експериментальне дослідження ефективності моделей; здійснити порівняльний аналіз парадигм за шістьма метриками (Accuracy, Precision(FAKE), Recall(FAKE), F1(FAKE), F1-macro, ROC-AUC) та сформулювати практичні рекомендації щодо вибору моделі залежно від сценарію застосування.

Об'єктом дослідження є процес автоматизованого виявлення дезінформаційного контенту у соціальних мережах.

Предметом дослідження є порівняльна ефективність методів та моделей машинного навчання різних парадигм у задачі бінарної класифікації новинних публікацій з урахуванням текстового контенту та соціального контексту їхнього поширення.

Методи дослідження. У роботі використано методи статистичного аналізу тексту (TF-IDF векторизація, наївний баєсівський класифікатор), глибинного навчання на основі трансформерної архітектури (тонке налаштування DistilBERT), графових нейронних мереж (GraphSAGE, GIN на графах поширення Twitter), промптингу великих мовних моделей (zero-shot та few-shot класифікація через моделі сімейства Claude), а також методи статистичної оцінки якості класифікації (стандартні метрики Accuracy, Precision, Recall, F1, F1-macro, ROC-AUC).

Наукова новизна отриманих результатів полягає у такому: (1) проведено систематичне порівняння чотирьох парадигм машинного навчання у задачі виявлення дезінформації на єдиному датасеті з однаковим розбиттям та метриками; (2) виявлено та емпірично підтверджено явище інверсії ієрархії ефективності парадигм при перенесенні між тематичними доменами – моделі з найвищою внутрішньодоменною якістю (графові мережі) демонструють найбільшу деградацію при міждоменному перенесенні; (3) встановлено емпіричні докази того, що соціальні ознаки (метадані авторів та структура каскаду поширення) кодують

універсальний сигнал дезінформації, інваріантний до тематичного домену; (4) продемонстровано перевагу LLM-підходу через Claude Haiku над усіма натренованими моделями у міждоменному сценарії за відсутності тренувального етапу.

Практичне значення отриманих результатів полягає у розробці модульного програмного прототипу системи виявлення дезінформації, що підтримує тренування та порівняння моделей чотирьох парадигм, ансамблеві стратегії об'єднання прогнозів та інтеграцію з зовнішніми фактчекер-сервісами. Сформульовані у роботі практичні рекомендації щодо вибору моделі залежно від типу домену, доступних обчислювальних ресурсів та вимог до інтерпретованості можуть бути використані розробниками систем медіа-моніторингу, інформаційно-аналітичних платформ та інструментів підтримки журналістської роботи.

Структура роботи. Магістерська робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. У першому розділі представлено огляд методів та підходів до виявлення дезінформації, що охоплює аналіз самого феномену, огляд доступних датасетів, класичні методи машинного навчання, нейромережеві та графові архітектури, великі мовні моделі та техніки інженерії ознак. У другому розділі описано проєктування та реалізацію програмного прототипу, його архітектуру, технологічний стек, процеси обробки даних та реалізацію моделей. Третій розділ містить результати експериментального дослідження ефективності моделей: внутрішньодоменне оцінювання, дослідження вкладу ознак для Naive Bayes, міждоменні експерименти, оцінювання LLM-підходу та порівняльний аналіз парадигм.

РОЗДІЛ 1. ОГЛЯД МЕТОДІВ ТА ПІДХОДІВ ДО ВИЯВЛЕННЯ ДЕЗІНФОРМАЦІЇ У СОЦІАЛЬНИХ МЕРЕЖАХ

1.1 Дезінформація як суспільне явище та об'єкт автоматизованого виявлення

1.1.1 Визначення та типологія дезінформації

У сучасному науковому дискурсі термін «фейкові новини» (fake news) дедалі частіше піддається критиці через концептуальну розмитість та політизацію. Замість нього у академічному середовищі поширилася модель «інформаційного безладу» (information disorder), запропонована Клер Уордл та Хосейном Деракшаном у звіті для Ради Європи 2017 року [1]. У межах цієї моделі автори виокремлюють три категорії хибної або шкідливої інформації, що розрізняються за двома критеріями – правдивістю контенту та намірами розповсюджувача.

Мізінформація (Misinformation) – неправдива або неточна інформація, що поширюється без усвідомленого наміру завдати шкоди. До цієї категорії належать ненавмисні помилки на кшталт неточних дат, статистичних показників, перекладів або сатиричного контенту, сприйнятого аудиторією як правдивий.

Дезінформація (Disinformation) – неправдива інформація, що цілеспрямовано створюється та розповсюджується з метою введення аудиторії в оману, маніпуляції громадською думкою або заподіяння шкоди особі, групі чи організації.

Малінформація (Malinformation) – інформація, що базується на реальних фактах, проте використовується з деструктивними намірами. Класичним прикладом є витоки приватного листування для дискредитації публічних осіб або поширення приватних знімків без згоди їх власників.

Ключова відмінність дезінформації від мізінформації полягає у наявності умислу – свідомого наміру ввести в оману. Цей нюанс має фундаментальне значення для побудови систем автоматизованого виявлення: якщо мізінформацію

часто можна виявити через формальні неточності тексту, то дезінформація типово ретельно структурована для маскування під правдиву інформацію та потребує більш тонких підходів аналізу – стилістичних, контекстних, перевірки фактологічної бази через зовнішні джерела.

З погляду формалізації для задач машинного навчання важливою є робота Шу та співавторів [2], у якій сформульовано тривимірне представлення феномену дезінформації через взаємодію трьох компонентів: контенту повідомлення, його соціального контексту (реакції користувачів, мережі поширення, профілі поширювачів) та часової динаміки розповсюдження. Ця модель безпосередньо лягла в основу датасету FakeNewsNet, що використовується у даній роботі та детально розглядається у підрозділі 1.2.

1.1.2 Особливості поширення дезінформації у соціальних мережах

Розуміння механізмів поширення дезінформації у соціальних мережах є необхідною передумовою для побудови ефективних систем її автоматизованого виявлення. Найбільш масштабне емпіричне дослідження цього явища проведене Восоугі, Роєм та Аралем у статті «Поширення правдивих та неправдивих новин онлайн», опублікованій у журналі Science 2018 року [3]. Автори проаналізували приблизно 126 000 верифікованих історій, що поширювалися у Twitter протягом 2006–2017 років близько 3 мільйонами користувачів та згенерували понад 4,5 мільйона твітів. Класифікація історій як правдивих чи неправдивих здійснювалася на основі узгодженості шести незалежних фактчекер-організацій, що демонстрували від 95 до 98 відсотків узгодженості у класифікаціях.

Дослідження зафіксувало парадоксальну закономірність: неправдиві новини поширювалися значно далі, швидше, глибше та ширше за правдиві у всіх категоріях інформації, а ефект був особливо вираженим для політичних новин у порівнянні з новинами про тероризм, природні катастрофи, науку, міські легенди чи фінанси [3]. Зокрема, у топ-1% каскадів неправдивих новин охоплення сягало від тисячі до ста тисяч користувачів, тоді як правдиві новини рідко поширювалися

більш ніж на тисячу осіб. Як ймовірні причини автори виокремлюють вищу новизну неправдивого контенту порівняно з правдивим та сильніші емоційні реакції, які він викликає у одержувачів – переважно подив, страх та огиду, на противагу довірі та радості, властивим реакціям на правдиві повідомлення.

Окрім феномену вірусності, у літературі виокремлюється кілька системних факторів, що сприяють поширенню дезінформації у цифровому середовищі: архітектура соціальних платформ забезпечує миттєву публікацію без попередньої верифікації; алгоритмічна фільтрація формує «ехо-камери» з гомогенним контентом, що підтверджує існуючі переконання користувача; скоординовані мережі ботів [11] штучно маніпулюють показниками популярності, надаючи дезінформаційному контенту видимості природного зростання; економіка уваги мотивує платформи до поширення сенсаційного матеріалу; когнітивні упередження роблять користувачів вразливими до маніпулятивного контенту [1].

1.1.3 Постановка задачі автоматизованого виявлення

З формальної точки зору задача автоматизованого виявлення дезінформації у класичній постановці формулюється як бінарна класифікація текстових повідомлень за двома взаємовиключними мітками: FAKE (дезінформація) та REAL (правдива інформація). Вхідними даними для класифікатора слугує текст публікації, який часто доповнюється супутніми ознаками – метаданими автора, реакціями інших користувачів, часовими характеристиками поширення. На виході модель повертає прогнозовану мітку класу та значення впевненості у прогнозі (число від 0 до 1).

Бінарна постановка є усталеною у літературі та використовується у більшості академічних датасетів – LIAR [4], ISOT [5], FakeNewsNet [2, 6], CoAID [7]. Вона надає чітку та інтерпретовану відповідь для кінцевого користувача, а також спрощує побудову та оцінювання моделей через використання стандартних метрик і порівняння з еталонними значеннями попередніх досліджень. Альтернативні постановки – багатокласова класифікація з градаціями правдивості (наприклад,

шкала LIAR: «pants on fire – false – mostly false – half true – mostly true – true» [4]) або регресійні підходи з безперервною оцінкою достовірності – становлять окремі напрями досліджень, проте залишаються за межами цього огляду.

1.2 Огляд датасетів для дослідження дезінформації

Якість систем автоматизованого виявлення дезінформації безпосередньо залежить від якості тренувальних даних. Цей підрозділ містить огляд найбільш поширених у дослідницькій спільноті датасетів, аналіз їх характеристик з точки зору придатності до різних експериментальних завдань та обґрунтування вибору датасету FakeNewsNet як основи для практичної частини роботи.

1.2.1 Загальні вимоги до датасетів

При виборі датасету для дослідницької роботи з виявлення дезінформації слід враховувати критерії, що визначають його придатність для конкретних експериментальних завдань.

Перший критерій – обсяг датасету. Для тренування статистичних моделей, таких як наївний баєсівський класифікатор, достатньо 2-5 тисяч прикладів, тоді як для тонкого налаштування трансформерних моделей рекомендується 5-10 тисяч прикладів, хоча fine-tuning можливий і на менших обсягах (1-3 тисячі) за умови обережної регуляризації.

Другий критерій – врахування розподілу класів: датасет з істотно непропорційним розподілом FAKE та REAL прикладів вимагає відповідних методологічних рішень (стратифікованого розбиття вибірок, акценту на F1 для класу меншості замість accuracy, можливих технік балансування під час тренування). Реальні датасети дезінформації типово демонструють дисбаланс, що відображає природний розподіл контенту у медіапросторі.

Третій критерій – наявність соціального контексту. Дослідження поширення дезінформації у соціальних мережах вимагає не лише текстів самих публікацій, але

й метаданих про реакції користувачів, мережі поширення, профілі поширювачів. Більшість ранніх датасетів містили лише тексти статей з мітками, що обмежувало можливості для побудови моделей з урахуванням соціального контексту.

Четвертий критерій – авторитетність розмітки: мітки FAKE/REAL повинні базуватися на висновках професійних фактчекер-організацій, а не на автоматичних евристиках чи crowdsourcing-розмітці, що може містити значну частку шуму.

1.2.2 Огляд відомих датасетів

Серед публічних датасетів, що набули найбільшого поширення у дослідницькій спільноті, виокремлюються чотири: LIAR, FakeNewsNet, ISOT та CoAID. Розглянемо кожен з них.

LIAR [4] – датасет, опублікований у 2017 році, що складається з 12 836 коротких політичних тверджень, зібраних з фактчек-сервісу PolitiFact.com за десятирічний період (2007–2016). Особливістю датасету є шестикласова шкала розмітки: pants-on-fire (явна брехня), false (неправда), barely-true (майже неправда), half-true (напівправда), mostly-true (переважно правда), true (правда). Окрім тексту твердження, кожен запис містить багатий набір метаданих: автор, його партійна приналежність, посада, штат, тематика, контекст висловлювання, історія попередніх перевірок. Перевагою LIAR є висока якість розмітки, виконаної професійними редакторами PolitiFact. Недоліком для задач бінарної класифікації є необхідність агрегувати шість класів у два, що знижує точність експериментів. Крім того, датасет містить лише тексти тверджень без супровідного соціального контексту або повних статей.

ISOT [5] – датасет, опублікований у 2017 році, що містить 44 898 новинних статей, з яких 21 417 правдивих та 23 481 неправдивих. Правдиві статті зібрано з ресурсу Reuters.com, неправдиві – з вебсайтів, помічених як ненадійні фактчек-організацією PolitiFact, а також з Wikipedia. Більшість контенту стосується політичних та світових новин. Перевагою ISOT є значний обсяг та збалансованість класів. Однак важливим обмеженням є неоднорідність джерел правдивих та

неправдивих статей: оскільки REAL-клас походить виключно з Reuters, а FAKE – з різних джерел, моделі можуть навчатися розрізняти не дезінформацію як таку, а стилістичні відмінності між Reuters та іншими виданнями, що призводить до завищених показників ефективності, які не переносяться на нові домени.

FakeNewsNet [2, 6] – датасет, опублікований у 2018 році. Це найбільш комплексний з розглянутих датасетів з точки зору охоплення інформації – він містить не лише тексти новинних статей з мітками FAKE/REAL, але й супровідний соціальний контекст: твіти, що поширюють кожну статтю, профілі їх авторів, мережі підписок, часові характеристики поширення. Датасет складається з двох підмножин – GossipCop (статті про знаменитостей з відповідного фактчек-ресурсу) та PolitiFact (політичні новини). Розмітка ґрунтується на вердиктах професійних фактчекер-організацій, що забезпечує її надійність. Завдяки наявності соціального контексту FakeNewsNet дозволяє проводити дослідження з різних перспектив – як content-based класифікація, так і моделі на основі мережевого розповсюдження інформації.

CoAID [7] – спеціалізований датасет, опублікований у 2020 році для дослідження дезінформації, пов'язаної з пандемією COVID-19. Містить понад чотири тисячі новинних статей про здоров'я, понад дев'ятсот публікацій соціальних платформ та близько трьохсот тисяч пов'язаних взаємодій користувачів. Особливістю CoAID є вузька доменна спрямованість на медичну дезінформацію, що робить його придатним для досліджень у сфері health misinformation, проте обмежує застосовність для загальних задач виявлення дезінформації.

1.2.3 Обґрунтування вибору FakeNewsNet

Для практичної частини роботи обрано датасет FakeNewsNet з чотирьох ключових міркувань.

Розмір та структура. Десятки тисяч статей у підмножині GossipCop є оптимальним обсягом для тренування як статистичних моделей, так і трансформерних архітектур, а наявність двох тематично відмінних підмножин

(GossipCop – знаменитості, PolitiFact – політика) дозволяє проводити експерименти у двох сценаріях оцінювання моделей.

Наявність соціального контексту. На відміну від більшості датасетів виявлення дезінформації, FakeNewsNet містить не лише тексти статей з мітками FAKE/REAL, а й повні Twitter-каскади їх поширення – оригінальні твіти, ретвіти, відповіді, профілі авторів з метаданими (кількість підписників, дата створення облікового запису, статус верифікації). Це створює технічну основу для реалізації двох незалежних напрямів дослідження: класифікації на рівні текстового контенту та інженерії соціальних ознак на основі поведінкових патернів поширювачів – двох з трьох вимірів моделі Tri-Relationship [2].

Надійність розмітки. Мітки FAKE/REAL ґрунтуються на вердиктах фактчек-ресурсів GossipCop та PolitiFact – членів International Fact-Checking Network (IFCN) [25], що дотримуються встановлених принципів верифікації, забезпечуючи якість тренувальних сигналів.

Академічна релевантність. Датасет активно використовується у дослідницькій спільноті, що уможливлює порівняння отриманих результатів з опублікованими у літературі. Технічні характеристики FakeNewsNet та результати його використання в експериментах детально розглянуто у Розділі 3.

Перевагу FakeNewsNet над альтернативними публічними датасетами наочно демонструє порівняльна таблиця, представлена авторами у супровідній статті [6] (табл. 1.1). На відміну від інших датасетів, що покривають лише окремі категорії інформації – переважно лінгвістичний контент та частковий соціальний контекст – FakeNewsNet є єдиним, що містить усі три виміри: контент новин (лінгвістичний та візуальний), соціальний контекст (профілі користувачів, пости, відповіді, мережа поширення) та просторово-часові характеристики (геолокація та часові мітки). Така комплексність робить датасет придатним для широкого спектра задач – від класифікації на основі тексту до моделювання каскадів поширення через графові нейронні мережі, що використовуються у даній роботі.

Таблиця 1.1 *Порівняння FakeNewsNet з іншими публічними датасетами виявлення фейкових новин за категоріями доступних ознак. Адаптовано з [6].*

Датасет	Контент		Соціальний контекст				Просторово-часова інформація	
	Лінгвістика	Візуальний	Користувач	Пост	Відповіді	Мережа	Геолокація	Час
BuzzFeedNews	✓							
LIAR	✓							
BS Detector	✓							
CREDBANK	✓		✓	✓			✓	✓
BuzzFace	✓			✓	✓			✓
FacebookHoax	✓		✓	✓				
FakeNewsNet	✓	✓	✓	✓	✓	✓	✓	✓

1.3 Класичні методи машинного навчання у виявленні дезінформації

Класичні методи машинного навчання – статистичні моделі, що ґрунтуються на ручному виборі ознак та класичних алгоритмах класифікації – стали першим систематичним підходом до автоматизованого виявлення дезінформації. Попри стрімкий розвиток нейромережевих методів, класичні підходи зберігають своє місце у дослідницькій спільноті завдяки обчислювальній простоті, високій швидкості тренування, інтерпретованості рішень та придатності для базового порівняння з більш складними архітектурами. До цього класу належать наївний баєсівський класифікатор, логістична регресія, метод опорних векторів (SVM), Random Forest та gradient boosting. Цей підрозділ детально розглядає наївний

баєсівський класифікатор, який у практичній частині роботи використовується як базова модель для порівняння з нейромережевими підходами.

1.3.1 Наївний баєсівський класифікатор

Наївний баєсівський класифікатор (Naive Bayes, NB) є одним з найбільш поширених алгоритмів у задачах класифікації тексту. В його основі лежить теорема Баєса, що дозволяє обчислювати апостеріорну ймовірність належності об'єкта до класу C при спостереженні набору ознак X через формулу:

$$P(C \mid X) = P(X \mid C) \cdot P(C) / P(X)$$

де $P(C \mid X)$ – апостеріорна ймовірність класу C при спостереженні ознак X; $P(X \mid C)$ – функція правдоподібності; $P(C)$ – апріорна ймовірність класу; $P(X)$ – нормуюча константа, що не залежить від класу і тому може ігноруватися при порівнянні класів. «Наївність» алгоритму полягає у ключовому припущенні про умовну незалежність ознак: вважається, що кожна ознака у тексті вносить незалежний внесок у визначення класу. Незважаючи на те, що це припущення зазвичай порушується у реальному тексті, де слова статистично залежні, на практиці модель демонструє конкурентоспроможні результати.

У текстовій класифікації найчастіше застосовується варіант Multinomial Naive Bayes, що працює з частотними ознаками (TF, TF-IDF). Для подолання проблеми нульових ймовірностей при зустрічі терма, відсутнього у тренувальній вибірці, застосовується Лапласове згладжування – додавання невеликої константи (зазвичай $\alpha = 1$) до всіх частот при оцінюванні ймовірностей.

У задачах виявлення дезінформації наївний баєсівський класифікатор традиційно виступає як основна базова модель. Гранік та Месюра [8] одними з перших застосували цей метод до завдання виявлення фейкових новин, протестувавши класифікатор на наборі даних з постами Facebook та зафіксувавши точність класифікації приблизно 74 % – результат, що автори оцінюють як гідний з огляду на відносну простоту моделі. У висновках роботи зазначається, що

результат може бути покращений за рахунок попередньої обробки тексту, регуляризації та використання додаткових стилістичних ознак.

До ключових переваг наївного баєсівського класифікатора у контексті виявлення дезінформації належать обчислювальна ефективність, інтерпретованість (можливість витягнути перелік найвпливовіших ознак для конкретного рішення через значення log-likelihoods) та стійкість до високої розмірності (модель добре працює зі словниками з тисячами або десятками тисяч термів). Основними ж обмеженнями класичних методів машинного навчання є нездатність моделювати контекстуальні залежності між словами та залежність від якості ручного feature engineering – саме подолання цих обмежень стало стимулом для розвитку нейромережевих методів, що розглядаються у наступному підрозділі.

1.4 Нейромережеві методи: трансформери

Серед різноманіття нейромережевих архітектур, що застосовуються до задач обробки природної мови, найбільший вплив протягом останнього десятиліття справили трансформерні моделі. Цей підрозділ розглядає фундаментальну архітектуру трансформера, її розвиток у напрямі попередньо натренованих моделей на прикладі BERT, а також ефективну дистильовану модифікацію DistilBERT, що використовується у практичній частині роботи.

1.4.1 Архітектура трансформера та механізм уваги

Архітектура трансформера, представлена Vaswani та співавторами у роботі «Attention Is All You Need» [14], стала революційним кроком у розвитку обробки природної мови. До її появи стандартом для моделювання послідовностей були рекурентні нейронні мережі (RNN, LSTM, GRU), що обробляли текст послідовно – токен за токеном. Це створювало два принципові обмеження: неможливість паралельного обчислення та згасання градієнта при моделюванні довгих залежностей.

Трансформер відмовився від рекурентності на користь механізму самоуваги (self-attention), що дозволяє кожному токену безпосередньо «звертати увагу» на всі інші токени у послідовності. Базова операція уваги формалізується як:

$$Attention(Q, K, V) = softmax(QK^T / \sqrt{d_k}) \cdot V$$

де Q (Query), K (Key), V (Value) – лінійні проєкції вхідного представлення; d_k – розмірність ключів, що використовується як фактор масштабування для стабілізації градієнтів. У результаті обчислення кожен токен отримує контекстуалізоване представлення як зважену суму представлень усіх інших токенів, де вагові коефіцієнти визначаються через скалярний добуток query та key. Класична архітектура трансформера складається зі стеку енкодерів та декодерів з шарами multi-head attention (паралельне застосування кількох механізмів уваги для одночасного врахування різних типів залежностей) та позиційного кодування для збереження інформації про порядок токенів.

1.4.2 BERT та парадигма попереднього навчання

BERT (Bidirectional Encoder Representations from Transformers), представлений Devlin та співавторами [15], став першою трансформерною моделлю, що реалізувала парадигму глибокого двонаправленого попереднього навчання на великих корпусах нерозміченого тексту з подальшим тонким налаштуванням (fine-tuning) під конкретні задачі. На відміну від однонаправленої моделі GPT-1, що вийшла кількома місяцями раніше, BERT навчається будувати двонаправлені контекстуальні представлення – кожен токен отримує вектор, що враховує як попередні, так і наступні токени у реченні.

Попереднє навчання BERT здійснюється через дві задачі. Перша – Masked Language Modeling (MLM): випадково маскується 15% токенів вхідного тексту, які модель навчається відновлювати на основі контексту, що формує глибоке розуміння лексичної семантики та синтаксичних залежностей. Друга – Next Sentence Prediction (NSP): задача класифікації пари речень на предмет того, чи є друге речення логічним продовженням першого, що формує розуміння

міжречевих зв'язків. Для задач класифікації до виходу спеціального токена [CLS] додається лінійний класифікаційний шар, після чого вся модель проходить fine-tuning на розміченому датасеті конкретної задачі. Такий підхід вимагає на порядки менше розмічених прикладів порівняно з навчанням з нуля та забезпечує state-of-the-art результати на широкому спектрі NLP-задач. У задачах виявлення дезінформації BERT та його модифікації стали стандартним інструментом, що перевершує класичні методи на більшості бенчмарків.

1.4.3 DistilBERT як ефективна дистильована альтернатива

Головним недоліком BERT-base моделі (110 мільйонів параметрів) є висока обчислювальна вартість тренування та інференсу. Розв'язанням цієї проблеми стала техніка дистиляції знань (knowledge distillation), що дозволяє переносити «знання» від великої моделі-вчителя до меншої моделі-учня.

DistilBERT [16] є дистильованою версією BERT-base, що зменшує розмір на 40% (66 мільйонів параметрів) при збереженні 97% ефективності BERT на бенчмарку GLUE та забезпечує приблизно у півтора рази швидший інференс. Для досягнення такого балансу автори застосовують комбіновану функцію втрат з трьох компонентів: стандартна функція Masked Language Modeling; soft cross-entropy loss між розподілами ймовірностей моделі-учня та моделі-вчителя (distillation loss); та cosine embedding loss для вирівнювання напрямків векторів прихованих станів обох моделей.

У задачах виявлення дезінформації DistilBERT демонструє ефективність, що наближається до повного BERT при істотно менших обчислювальних витратах. Зокрема, у дослідженні Patel та співавторів на датасеті FakeNewsNet GossipCop DistilBERT показав точність, порівнянну з RoBERTa, при значно нижчих вимогах до обчислювальних ресурсів [17]. Це робить DistilBERT оптимальним вибором для дослідницького прототипу з обмеженими GPU-ресурсами, що детально розглядається у наступному розділі.

1.5 Графові нейронні мережі для виявлення дезінформації

Класичні методи машинного навчання та трансформерні моделі, розглянуті у попередніх підрозділах, обробляють текстові дані як ізольовані одиниці – окремі документи або послідовності токенів – ігноруючи при цьому принципову структурну особливість поширення інформації у соціальних мережах: наявність явних зв'язків між публікаціями, користувачами та їх взаємодіями. Графові нейронні мережі (Graph Neural Networks, GNN) становлять окремий клас нейромережевих архітектур, що спеціально призначені для обробки даних з графовою структурою та дозволяють моделювати складні відносини між сутностями у мережі поширення дезінформації.

1.5.1 Основні принципи графових нейронних мереж

Графова нейронна мережа оперує на вхідному графі $G = (V, E)$, де V – множина вершин (nodes), що відповідають окремим сутностям (статті, користувачі, твіти), а E – множина ребер (edges), що відповідають відносинам між ними (поширення, цитування, підписки). Кожна вершина $v \in V$ має асоційований вектор ознак x_v, що може містити текстові вкладення, метадані користувача або інші атрибути. Центральною операцією GNN є механізм передачі повідомлень (message passing), що ітеративно оновлює представлення кожної вершини шляхом агрегації інформації від її сусідів у графі:

$$h_v^{(k)} = UPDATE^{(k)}(h_v^{(k-1)}, AGGREGATE^{(k)}(\{h_u^{(k-1)} : u \in N(v)\}))$$

де $h_v^{(k)}$ – представлення вершини v після k ітерацій; $N(v)$ – множина сусідів вершини v; *AGGREGATE* – функція агрегації, що об'єднує представлення сусідніх вершин; *UPDATE* – функція оновлення, що інтегрує попереднє представлення вершини з агрегованою інформацією від сусідів. Після K ітерацій message passing кожна вершина отримує контекстно-залежне представлення, що враховує інформацію з її K-hop околиці у графі. Ключова перевага GNN полягає у здатності моделювати нелокальні залежності – інформація агрегується через структуру графа

на глибину K кроків, що дозволяє захоплювати складні структурні паттерни, недоступні текстовим моделям.

1.5.2 Архітектури GNN: GraphSAGE та Graph Isomorphism Network

Серед різноманіття архітектур графових нейронних мереж дві отримали найширше застосування у задачах виявлення дезінформації: GraphSAGE та Graph Isomorphism Network (GIN).

GraphSAGE (Graph Sample and Aggregate) [9] вирішує проблему масштабованості класичних GNN на великих графах через механізм семплювання сусідів. Замість агрегації всіх сусідів вершини, що може бути обчислювально неможливим у графах з мільйонами вершин, GraphSAGE семплює фіксовану кількість сусідів на кожному кроці та агрегує їх представлення через навчувані функції агрегації. Це дозволяє моделі масштабуватися до графів соціальних мереж з мільйонами користувачів та забезпечує можливість inductive learning – застосування натренованої моделі до нових, раніше не бачених вершин графа без перетренування.

Graph Isomorphism Network (GIN) [10] є теоретично обґрунтованою архітектурою, що досягає максимальної виразності серед GNN у сенсі Weisfeiler-Lehman graph isomorphism test. Ключовою особливістю GIN є використання багатошарового перцептрона (MLP) у функції UPDATE та навчуваного параметра ϵ для зважування власного представлення вершини відносно агрегованої інформації від сусідів. Теоретичні гарантії GIN забезпечують її здатність розрізняти широкий клас графових структур, що робить цю архітектуру особливо привабливою для задач, де топологічні паттерни є критичними індикаторами.

1.5.3 Застосування GNN до виявлення дезінформації

У задачах виявлення дезінформації графові нейронні мережі застосовуються до кількох типів графових структур. **Граф поширення** – вершини представляють окремі твіти або пости, а ребра кодують відносини ретвіту або цитування; такий граф фіксує каскад поширення інформації та дозволяє моделювати її часову динаміку [3]. **Соціальний граф** – вершини представляють користувачів, а ребра – підписки, згадування або інші форми соціальної взаємодії; аналіз такого графа дозволяє виявляти координовані мережі поширювачів дезінформації [11]. **Гетерогенний граф** об'єднує різні типи вершин (статті, користувачі, твіти) та ребер (автор твіту, поширює статтю, підписаний на користувача) у єдину мультимодальну структуру, що відповідає моделі Tri-Relationship [2].

Емпіричні дослідження демонструють ефективність GNN у виявленні дезінформації. Зокрема, у роботі Monti та співавторів [12] автори застосували графовий підхід до датасетів Twitter15 та Twitter16 і досягли точності класифікації 85-90%, що суттєво перевищує результати простіших моделей на основі текстових ознак. Ключовим висновком дослідження стало те, що графові паттерни поширення несуть комплементарну інформацію до текстового контенту – поєднання GNN з текстовими класифікаторами дає вищу точність, ніж будь-який з підходів окремо.

1.5.4 Обмеження та виклики графових підходів

Попри теоретичні переваги, графові нейронні мережі мають кілька принципових обмежень у контексті виявлення дезінформації. По-перше, необхідність повного графа – для тренування та інференсу GNN потрібна інформація про зв'язки між сутностями, яка не завжди доступна у публічних датасетах або може бути недоступна через обмеження API соціальних платформ. По-друге, обчислювальна складність – хоча архітектури на кшталт GraphSAGE частково вирішують проблему масштабованості, тренування GNN на графах з мільйонами вершин все ще вимагає суттєвих обчислювальних ресурсів. По-третє, часова динаміка – більшість стандартних архітектур GNN припускають статичну

структуру графа, тоді як реальні соціальні графи постійно змінюються; розширення класичних GNN для динамічних графів (Temporal GNN) становить активний напрям досліджень [13], проте їх практичне застосування ще обмежене. По-четверте, інтерпретованість – на відміну від лінійних моделей, де можна інспектувати вагові коефіцієнти найвпливовіших ознак, GNN діють як чорні скриньки, що ускладнює розуміння причин конкретних рішень моделі. У межах практичної частини магістерської роботи GNN-підходи реалізовані на базі архітектур GraphSAGE та GIN з використанням графів поширення FakeNewsNet, що детально описано у Розділі 3.

1.6 Великі мовні моделі у виявленні дезінформації

Розвиток нейромережевих архітектур у напрямі масштабування призвів до появи якісно нової категорії моделей – великих мовних моделей (Large Language Models, LLM), що відрізняються від попередників не лише розміром, але й способом застосування до прикладних задач. Якщо BERT та DistilBERT потребують тонкого налаштування під конкретну задачу з використанням розміченого датасету, LLM здатні розв'язувати нові задачі лише через відповідне формулювання запиту, без оновлення вагових коефіцієнтів моделі. Цей підрозділ розглядає принципи парадигми zero-shot та few-shot класифікації, сімейство моделей Claude від компанії Anthropic, що використовується у практичній частині магістерської роботи, а також критичний огляд досліджень з застосування LLM до задач виявлення дезінформації.

1.6.1 Парадигма zero-shot та few-shot класифікації

Систематичну демонстрацію можливостей zero-shot та few-shot навчання у великих мовних моделях представлено у роботі Брауна та співавторів «Language Models are Few-Shot Learners» [18] на прикладі моделі GPT-3 з 175 мільярдами параметрів. Автори продемонстрували, що достатньо великі попередньо

натреновані мовні моделі здатні розв'язувати нові задачі лише за умови надання їм природномовного опису задачі (zero-shot) або кількох прикладів виконання (few-shot) без оновлення вагових коефіцієнтів моделі. Технічно цей механізм реалізується через in-context learning: модель отримує на вхід промпт з описом задачі, опційно кілька прикладів та конкретний запит, після чого згенерована відповідь інтерпретується як прогноз.

Розрізняють два основні режими роботи з LLM. Zero-shot режим передбачає лише природномовний опис задачі без жодних прикладів. Few-shot режим доповнює інструкцію кількома прикладами виконання задачі, що значно покращує точність – Браун та співавтори показали, що для більшості задач few-shot performance росте з кількістю прикладів швидше за zero-shot.

1.6.2 Сімейство моделей Claude від Anthropic

Сімейство моделей Claude розроблене компанією Anthropic та представлене у березні 2024 року як набір з трьох моделей у порядку зростання обчислювальних можливостей: Claude Haiku – найшвидша та економічно ефективна модель для базових задач, Claude Sonnet – модель середнього класу з оптимальним балансом швидкості та інтелектуальних можливостей, Claude Opus – найпотужніша модель сімейства, призначена для найскладніших задач [20]. Усі моделі сімейства є мультимодальними та підтримують обробку як текстових, так і візуальних входів.

Принциповою особливістю моделей Claude є застосування методології Constitutional AI – підходу, розробленого Anthropic для покращення відповідності моделей етичним нормам. У межах цього підходу модель тренується дотримуватися набору принципів, що походять з кількох джерел – Загальної декларації прав людини ООН, найкращих практик у сфері безпеки, принципів, запропонованих іншими дослідницькими лабораторіями, а також внутрішніх досліджень Anthropic. У результаті Claude демонструє меншу схильність до невиправданих відмов у відповідях на запити, що знаходяться у межах допустимого, порівняно з попередніми поколіннями моделей.

Для задач виявлення дезінформації моделі Claude мають кілька важливих переваг: контекстне вікно у двісті тисяч токенів дозволяє обробляти великі тексти статей цілком, без необхідності розбиття на фрагменти; моделі демонструють високу точність у задачах, що вимагають аналітичних здібностей – оцінювання правдоподібності тверджень, виявлення внутрішніх суперечностей у тексті, ідентифікація маніпулятивних риторичних прийомів; підтримка структурованих відповідей у форматі JSON спрощує інтеграцію у автоматизовані системи. У межах практичної частини магістерської роботи використовується модель Claude Haiku як представник парадигми великих мовних моделей.

1.6.3 Огляд досліджень з застосування LLM до виявлення дезінформації

Систематичне дослідження ефективності великих мовних моделей у задачі виявлення дезінформації проведене у роботі «Bad Actor, Good Advisor» [21]. Автори провели порівняльний експеримент між GPT-3.5-turbo у чотирьох режимах промптингу (zero-shot, few-shot, zero-shot з chain-of-thought, few-shot з chain-of-thought) та тонко налаштованою моделлю BERT-base на двох датасетах виявлення дезінформації – англомовному GossipCop та китайськомовному Weibo21.

Результати цього дослідження виявилися неочікуваними з огляду на загальне переконання щодо переваги великих моделей. У всіх чотирьох режимах промптингу GPT-3.5-turbo поступився тонко налаштованій BERT-моделі: відносний приріст BERT над GPT-3.5 складав від 3.8% до 34.6% залежно від режиму. Few-shot режими демонстрували вищу точність порівняно з zero-shot, проте навіть у найкращій конфігурації не змогли перевершити тонко налаштовану BERT. Chain-of-thought промптинг загалом покращував результати, проте у деяких випадках навпаки призводив до зниження точності.

Автори [21] інтерпретують ці результати як свідчення того, що великі мовні моделі у режимах prompt-based класифікації не здатні замінити тонко налаштовані малі моделі у задачах виявлення дезінформації. Причиною є нездатність LLM

ефективно інтегрувати численні релевантні аспекти аналізу у єдиний фінальний висновок – модель може формулювати правильні часткові спостереження, проте робить помилкові узагальнення. На основі цього спостереження автори запропонували архітектуру ARG (Adaptive Rationale Guidance), у якій LLM відіграє роль «радника», що генерує мультиперспективні раціонали, тоді як фінальний висновок робить тонко налаштована мала модель. Ці результати не означають, проте, що великі мовні моделі непридатні для задач виявлення дезінформації – вони лише вказують на обмеженість режиму прямої класифікації. Як буде показано у наступному розділі, LLM знаходять важливе застосування у допоміжних задачах: витягування перевірюваних claims з тексту, формулювання пошукових запитів до фактчекер-сервісів, генерація пояснень рішень моделі. Саме у такому допоміжному режимі Claude Haiku інтегрується у програмний прототип системи, що детально описано у Розділі 2.

1.7 Feature engineering для текстової класифікації

Класичні методи машинного навчання, що розглядалися у підрозділі 1.3, не здатні самостійно витягувати з тексту значущі ознаки – модель отримує вхід у вигляді числового вектора фіксованої розмірності, тому перетворення сирого тексту на такий вектор становить окремий етап побудови класифікатора, що називається feature engineering. Якість фінального класифікатора у значній мірі визначається саме якістю формованого простору ознак – навіть найдосконаліший класифікатор не зможе досягти високої точності за наявності неінформативних ознак. Цей підрозділ розглядає чотири основні категорії ознак, що традиційно використовуються у задачах виявлення дезінформації: статистичні представлення на основі TF-IDF, емоційні ознаки на основі лексичних ресурсів, стилістичні маркери та соціальні ознаки на основі метаданих авторів.

1.7.1 TF-IDF та статистичні представлення

TF-IDF (Term Frequency – Inverse Document Frequency) є класичним методом векторизації тексту, що зважує важливість слова у документі з урахуванням його загальної поширеності у всьому корпусі. Метод був формалізований Салтоном та Баклі у роботі 1988 року «Term-Weighting Approaches in Automatic Text Retrieval» [22] як основний підхід до текстового пошуку у векторному просторі та згодом став стандартним інструментом текстової класифікації. Формальне визначення TF-IDF має вигляд:

$$TF\text{-}IDF(t, d, D) = TF(t, d) \cdot \log(|D| / |\{d \in D : t \in d\}|) \quad (1.3)$$

де TF(t, d) – частота терма t у документі d; |D| – загальна кількість документів у корпусі; знаменник у логарифмі – кількість документів, що містять терм t. Принцип роботи методу можна описати так: терми, що часто з'являються у конкретному документі (висока TF), але рідко зустрічаються у решті корпусу (висока IDF), отримують найбільшу вагу та вважаються найбільш характерними для цього документа. Натомість дуже поширені у корпусі службові слова отримують низьку вагу, оскільки не несуть значущої інформації для розрізнення документів.

На практиці TF-IDF-векторизація реалізується через формування словника всіх унікальних термів корпусу та подальше відображення кожного документа у розріджений вектор фіксованої розмірності, де ненульові компоненти відповідають TF-IDF-вагам термів, що зустрічаються у цьому документі. Стандартні модифікації методу включають врахування біграм (пар сусідніх слів) поряд з окремими токенами, що дозволяє фіксувати стійкі словосполучення на кшталт «breaking news», «sources say», «fact check» – характерні для журналістського стилю та маркери, що часто експлуатуються у дезінформаційних публікаціях. Сублінеарне нормалізування TF (заміна частоти слова на її логарифм) знижує вплив надмірно частих слів та емпірично покращує точність класифікації у задачах виявлення дезінформації.

1.7.2 Емоційні ознаки на основі лексикону NRC EmoLex

Однією з характерних рис дезінформаційного контенту є його емоційна забарвленість – дезінформаційні публікації, як правило, апелюють до сильних емоцій (страх, обурення, подив) для забезпечення вірального поширення. Ця особливість лежить в основі цілого напряму feature engineering, що використовує лексичні ресурси для кількісної оцінки емоційного навантаження тексту.

Найбільш широко використовуваним лексиконом цього типу є NRC Emotion Lexicon (NRC EmoLex) [23]. Лексикон містить понад чотирнадцять тисяч англійських слів, кожне з яких розмічене за наявністю асоціації з вісьмома базовими емоціями за моделлю Плутчика – anger, anticipation, disgust, fear, joy, sadness, surprise, trust – а також з двома сентиментними категоріями (positive, negative). Окрім базової версії, лексикон NRC EmoLex має розширення NRC Emotion Intensity Lexicon з числовими оцінками інтенсивності у діапазоні від нуля до одиниці – це дозволяє розрізняти, наприклад, легкий смуток від глибокого горя у текстах з однаковою кількістю «sad»-слів.

Використання NRC EmoLex для feature engineering реалізується через простий алгоритм: для кожного документа обчислюється частка слів, асоційованих з кожною категорією емоцій, що формує базовий десятивимірний вектор емоційного профілю тексту. До цього вектора можна додати похідні ознаки інтенсивності з NRC Emotion Intensity Lexicon, що дозволяє точніше моделювати емоційне забарвлення тексту у задачі виявлення дезінформації.

1.7.3 Стилістичні та риторичні маркери дезінформації

Третя категорія ознак ґрунтується на спостереженні, що фейкові новини мають характерні стилістичні відмінності від легітимних новинних публікацій. Ключове дослідження у цьому напрямі [24] виявило статистично значущі стилістичні відмінності між фейковими та правдивими новинами: основний текст короткий та простий, з більшою часткою повторів та нижчим індексом читабельності; стилістично фейки ближче до сатиричного контенту, ніж до якісних журналістських матеріалів – у вигляді більшої частки знаків оклику, питань та слів

у верхньому регістрі. Автори дійшли висновку, що переконливість фейкових публікацій досягається не через якість аргументації, а через евристики – створення ментальних асоціацій у заголовку, до якого часто й обмежується сприйняття читача.

На основі цих спостережень формується типовий набір стилістичних ознак для feature engineering: довжина тексту у словах, лексичне різноманіття (відношення кількості унікальних слів до загальної), середня довжина речення, частка довгих слів (понад шість символів) як індикатор формальності, частка знаків оклику та питань, частка слів у верхньому регістрі, наявність clickbait-патернів на кшталт «You Won't Believe», що детектуються через регулярні вирази.

1.7.4 Соціальні ознаки на основі метаданих авторів

Окрема категорія ознак не пов'язана зі змістом самої публікації, а ґрунтується на метаданих користувачів, що поширювали публікацію у соціальній мережі. Цей підхід ґрунтується на гіпотезі про те, що дезінформаційний контент розповсюджується переважно специфічним типом користувачів, відмінним за поведінковими характеристиками від звичайних користувачів, і безпосередньо випливає з моделі Tri-Relationship [2], що передбачає одночасне врахування контенту, соціального контексту та часової динаміки.

У роботі використано 23 соціальні ознаки, згрупованих у чотири категорії. Лічильники профілю (6 ознак) – log-нормалізовані характеристики масштабу акаунту: кількість підписників, підписок, опублікованих статусів, вік акаунту, частота публікацій (кількість статусів на день як сигнал активності бота) та відношення підписників до підписок (followers/friends ratio як індикатор впливовості чи автоматизованого поведінки). Атрибути профілю (6 ознак) – бінарні та нормалізовані характеристики «обличчя» акаунту: статус верифікації, наявність заповненого опису та вказаної локації, довжина опису та імені користувача, а також частка цифр у імені користувача (автоматично згенеровані бот-акаунти типу user1738492 мають характерно высокий показник). Engagement-метрики (5 ознак)

– характеристики реакції аудиторії на конкретний твіт: кількість лайків, ретвітів, відповідей, відношення лайків до ретвітів (правдиві новини мають вищий показник, оскільки люди частіше лайкають, ніж ретвітять достовірний контент) та загальний engagement rate відносно розміру аудиторії. Характеристики каскаду поширення (6 ознак) – структурні параметри всього графа поширення статті: глибина та ширина дерева реплаїв, тривалість поширення у годинах, середня кількість ретвітів та відповідей на твіт, кількість унікальних учасників каскаду.

При обробці на рівні статті лічильники профілю та engagement-метрики агрегуються (середнє значення) по всіх твітах, що поширюють публікацію, тоді як каскадні ознаки обчислюються один раз для всього графа поширення статті. Емпірична перевірка ефективності соціальних ознак у різних дослідженнях дала змішані результати залежно від домену – це питання детально досліджується у підрозділі 3.3.

1.8 Інтеграція з фактчекер-сервісами як підхід до верифікації

Розглянуті у попередніх підрозділах підходи ґрунтуються на пошуку статистичних або стилістичних патернів у тексті без безпосередньої перевірки фактологічної бази стверджень. Природним доповненням до автоматизованих моделей є фактчекер-сервіси, що надають експертно перевірені вердикти на основі реальної верифікації фактів. У межах даної роботи такі сервіси використовуються не як компонент класифікаційного механізму, а як незалежний інструмент верифікації результатів моделі – для зіставлення статистичного прогнозу з експертним висновком.

Фактчек-екосистема об'єднана навколо Міжнародної мережі фактчекерів (International Fact-Checking Network, IFCN) [25], що сертифікує організації за принципами непартійності, прозорості джерел та методології. До найвідоміших членів IFCN належать Snopes, PolitiFact, AFP Fact Check, Reuters Fact Check та GossipCop, який є одним із джерел розмітки датасету FakeNewsNet. Для уніфікованого доступу до результатів цих організацій існує Google Fact Check Tools

API [26], що агрегує понад триста фактчекерських джерел та повертає структуровані метадані у стандарті ClaimReview. Перед запитом до фактчекер-сервісу з тексту публікації витягуються конкретні твердження (claims), що піддаються перевірці – задача, яку класично розв'язували системи на кшталт ClaimBuster [27], а з розвитком великих мовних моделей вона все частіше виконується через prompt-based підходи.

Технічна реалізація механізму паралельної верифікації, що дозволяє порівняти висновок натренованої моделі з консенсусом фактчекерів, детально розглянута у Розділі 2.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОТОТИПУ СИСТЕМИ ВИЯВЛЕННЯ ДЕЗІНФОРМАЦІЇ

2.1 Функціональні можливості системи

Розроблений програмний прототип являє собою веб-систему для автоматизованого виявлення дезінформації у новинних публікаціях та постах соціальних мереж. На відміну від поширених монолітних рішень, що зосереджені на одному методі класифікації, представлена система інтегрує чотири парадигми машинного навчання: статистичну (наївний баєсівський класифікатор), нейромережеву (тонке налаштування DistilBERT), графову (GraphSAGE та GIN на графах поширення) та підхід на основі великих мовних моделей (Claude Haiku через два режими промптингу). Така інтегрованість надає можливість провести порівняльне оцінювання ефективності різних архітектур у задачі бінарної класифікації текстів за категоріями FAKE та REAL у межах єдиного експериментального середовища.

2.1.1 Основні можливості системи

Система надає комплексний набір інструментів для роботи з виявленням дезінформації на всіх етапах дослідницького циклу. Загальну структуру функціональних можливостей у вигляді Use Case діаграми представлено на рисунку 2.1.

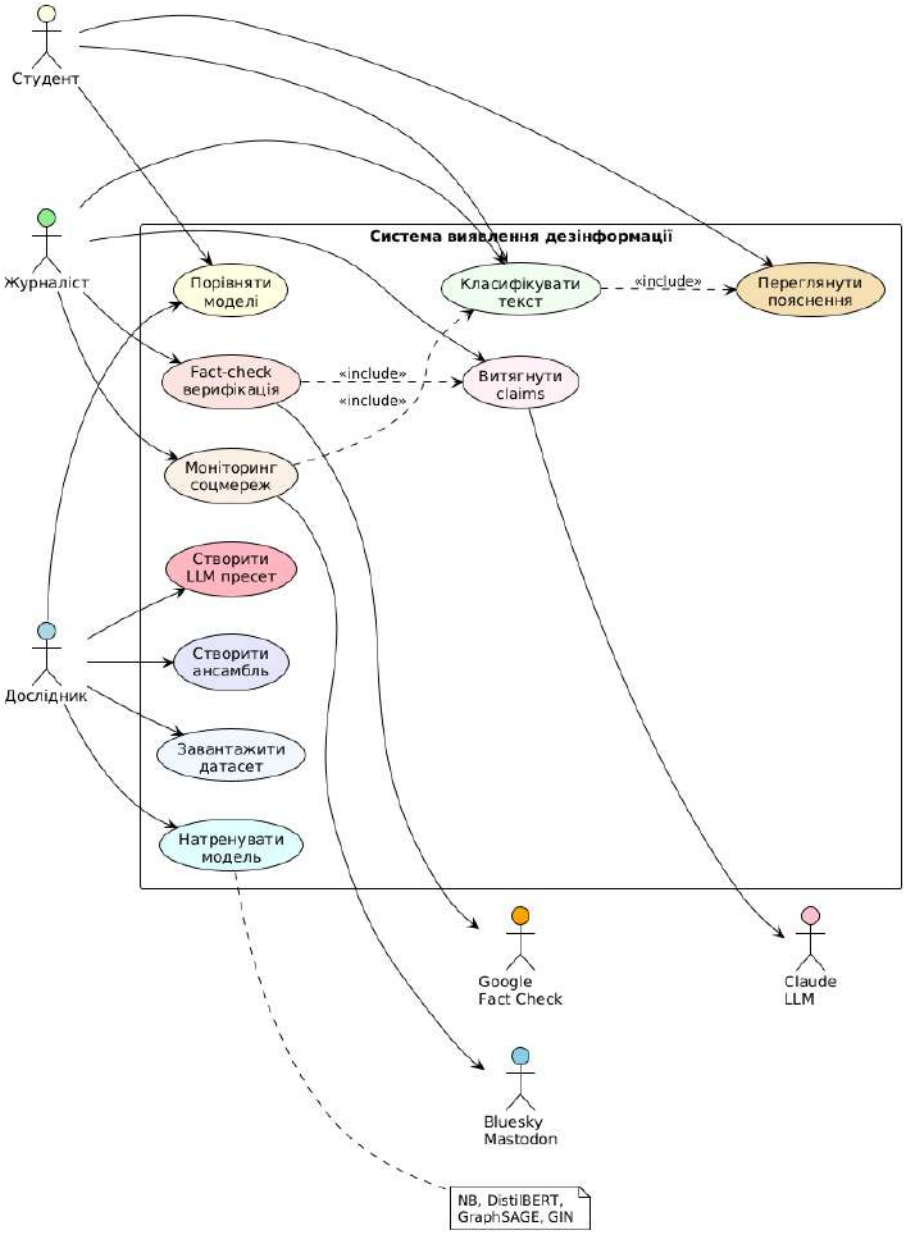

***Рисунок 2.1**. Use Case діаграма функціональних можливостей системи*

Нижче наведено структурований перелік функціональних можливостей, згрупованих за категоріями. Навігаційне меню системи, що забезпечує доступ до всіх розділів, зображено на рисунку 2.2.

Управління датасетами:

- Завантаження власних датасетів та сплітів даних у форматі ZIP-архіву з валідацією структури

- Обчислення статистики розподілу класів FAKE/REAL

- Активація датасету для використання у тренуванні моделей

Навчання моделей:

- Тренування чотирьох типів моделей: Naive Bayes, DistilBERT, GraphSAGE, GIN
- Конфігурація гіперпараметрів та набору ознак для кожної моделі
- Автоматичне обчислення feature engineering (TF-IDF, емоційні, стилістичні, соціальні ознаки)
- Відстеження прогресу тренування у реальному часі
- Збереження натренованих моделей у Google Drive користувача

Робота з моделями:

- Перегляд списку всіх натренованих моделей з метриками
- Перегляд детальних метрик (accuracy, F1(FAKE), precision(FAKE), recall(FAKE), F1-macro, ROC-AUC)
- Видалення моделей

Класифікація текстів:

- Інтерактивна класифікація довільного тексту обраною моделлю
- Отримання мітки класу (FAKE/REAL) з показником упевненості (confidence)
- Для Naive Bayes – перелік найвпливовіших ознак з поясненням рішення
- Для DistilBERT – візуалізація attribution через Integrated Gradients з підсвіткою впливових токенів
- Для LLM – текстове обґрунтування висновку моделі

LLM пресети:

- Створення іменованих конфігурацій для великих мовних моделей
- Вибір моделі (Claude Haiku, Sonnet, Opus)
- Вибір режиму взаємодії: zero-shot, few-shot
- Налаштування системного промпту та параметрів генерації
- Для few-shot режиму – редактор прикладів класифікації

Ансамблі:

- Створення ансамблевих моделей з комбінації натренованих моделей
- Вибір стратегії об'єднання: hard voting, soft voting, weighted voting

Верифікація через фактчекерів:

- Витягування перевірюваних тверджень (claims) з тексту через LLM з визначенням позиції автора (supports/refutes/neutral)
- Автоматичний пошук фактчек-публікацій через Google Fact Check Tools API
- Нормалізація рейтингів різних фактчекер-організацій до єдиної шкали (FAKE/REAL/MIXED)
- Обчислення загального консенсусу на основі мажоритарного голосування claims
- Порівняння висновку моделі з консенсусом фактчекерів
- Відображення статусу збігу (MATCH/MISMATCH) з детальною розбивкою по claims

Моніторинг соціальних мереж:

- Інтеграція з Bluesky AT Protocol та Mastodon API
- Пошук публічних публікацій за ключовими словами
- Автоматична класифікація знайдених постів обраною моделлю (покроково або масово)
- Додаткова верифікація результатів через фактчекерів
- Візуалізація результатів з маркуванням потенційної дезінформації та статистикою

Аналіз постів:

- Комплексний аналіз окремого поста з соціальної мережі
- Класифікація тексту
- Витягування claims
- Fact-checking через Google API
- Детальний звіт про результати верифікації

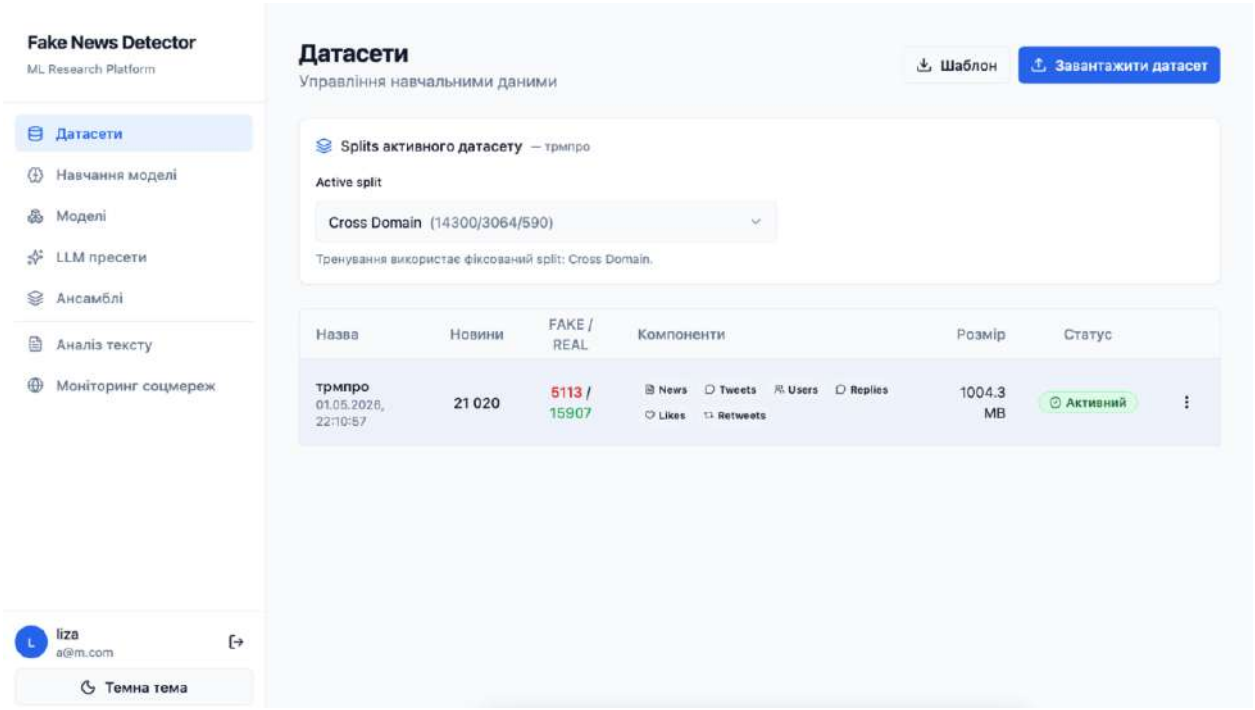

***Рисунок 2.2**. Головна сторінка системи з навігаційним меню*

2.1.2 Цільова аудиторія та сценарії використання

Цільову аудиторію системи становлять три основні категорії користувачів, кожна з яких має специфічні потреби та сценарії використання.

Дослідники у галузі обробки природної мови використовують систему як експериментальне середовище для порівняльного аналізу ефективності різних архітектур машинного навчання. Типовий сценарій: дослідник завантажує датасет FakeNewsNet, конфігурує набір моделей для тренування (наприклад, NB з різними наборами ознак, DistilBERT з різною кількістю епох), після чого порівнює їх ефективність на єдиній тестовій вибірці. Система автоматично обчислює метрики та надає можливість порівняти результати, що дозволяє виявити оптимальну конфігурацію для конкретного домену.

Журналісти та фактчекери використовують систему як інструмент попередньої перевірки публікацій. Типовий сценарій: фактчекер вводить текст сумнівної новини у розділ "Аналіз тексту", система класифікує її через кілька моделей та надає висновок. Додатково система витягує перевірювані твердження

та автоматично шукає їх у базі фактчек-публікацій, повертаючи посилання на оригінальні перевірки від визнаних організацій. Це дозволяє фактчекеру швидко отримати попередню оцінку та посилання на релевантні джерела для подальшого дослідження.

Студенти та викладачі профільних дисциплін використовують систему як навчальний інструмент для демонстрації принципів роботи різних моделей машинного навчання на реальних даних. Типовий сценарій: викладач проводить ablation-експеримент, послідовно змінюючи набір ознак для базової моделі. Спочатку тренується NB лише на лексичних ознаках (текстові слова), потім досліджуються емоційні маркери (сентимент та інтенсивність емоцій), далі – стилістичні патерни (великі літери, clickbait-фрази). Студенти спостерігають, як кожна група ознак впливає на accuracy та F1-score, і вчаться розуміти, що деякі типи дезінформації краще розпізнаються за емоційним забарвленням, інші – за стилістичними прийомами, а оптимальний вибір груп ознак залежить від домену та не завжди відповідає інтуїтивному очікуванню "більше ознак – краще".

2.1.3 Обмеження прототипу

Розроблений прототип має обмеження, які важливо зафіксувати для коректної інтерпретації результатів подальшого експериментального дослідження.

По-перше, тренувальний датасет обмежено англомовним матеріалом FakeNewsNet з двох доменів – celebrity news (GossipCop) та політичних новин (PolitiFact). Внаслідок цього натреновані моделі демонструють найкращу ефективність на текстах подібного стилю; класифікація українськомовного контенту або матеріалів інших доменів виходить за межі поточного дослідження.

По-друге, ML-сервер реалізовано на базі безкоштовної підписки Google Colab, що накладає обмеження на тривалість безперервної роботи (до дванадцяти годин на сесію) та потребує оновлення ngrok-тунелю після кожного перезапуску. Для повноцінного production-розгортання необхідна заміна цього компонента на виділений GPU-інстанс через сервіси AWS EC2 або Google Cloud Compute Engine.

По-третє, інтеграція з великою мовною моделлю Claude використовує механізм Claude Code CLI, що обмежує кількість одночасних запитів обмеженнями плану підписки. Це є свідомим рішенням на користь дослідницької гнучкості – дозволяє уникнути додаткових витрат на API-токени під час експериментального етапу, проте у production-розгортанні цей компонент має бути замінений на безпосередню інтеграцію з Anthropic API.

По-четверте, GNN моделі працюють на графах поширення новин, що вимагає наявності структурованих даних про твіти, ретвіти та відповіді. Для датасетів, що містять лише тексти статей без соціального контексту, GNN-підходи не застосовні. У такому випадку система автоматично пропонує NB, DistilBERT та LLM-моделі як альтернативи.

Нарешті, поточна реалізація верифікації через фактчекер-організації покладається виключно на Google Fact Check Tools API, що агрегує результати близько трьохсот організацій. Це значно розширює покриття порівняно з окремими джерелами на кшталт Snopes чи PolitiFact, проте все ще не охоплює значну частину публікацій, особливо у нішевих темах та неангломовному контенті. Розширення джерел верифікації становить один із напрямів подальшого розвитку прототипу.

2.2 Архітектура системи та технологічний стек

2.2.1 Загальна архітектура

В основу архітектури системи (рис. 2.3) покладено принцип розподіленої клієнт-серверної архітектури з чіткою декомпозицією за рівнями відповідальності. Система складається з трьох логічних рівнів: рівня клієнта (Frontend), рівня бізнес-логіки (Backend API), рівня машинного навчання (ML Server). Кожен рівень має чітко визначені межі відповідальності, спілкується з іншими рівнями через REST API і може бути розгорнутий незалежно.

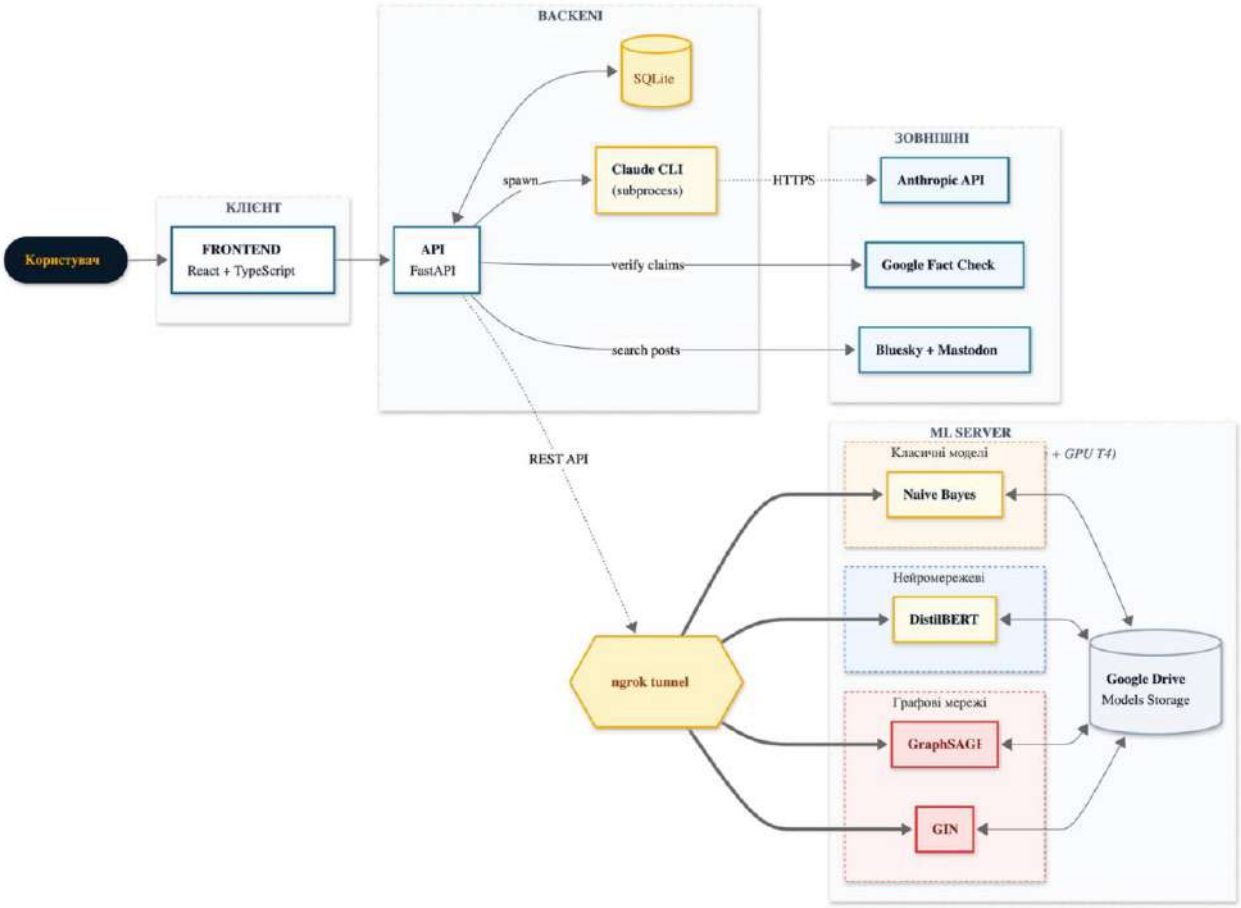

***Рисунок 2.3**. Архітектура системи виявлення дезінформації*

Рівень клієнта (Frontend) реалізовано як односторінковий веб-застосунок на основі бібліотеки React з TypeScript. Клієнт відповідає виключно за відображення інтерфейсу та взаємодію з користувачем – він не містить бізнес-логіки, що працює з даними напряму. Усі операції з даними виконуються через REST API запити до серверної частини. Стилізація реалізована через Tailwind CSS та компонентну бібліотеку shadcn/ui, що забезпечує сучасний та консистентний дизайн.

Рівень бізнес-логіки (Backend API) реалізовано як REST API сервер на основі фреймворка FastAPI. Цей рівень відповідає за прийняття та валідацію запитів від клієнта, координацію взаємодії з базою даних SQLite через ORM-бібліотеку SQLAlchemy, проксіювання запитів до ML-серверу, інтеграцію з зовнішніми сервісами (Bluesky, Mastodon, Google Fact Check, Claude). Backend є центральним компонентом системи – усі інші компоненти спілкуються виключно через нього.

Рівень машинного навчання (ML Server) реалізовано як окремий Flask-сервер, що запущений у середовищі Google Colab з доступом до GPU T4. ML-сервер виконує тренування моделей (NB, DistilBERT, GraphSAGE, GIN), класифікацію текстів та обчислення метрик. Винесення цього компонента в окремий процес дозволяє ізолювати обчислювально-важкі завдання від легкого backend, а також використовувати безкоштовні GPU-ресурси Google Colab без необхідності платних інстансів.

Окрім зазначених рівнів, до складу системи входять допоміжні сховища: SQLite для зберігання інформації про користувачів, датасети, моделі та ансамблі, а також Google Drive для зберігання великих бінарних артефактів – натренованих моделей та самих датасетів.

2.2.2 Інтеграція з зовнішніми сервісами

Система інтегрована з чотирма категоріями зовнішніх сервісів, кожна з яких відіграє специфічну роль у функціональності.

Соціальні мережі. Bluesky AT Protocol та Mastodon API забезпечують доступ до публічних публікацій реальних користувачів. Bluesky обрано як молоду децентралізовану мережу з відкритим протоколом, що активно зростає станом на 2026 рік. Mastodon доповнює покриття як зріла федеративна мережа з широкою аудиторією. Обидва сервіси інтегровано через офіційні REST API з підтримкою пошуку за ключовими словами.

Сервіс верифікації фактів. Google Fact Check Tools API надає доступ до фактчек-публікацій понад трьохсот організацій по всьому світу, серед яких Snopes, PolitiFact, AFP Fact Check, Reuters Fact Check. Інтеграція реалізована як HTTP-запити з ключовими словами тверджень, витягнутих з тексту публікації через LLM. API повертає структуровані дані у стандартизованому форматі ClaimReview, що містять оригінальне твердження, оцінку фактчекера, посилання на публікацію та назву організації.

Велика мовна модель. Моделі сімейства Claude (Haiku, Sonnet, Opus) від Anthropic використовуються для трьох завдань: класифікації текстів у режимах zero-shot, few-shot, витягування перевірюваних тверджень (claim extraction) для подальшої верифікації, та генерації пояснень рішень моделі. Інтеграція реалізована через офіційний CLI-інструмент Claude Code, що дозволяє використовувати підписку без додаткових API-токенів. Для масової обробки реалізовано batch-режим, що об'єднує до дванадцяти текстів у єдиний промпт.

Допоміжні сервіси інфраструктури. Google Drive слугує сховищем бінарних артефактів моделей та датасетів, забезпечуючи їх збереженість незалежно від стану ефемерного середовища Google Colab. ngrok забезпечує публічний HTTPS-тунель до Flask-серверу у Colab, надаючи стабільну точку доступу попри динамічну природу Colab-сесій.

2.2.3 Технологічний стек

Вибір конкретних технологій для реалізації кожного з рівнів архітектури зумовлений набором функціональних і нефункціональних вимог та дослідницьким характером проєкту. Повний стек представлено у таблиці 2.1.

Таблиця 2.1. *Технологічний стек системи*

Рівень	**Технологія**	**Обґрунтування вибору**
Frontend	React 18 + TypeScript	Компонентна модель, типобезпека, зріла екосистема
	Tailwind CSS + shadcn/ui	Швидкість розробки, консистентний дизайн
	Vite	Hot Module Replacement, швидка збірка
Backend	FastAPI + Python 3.11	Async/await, автоматична валідація через Pydantic
	SQLAlchemy + SQLite	Декларативний ORM, легковага вбудована БД

	Uvicorn	Асинхронний ASGI-сервер для FastAPI
ML Server	Flask + Jupyter (Colab)	Простота інтеграції з notebook-середовищем
	PyTorch + transformers	Fine-tuning DistilBERT
	PyTorch Geometric	Реалізація GNN (GraphSAGE, GIN)
	scikit-learn	Naive Bayes, TF-IDF, метрики
	sentence-transformers	MiniLM ембединги для GNN node features
Інтеграції	Claude Code CLI	Zero-cost LLM inference через підписку
	Google Fact Check API	Агрегація 300+ фактчекер-організацій
	Bluesky AT Protocol	Доступ до децентралізованої соцмережі
	Mastodon API	Доступ до федеративної соцмережі
Інфраструктура	Google Colab (GPU T4)	Безкоштовні GPU-ресурси для тренування
	Google Drive	Зберігання моделей та датасетів
	ngrok	Публічний тунель до Colab-серверу

Ключовим архітектурним рішенням стало винесення ML-серверу у Google Colab. Це дозволило використати безкоштовні GPU-ресурси для тренування DistilBERT та GNN моделей, без необхідності орендувати платні GPU-інстанси. Для production-розгортання цей компонент має бути замінений на виділений сервер, проте для дослідницького прототипу такий підхід є оптимальним балансом між вартістю та можливостями.

2.3 Датасети та обробка даних

2.3.1 Структура датасету FakeNewsNet

Датасет FakeNewsNet складається з п'яти логічно пов'язаних таблиць у форматі CSV. Перша таблиця, news.csv, містить інформацію про окремі новинні статті: ідентифікатор article_id, заголовок article_title, повний текст article_text, мітку article_label (FAKE або REAL), джерело публікації article_source та URL article_url. Друга таблиця, tweets.csv, містить оригінальні твіти, що поширюють посилання на ці статті: ідентифікатор tweet_id, ідентифікатор автора user_id, текст твіту tweet_text, кількість лайків та ретвітів, ідентифікатор article_id, що пов'язує твіт зі статтею. Третя та четверта таблиці – retweets.csv та replies.csv – фіксують вторинне поширення публікацій: ретвіти оригінальних твітів та відповіді на них з ідентифікаторами користувачів та часовими мітками. П'ята таблиця, users.csv, містить агреговані метадані користувачів Twitter: кількість підписників, кількість підписок, дату створення облікового запису, прапорець верифікації.

Така структура задає каскад поширення новини як орієнтований граф: стаття → твіти → ретвіти/відповіді – що є основою для побудови графових моделей GraphSAGE та GIN.

2.3.2 Очищення та feature engineering

Сирий датасет FakeNewsNet містить артефакти, що потребують очищення: HTML-розмітку та JavaScript-код, дублікати статей з однаковим заголовком, порожні або занадто короткі тексти (менше п'ятдесяти символів). Очищення HTML здійснюється через бібліотеку BeautifulSoup, дедуплікація – через обчислення MD5-хешу від конкатенації заголовка та початку тексту. У результаті обробки початкові 35 178 статей зменшуються до 32 894 унікальних.

Для тренування Naive Bayes реалізовано feature engineering pipeline з чотирьох категорій ознак, теоретичні засади яких описано у підрозділах 1.7.1–1.7.4. TF-IDF ознаки обчислюються через TfidfVectorizer (scikit-learn); користувач налаштовує тип векторизатора (TF-IDF або CountVectorizer) та діапазон n-грамів ((1,1), (1,2) або (1,3)), тоді як параметри max_features=50 000, min_df=2,

max_df=0.95 та sublinear_tf=True зафіксовані у коді. Емоційні ознаки (14) формуються на основі лексиконів NRC EmoLex та NRC Emotion Intensity Lexicon. Стилістичні ознаки (8) обчислюються через регулярні вирази та статистики тексту. Соціальні ознаки (23) агрегуються з метаданих Twitter-каскадів у чотири підгрупи: лічильники профілю, атрибути профілю, engagement-метрики та характеристики каскаду поширення.

Користувач налаштовує чотири параметри моделі: варіант класифікатора (ComplementNB або MultinomialNB), коефіцієнт Лапласового згладжування α (з фіксованого набору {0.01, 0.1, 0.5, 1.0, 2.0}), тип векторизатора та діапазон n-грамів. Решта параметрів ComplementNB використовує значення за замовчуванням бібліотеки scikit-learn.

Для DistilBERT feature engineering не потрібен – модель отримує необроблений текст через стандартний токенізатор. Для GNN моделей ознаки вузлів графа формуються як MiniLM ембединги (all-MiniLM-L6-v2) через бібліотеку sentence-transformers.

2.3.3 Формування train / validation / test вибірок

Розподіл здійснюється через функцію train_test_split з обов'язковим використанням стратифікації – параметр stratify=y гарантує збереження пропорції FAKE/REAL у всіх підвибірках. Для in-domain експериментів датасет GossipCop розбивається на три вибірки: train (14 300 публікацій), validation (3 064 публікації), test (3 066 публікацій), що зберігають вихідну пропорцію класів (близько 76% REAL та 24% FAKE). Валідаційна вибірка використовується для відстеження якості моделі під час тренування та підбору гіперпараметрів, тестова – для фінального оцінювання.

Для cross-domain evaluation використовується підмножина PolitiFact (590 публікацій після очищення), що повністю виключена з тренування та слугує екзаменом на здатність моделі до узагальнення на іншому домені.

2.4 Моделі машинного навчання

Підсистема моделей класифікації становить ядро функціональності прототипу. У ній реалізовано чотири парадигми машинного навчання, теоретичні засади яких розглянуто у Розділі 1: статистичну (наївний баєсівський класифікатор, підрозділ 1.3), нейромережеву (DistilBERT, 1.4), графову (GraphSAGE та GIN, 1.5) та підхід на основі великих мовних моделей (Claude, 1.6).

2.4.1 Наївний баєсівський класифікатор

Реалізацію виконано через клас ComplementNB бібліотеки scikit-learn. Тренувальний pipeline складається з послідовних кроків: TF-IDF-векторизація на тренувальному корпусі з фіксованими параметрами max_features=50 000, sublinear_tf=True, min_df=2, max_df=0.95 та конфігурованим діапазоном n-грамів); обчислення 45 додаткових ознак (14 емоційних, 8 стилістичних та риторичних, 23 соціальні); об'єднання у єдину розріджену матрицю; тренування ComplementNB з коефіцієнтом Лапласового згладжування α, що обирається користувачем з набору {0.01, 0.1, 0.5, 1.0, 2.0}; оцінювання на тестовій вибірці. Для забезпечення інтерпретованості після тренування витягуються найбільш дискримінативні ознаки через аналіз різниці log-likelihoods між класами, що дозволяє ідентифікувати лексичні маркери, найсильніше асоційовані з кожним класом

2.4.2 Тонке налаштування DistilBERT

Реалізацію виконано через бібліотеку transformers від HuggingFace з використанням попередньо натренованої моделі distilbert-base-uncased. Поверх базової моделі встановлено класифікаційну head – лінійний шар з двох виходів (FAKE та REAL), що приймає на вхід контекстне представлення токена [CLS]. Тонке налаштування здійснюється через інтерфейс Trainer з налаштовуваними

гіперпараметрами: learning rate $2 \cdot 10^{-5}$, batch size 16, weight decay 0.01, warmup_ratio 0.1, кількість епох на вибір {1, 2, 3, 5} та максимальна довжина послідовності {128, 256, 512} (рекомендовано 3 епохи та 256 токенів для балансу між обсягом контексту та швидкістю). Додатково підтримується режим заморозки базової моделі (freeze_base), у якому тренуються лише два останні шари трансформера та класифікатор, що прискорює навчання зі збереженням якості.

2.4.3 Графові нейронні мережі (GraphSAGE та GIN)

Обидві архітектури реалізовано на базі бібліотеки PyTorch Geometric з використанням графів поширення FakeNewsNet, де вузли представляють чотири типи сутностей (статті, твіти, ретвіти, відповіді), а ребра кодують відносини поширення. Ознаки вузлів формуються як MiniLM-ембединги текстів розмірністю 384, обчислені через бібліотеку sentence-transformers з моделлю all-MiniLM-L6-v2.

Реалізація обох моделей підтримує налаштування ключових гіперпараметрів: hidden dimension {64, 128, 256, 512}, кількість шарів {2, 3, 4, 5}, тип graph-level pooling (mean/max/sum). Архітектури відрізняються механізмом агрегації сусідів: GraphSAGE підтримує налаштування типу aggregator-функції (mean / max / LSTM), тоді як GIN використовує фіксовану суматорну агрегацію згідно з оригінальною специфікацією [10] з навчуваним параметром ε (train_eps=True).

2.4.4 Класифікація через велику мовну модель Claude

Інтеграція з моделями сімейства Claude реалізована через CLI-інструмент Claude Code з викликом через subprocess.run, що дозволяє використовувати OAuth-підписку без API-токенів. У системі підтримуються два режими взаємодії: zero-shot – модель отримує лише системний промпт з описом задачі та текст для класифікації (системний промпт спеціалізує модель як експертного аналітика медіаграмотності та вимагає повернення JSON-відповіді з полями label, confidence та reason); few-

shot – промпт доповнюється кількома прикладами правильної класифікації (з налаштуванням кількості FAKE та REAL прикладів окремо.

Для масштабованого оцінювання на повних тестових вибірках реалізовано batch-режим, що об'єднує до десяти-дванадцяти текстів у єдиний промпт та повертає JSON-масив відповідей, що знижує загальну кількість API-викликів на порядок. Конфігурація конкретного виклику задається через LLM-пресет, що зберігається у базі даних як запис ModelRecord з типом llm та JSON-полем llm_config, який містить вибір базової моделі (Claude Haiku, Sonnet, Opus), режим взаємодії, системний промпт, температуру та перелік few-shot прикладів.

2.5 Користувацький інтерфейс

Клієнтська частина системи являє собою односторінковий веб-застосунок, реалізований на React з TypeScript. Інтерфейс складається з дев’яти функціональних сторінок, доступних через бічну навігаційну панель. Кожна сторінка відповідає окремому use case системи та забезпечує інтуїтивну взаємодію з відповідною функціональністю.

2.5.1 Огляд сторінок інтерфейсу

Сторінка "Датасети" відображає список завантажених датасетів у табличному форматі (рис. 2.4). Таблиця містить колонки з назвою датасету, кількістю новин, розподілом міток FAKE/REAL, кількістю експериментів, доступними splits (завантажених користувачем) та статусом активації. Для кожного датасету доступні дії через контекстне меню: перегляд детальної статистики, активація для використання у тренуванні та видалення. У правому верхньому куті розміщено кнопку "Завантажити датасет" для додавання нових наборів даних. Детальна статистика датасету (рис. 2.5) відображається у модальному вікні та

містить загальну кількість статей, розподіл за мітками, кількість твітів та користувачів, а також додаткові характеристики поширення у соціальних мережах.

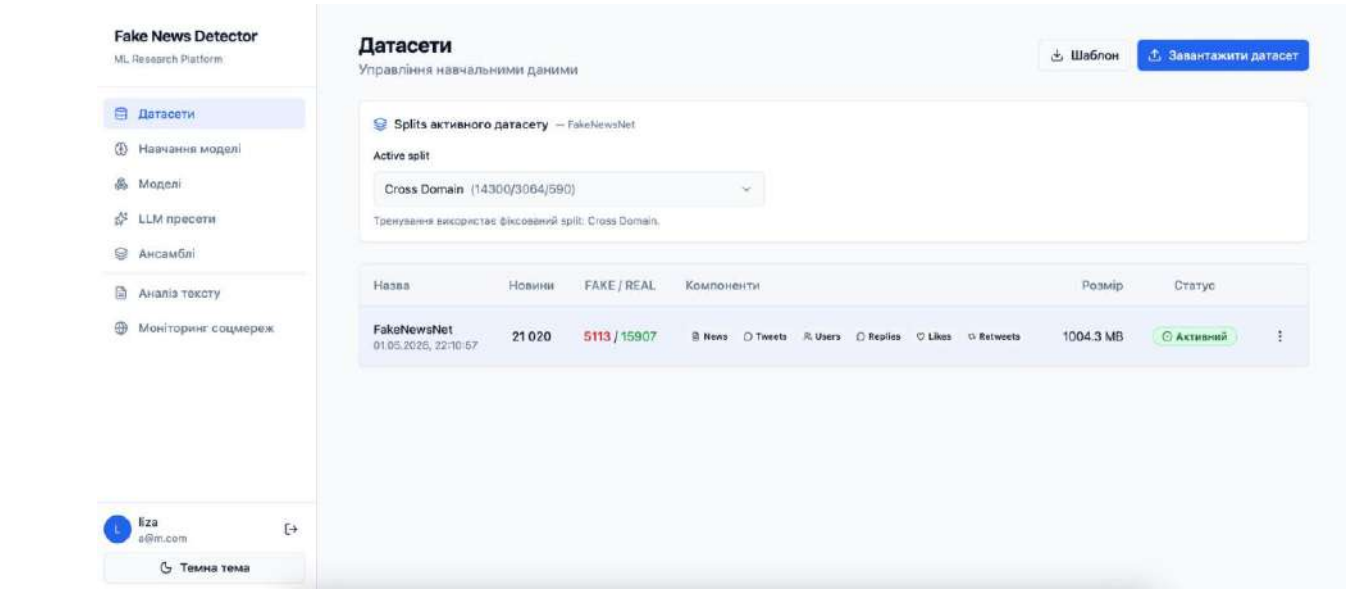

Рисунок 2.3. *Сторінка управління датасетами*

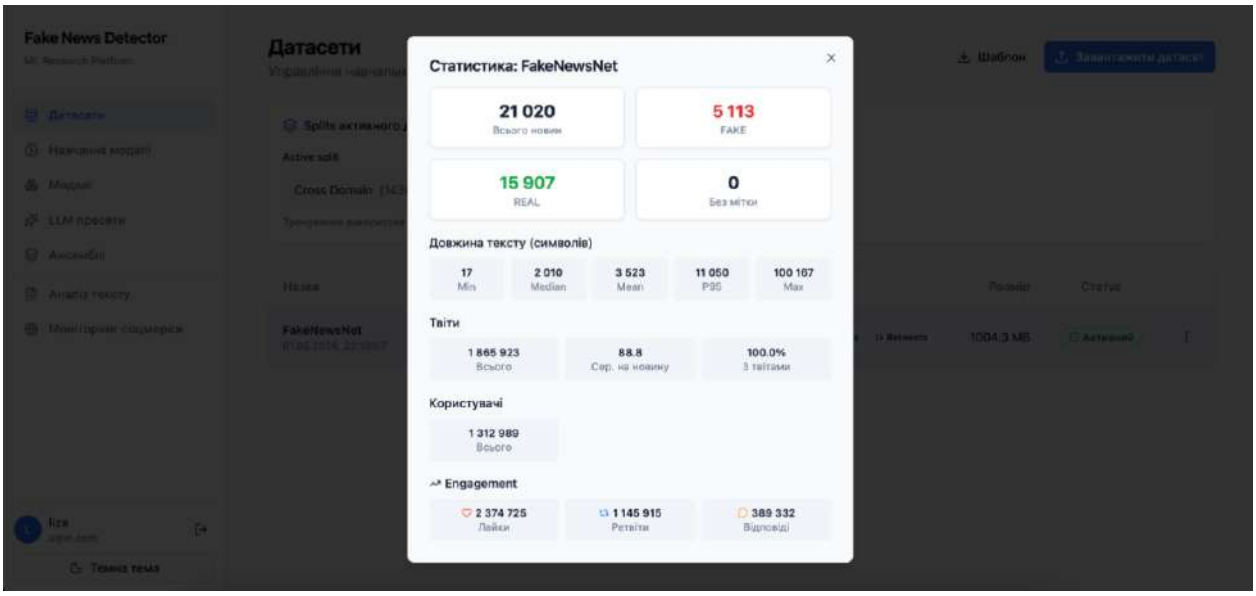

Рисунок 2.4. *Статистика датасету*

Сторінка "Навчання моделі" реалізує wizard-інтерфейс конфігурації параметрів тренування з валідацією на стороні клієнта. На першому етапі користувач обирає тип моделі: Naive Bayes, DistilBERT, GNN (GraphSAGE або GIN) (рис. 2.6). На другому етапі відбувається конфігурація гіперпараметрів специфічних для обраної моделі та вибір додаткових ознак (рис. 2.7). Для Naive Bayes доступні три групи ознак: емоційні (14 ознак, включаючи sentiment score, emotion intensity, показники гніву, страху, радості), стилістичні (8 ознак,

включаючи caps ratio, TTR, clickbait score, authority references) (рис. 2.8) та соціальні (23 ознаки профілю користувача, engagement metrics, характеристики каскаду поширення) (рис. 2.9). Для DistilBERT налаштовуються epochs, max length, freeze base та integration mode. Після відправки форми сторінка переходить у режим відстеження прогресу з періодичним опитуванням endpoint /training/status та оновленням індикатора прогресу. По завершенню тренування відображаються метрики натренованої моделі (accuracy, precision, recall, F1), матриця помилок та хмара найбільш дискримінативних слів для кожного класу (рис. 2.10).

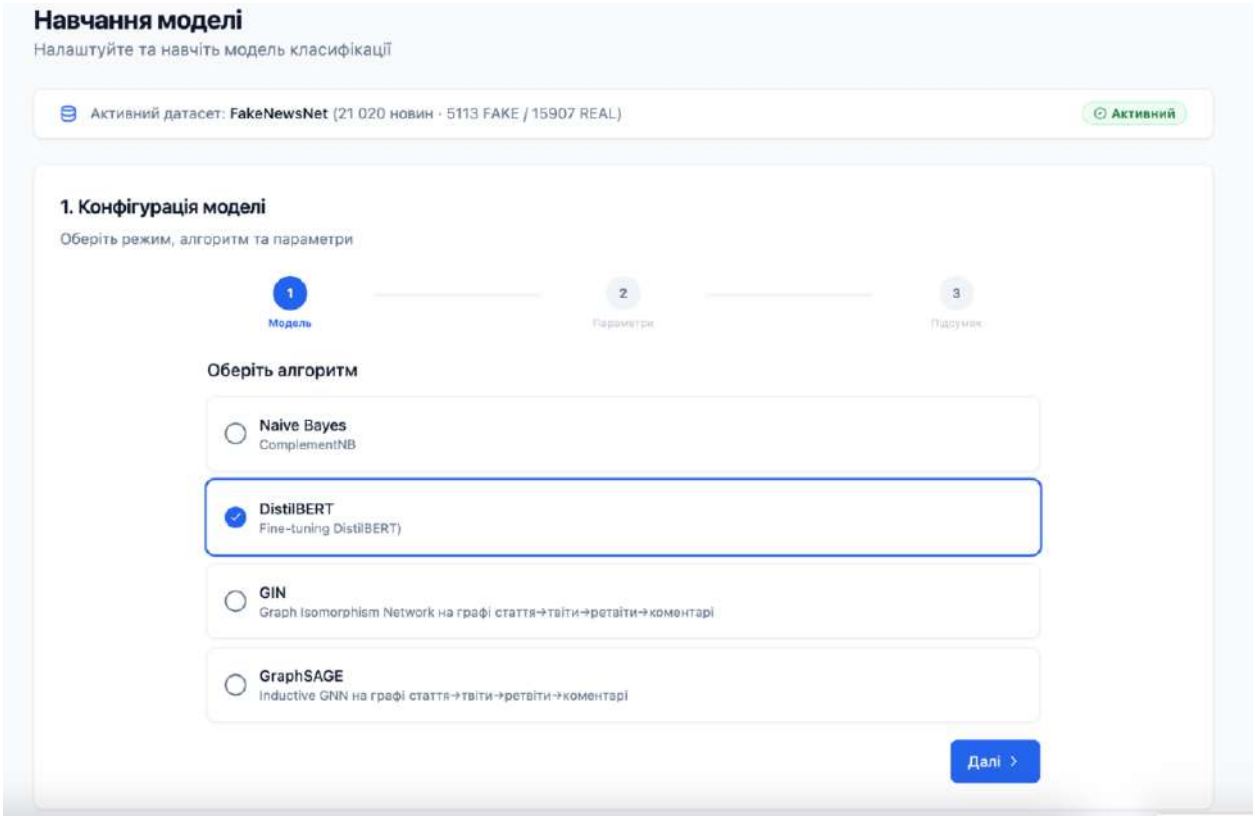

Рисунок 2.5. *Перший етап: Вибір алгоритму*

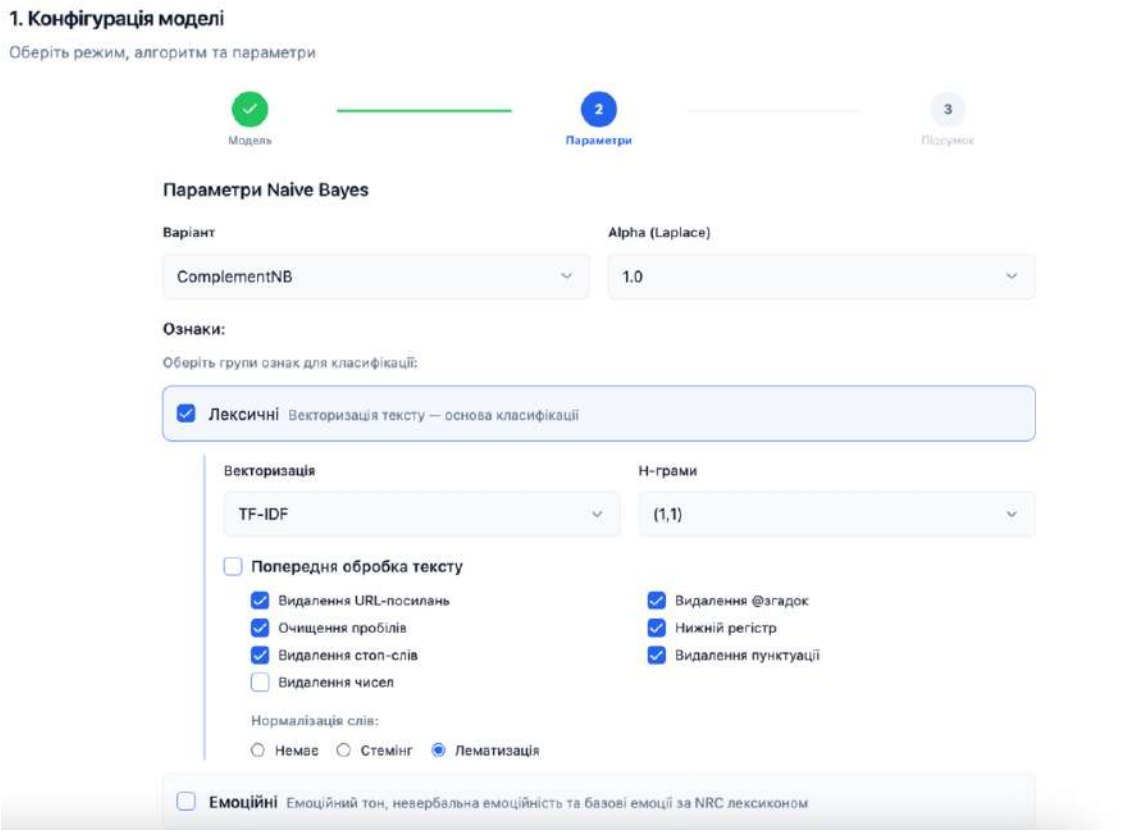

Рисунок 2.6. *Другий етап: налаштування параметрів і вибір ознак для наївного баєса*

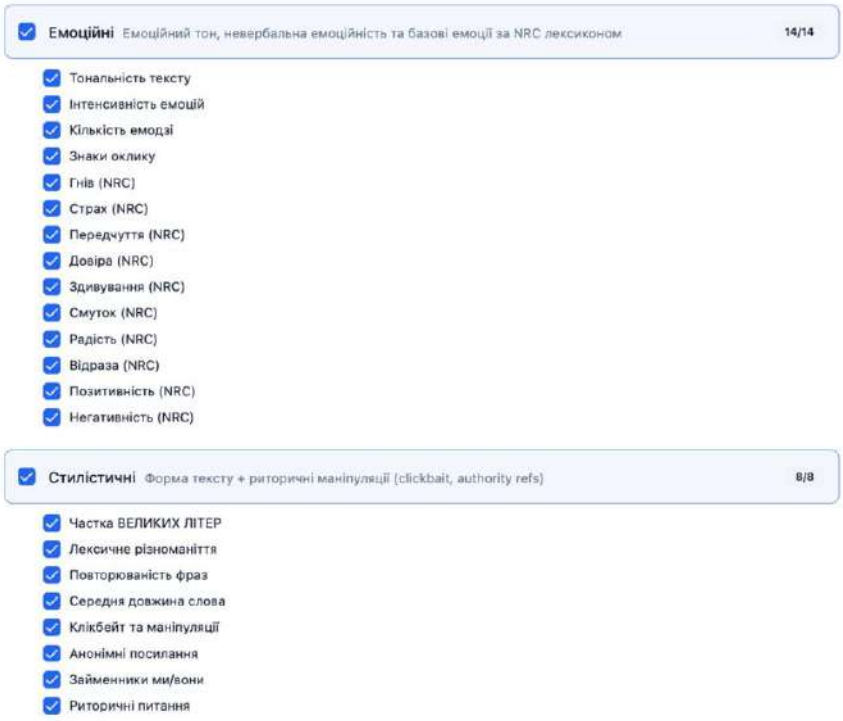

Рисунок 2.7. *Емоційні та стилістичні ознаки для наївного баєса*

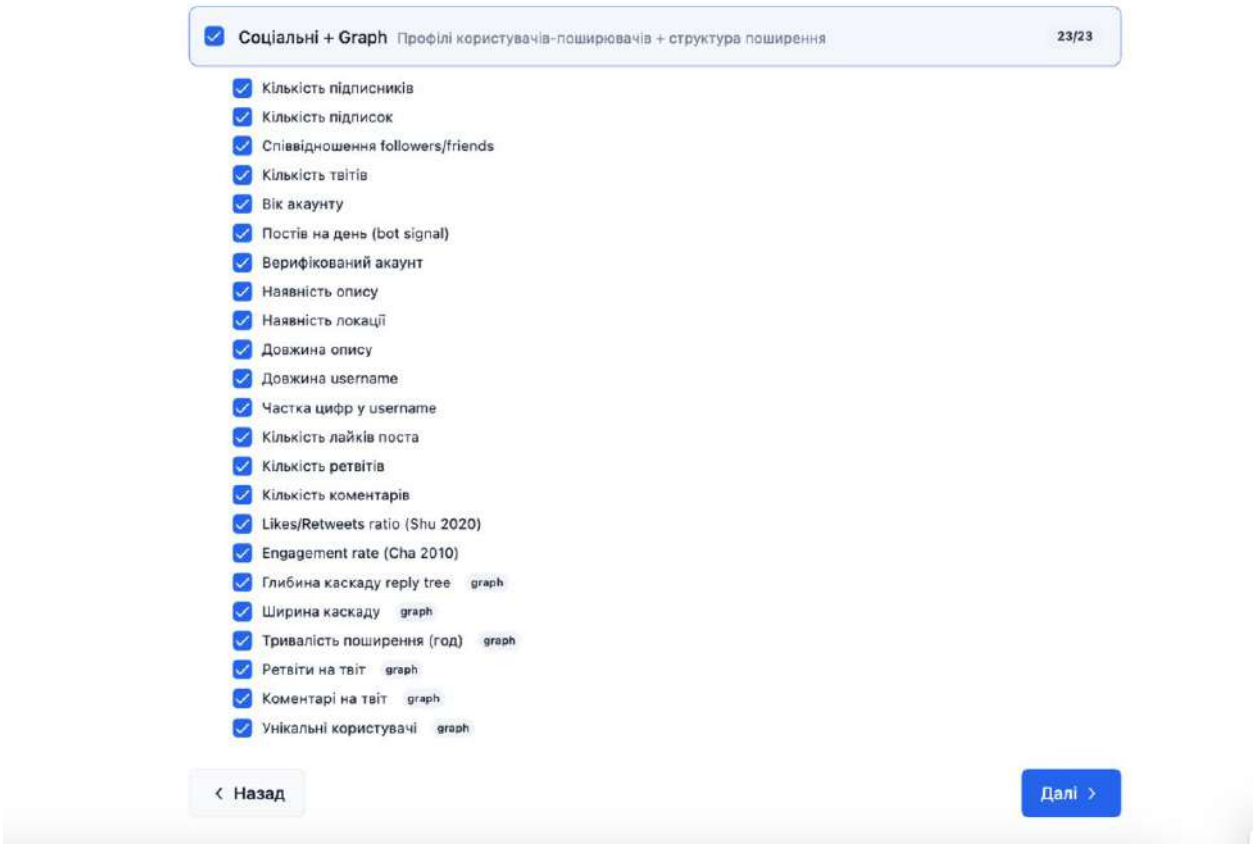

Рисунок 2.8. *Соціальні ознаки для наївного баєса*

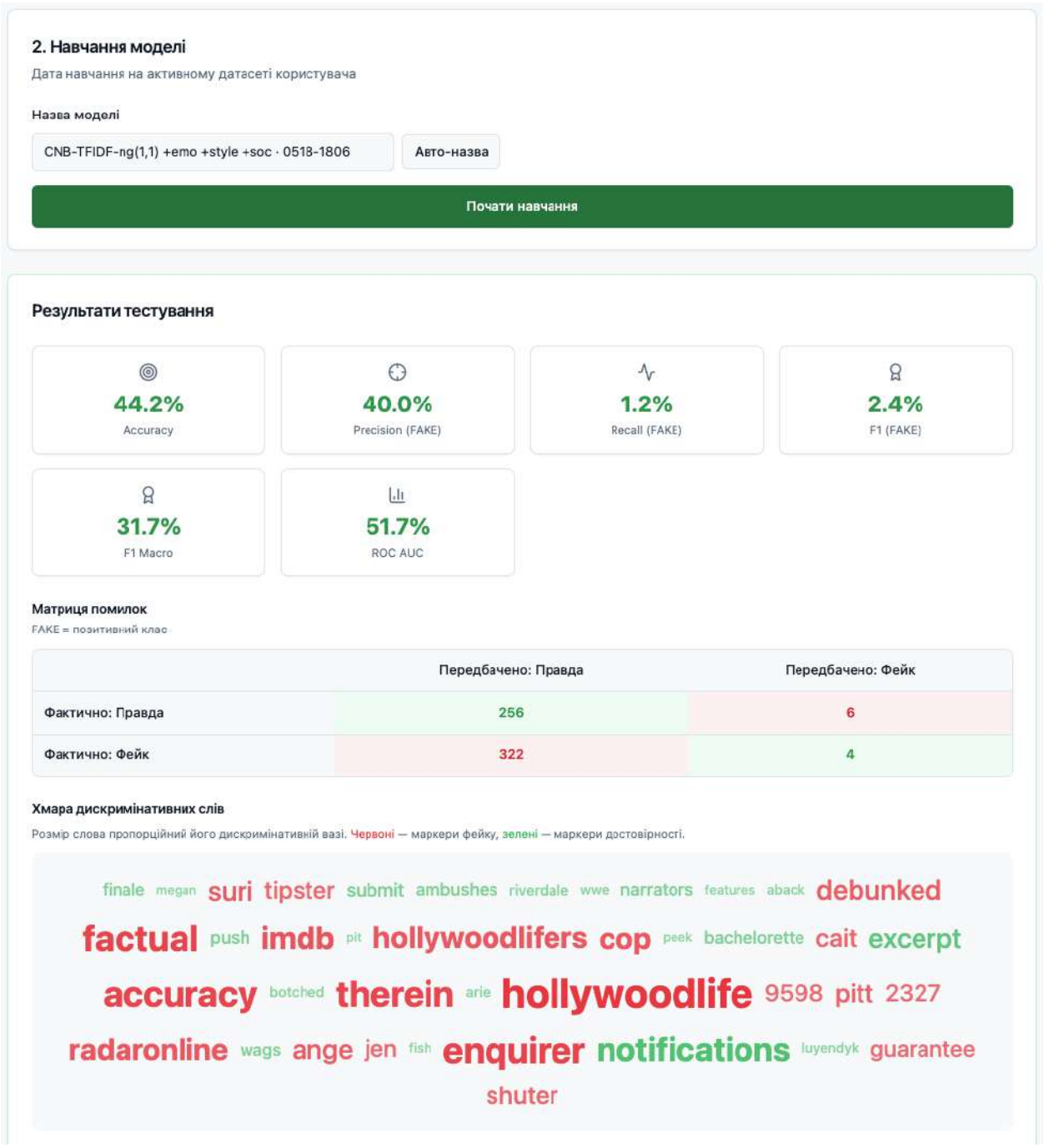

Рисунок 2.9. *Завершене навчання наївного баєса*

Сторінка “Моделі” містить список натренованих моделей у вигляді карток з ключовими метриками (рис. 2.11): тип моделі, accuracy, F1 score, precision, recall на класі FAKE, ROC-AUC, час тренування та дата створення.

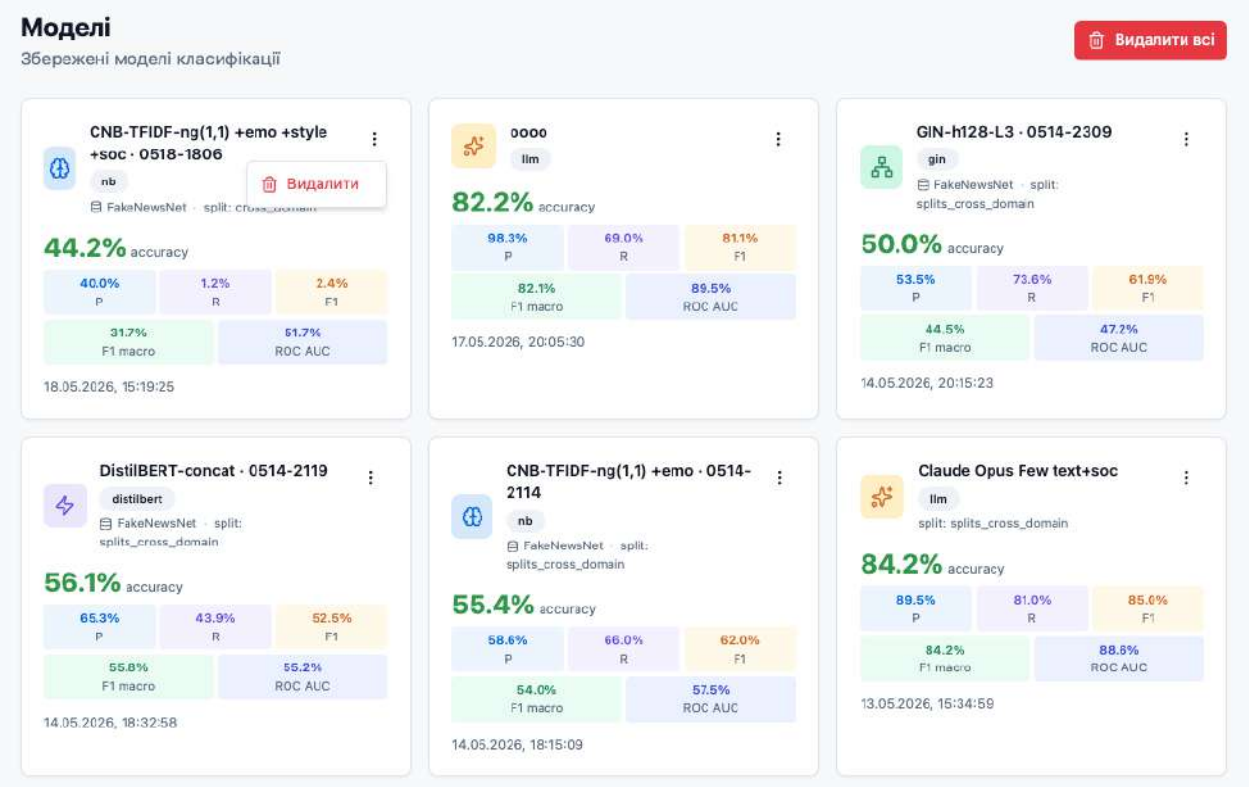

Рисунок 2.10. *Список моделей користувача*

Сторінка “LLM пресети” дозволяє створювати, зберігати та тестувати конфігурації для великих мовних моделей Claude. Список збережених пресетів (рис. 2.12) відображається у табличному форматі з колонками: назва пресету, базова модель (Claude Haiku/Sonnet/Opus), режим роботи (zero-shot, few-shot, chain-of-thought, bagging), accuracy на тестовому split, температура sampling та дата створення. Для кожного пресету доступні дії: редагування конфігурації, копіювання для створення варіанту та видалення.

Форма створення пресету (рис. 2.13) містить поля для базових параметрів: назва пресету, вибір базової моделі з випадного списку (Claude Haiku, Sonnet, Opus), температура sampling (0-1), максимальна кількість токенів відповіді. Центральним елементом є поле system prompt – текстова інструкція для моделі що визначає її поведінку при класифікації. Нижче розміщені чотири режими роботи у вигляді вкладок: zero-shot (класифікація без прикладів на основі лише system prompt), few-shot (з додаванням прикладів класифікованих текстів), chain-of-thought (з інструкцією міркувати покроково) та bagging (множинні незалежні класифікації з усередненням). Додатково доступна опція включення social context для

збагачення промпту соціальним контекстом публікації: характеристики профілю автора (кількість фоловерів, статус верифікації, вік акаунту), метрики engagement (лайки, ретвіти, відповіді), та характеристики каскаду поширення (глибина, ширина розповсюдження). У нижній частині розміщено поле для введення тестового тексту, що дозволяє швидко перевірити роботу пресету перед збереженням.

Після тестування пресету система відображає детальні метрики його якості (рис. 2.14): accuracy, precision та recall, F1-macro та F1 для класу FAKE, а також матрицю помилок у табличному форматі з кількістю правильних та хибних класифікацій для кожної комбінації фактичних та передбачених міток. Це дозволяє оцінити ефективність налаштувань перед використанням пресету для масової класифікації або збереженням у бібліотеку.

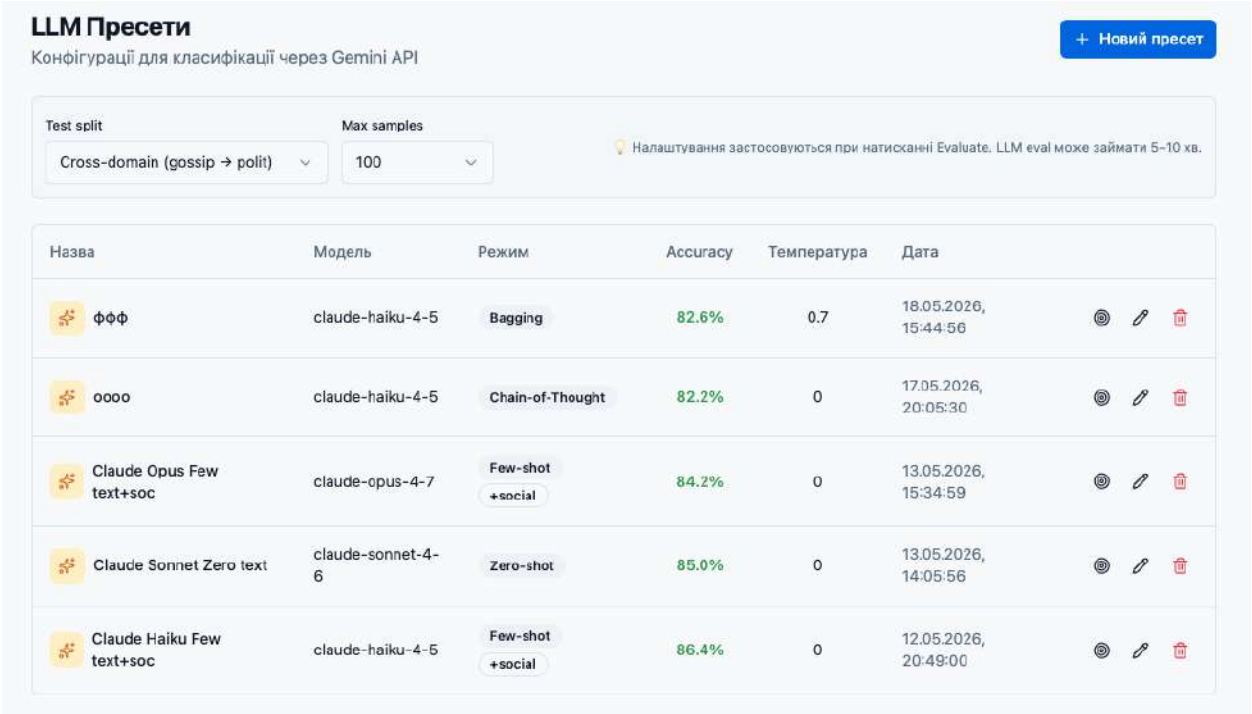

Рисунок 2.11. *Список LLM пресетів користувача*

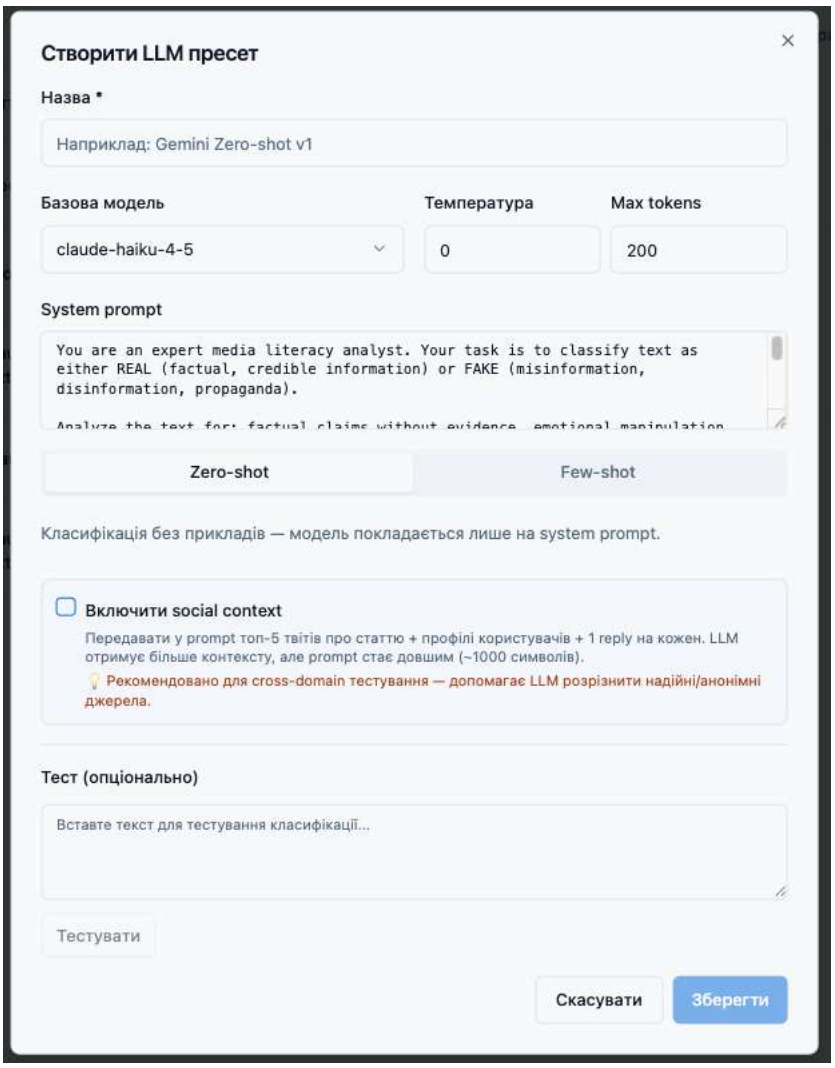

Рисунок 2.12. *Налаштування та створення LLM пресету*

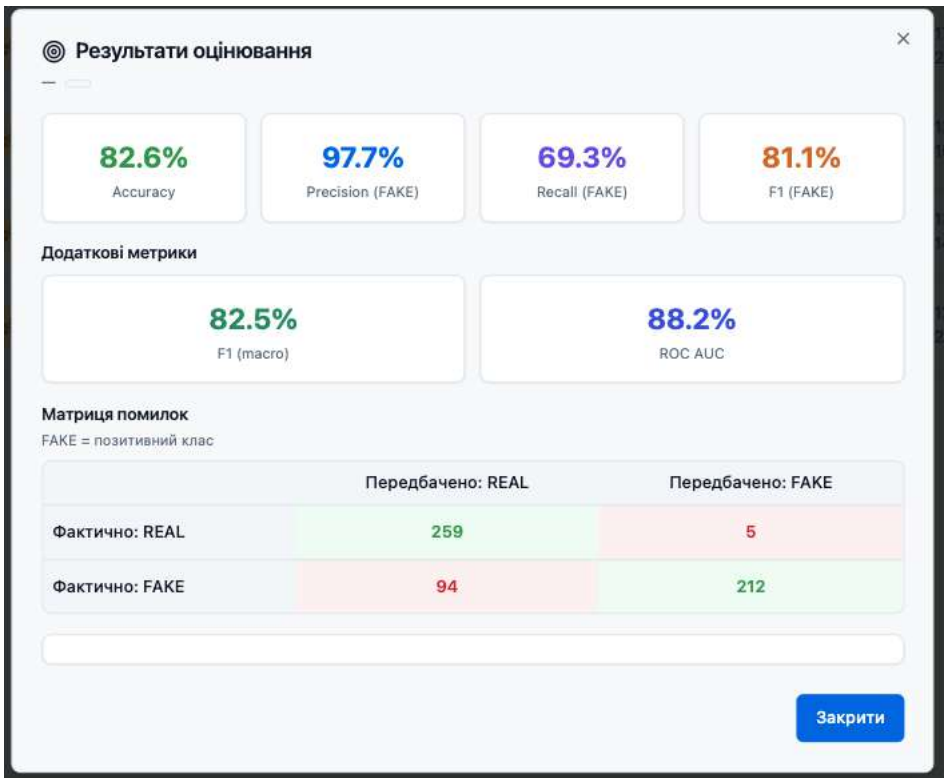

Рисунок 2.13. *Результат тестування LLM пресету*

Сторінка "Ансамблі" надає інтерфейс створення ансамблевих моделей з комбінації натренованих моделей. Список створених ансамблів відображається у вигляді карток з ключовими метриками: назва ансамблю, кількість моделей-членів, стратегія голосування, accuracy та F1-score (рис. 2.15). Детальна сторінка ансамблю (рис. 2.16) містить повні метрики (accuracy, precision, recall, F1), матрицю помилок та список моделей-членів з їх індивідуальними F1-scores. Створення нового ансамблю відбувається через wizard-інтерфейс: на першому етапі користувач обирає від 2 моделей зі списку доступних натренованих моделей (рис. 2.17), на другому етапі вибирає стратегію об'єднання прогнозів – hard voting (більшість голосів), soft voting (середнє ймовірностей FAKE) або weighted voting (зважена сума з можливістю налаштування ваг) (рис. 2.18). На фінальному етапі відображається підсумок конфігурації з переліком обраних моделей та параметрів голосування (рис. 19), після чого система створює ансамбль, виконує його

оцінювання на тестовій вибірці та зберігає результати для подальшого використання у класифікації.

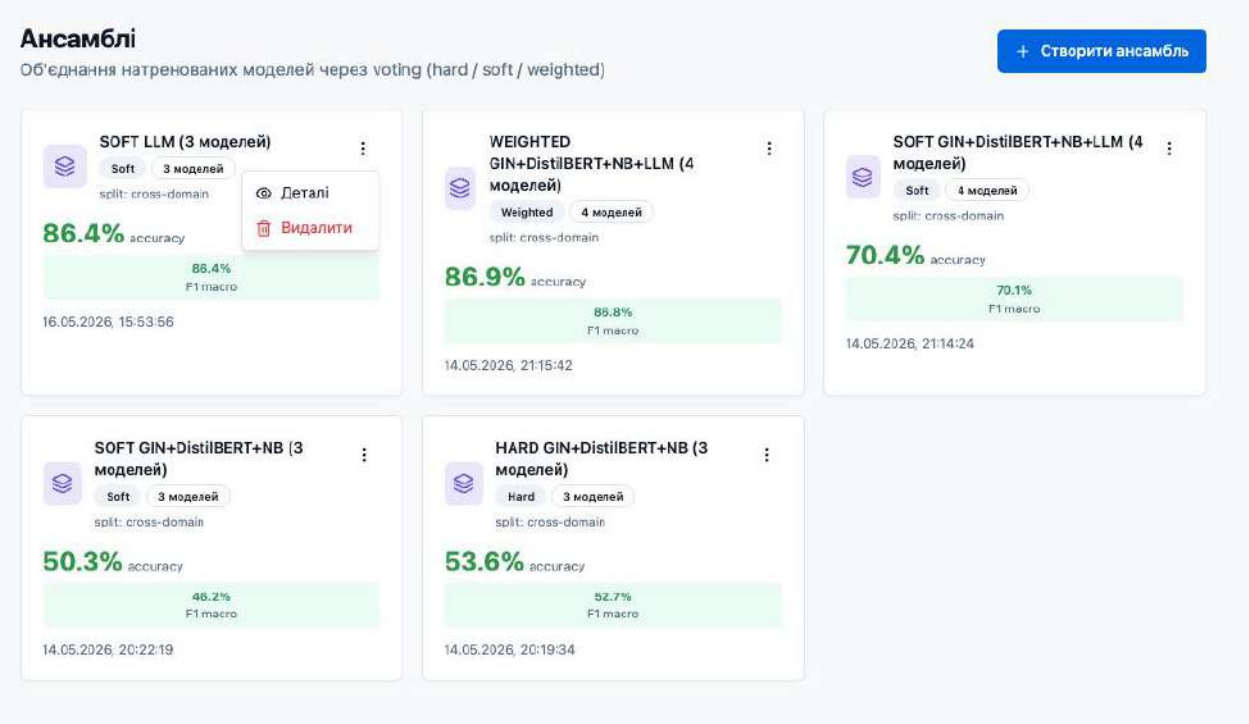

Рисунок 2.14. *Список ансамблів користувача*

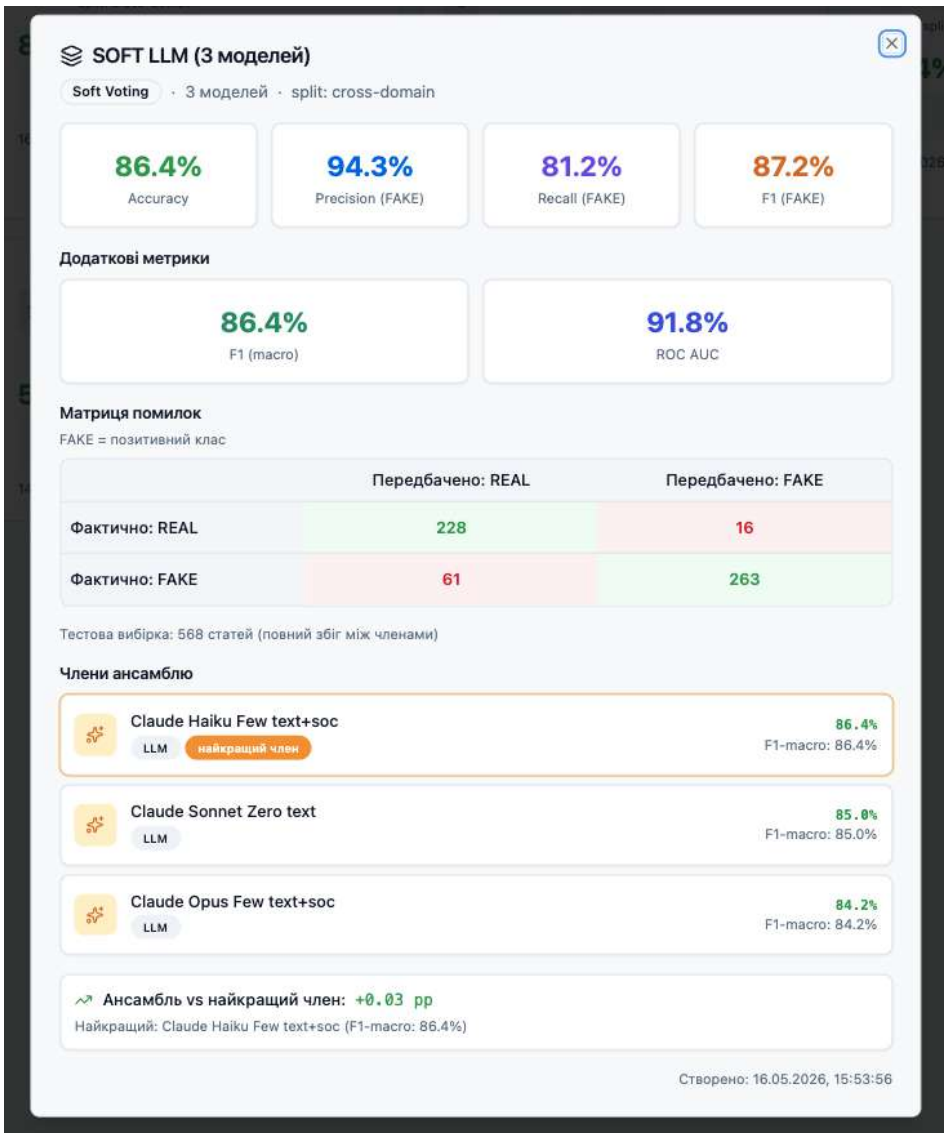

Рисунок 2.15. *Деталі ансамблю*

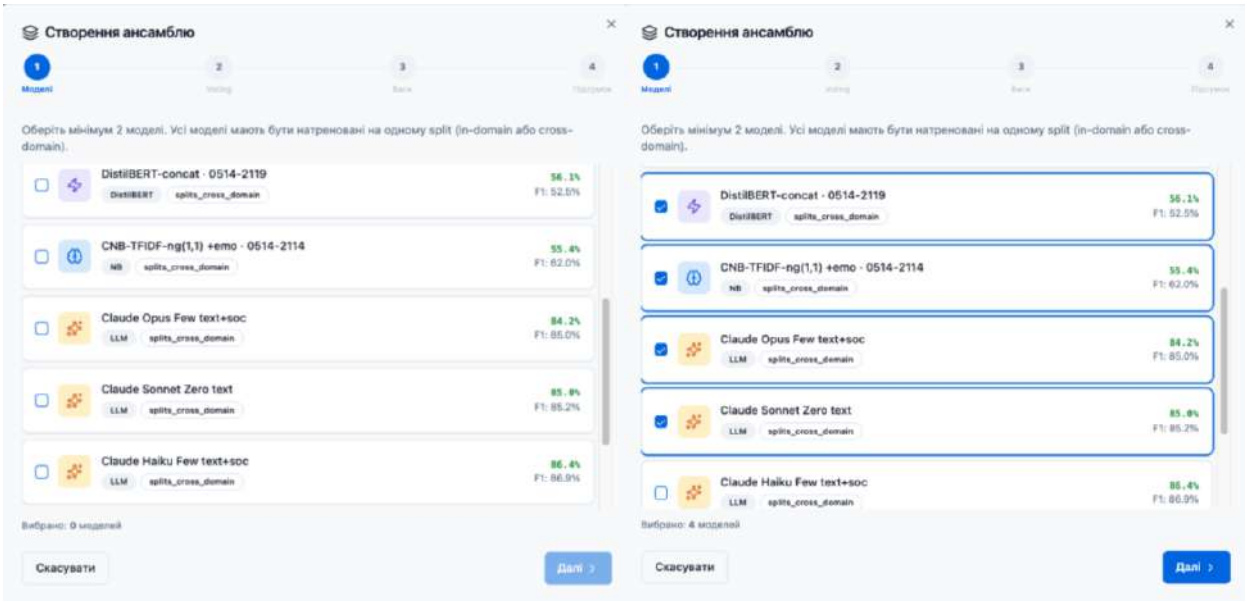

Рисунок 2.16. *Перший етап: вибір моделей для ансамблю*

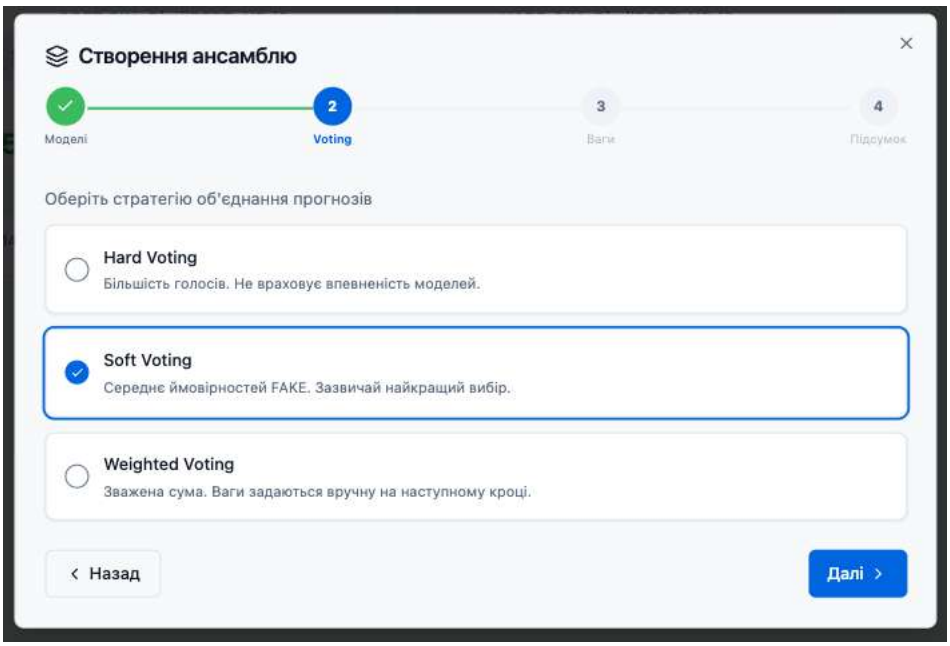

***Рисунок 2.17**. Другий етап: вибір методу голосування*

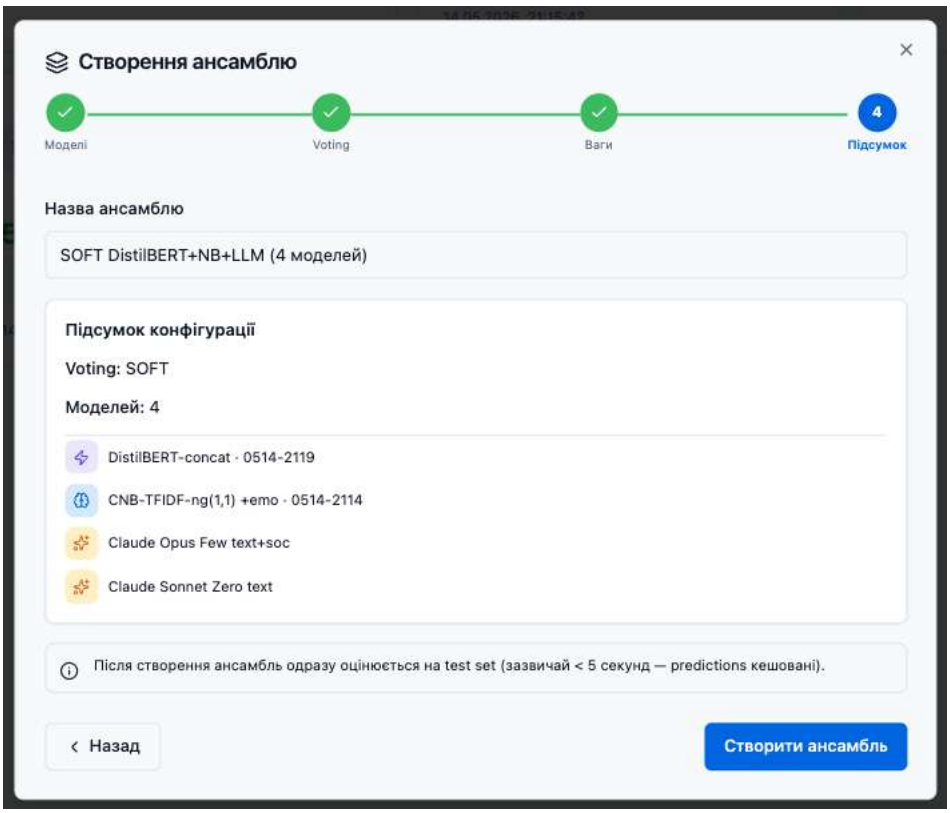

***Рисунок 2.18**. Підсумок конфігурації ансамблю*

Сторінка “Аналіз тексту” є основним інтерактивним інструментом для перевірки конкретних текстів та публікацій з соціальних мереж. Інтерфейс організовано у вигляді трьох вкладок для різних режимів аналізу: "Текст" для прямого введення тексту, "За URL" для витягування контенту з соціальних мереж, "Поширення" для аналізу тексту з урахуванням соціального контексту.

У режимі "Текст" (рис. 2.20) користувач вводить текст у поле вводу (до 10 000 символів), обирає модель або ансамбль з випадного списку та активує опціональні компоненти: витягування тверджень через LLM, класифікацію контенту моделлю та верифікацію через Google Fact Check API. Результат (рис. 2.21, 2.23) відображається у вигляді послідовних блоків: витягнуті твердження з stance detection (стверджує/заперечує), класифікація з міткою FAKE/REAL, показником впевненості та текстовим поясненням від моделі. Для Naive Bayes додатково відображається графік найвпливовіших слів та список з їх внеском у рішення (рис. 2.26). Блок fact-check показує результати верифікації з описом рейтингу фактчекера та посиланням на джерело.

У режимі "За URL" (рис. 2.24) система витягує контент з публікацій Bluesky або Mastodon за вказаним посиланням. Інтерфейс відображає метадані публікації (автор, дата, engagement metrics) та виконує ті самі кроки аналізу що й у режимі "Текст" (рис. 2.25, 2.26).

Режим "Поширення" (рис. 2.27) призначений для аналізу тексту з урахуванням їх соціального контексту. Користувач вказує текст, обирає соціальні мережі для пошуку (Mastodon, Bluesky) та задає максимальну кількість постів для аналізу. Система виконує масову класифікацію знайдених публікацій обраною моделлю та відображає результати у вигляді таблиці з окремими predictions для кожного поста (рис. 2.28). Додатково візуалізується розподіл тверджень за stance (позицією) (підтримка/заперечення/нейтральне) та загальний консенсус верифікації через fact-checking API (рис. 2.29), що дозволяє оцінити загальний характер дискусії навколо публікації у соціальних мережах.

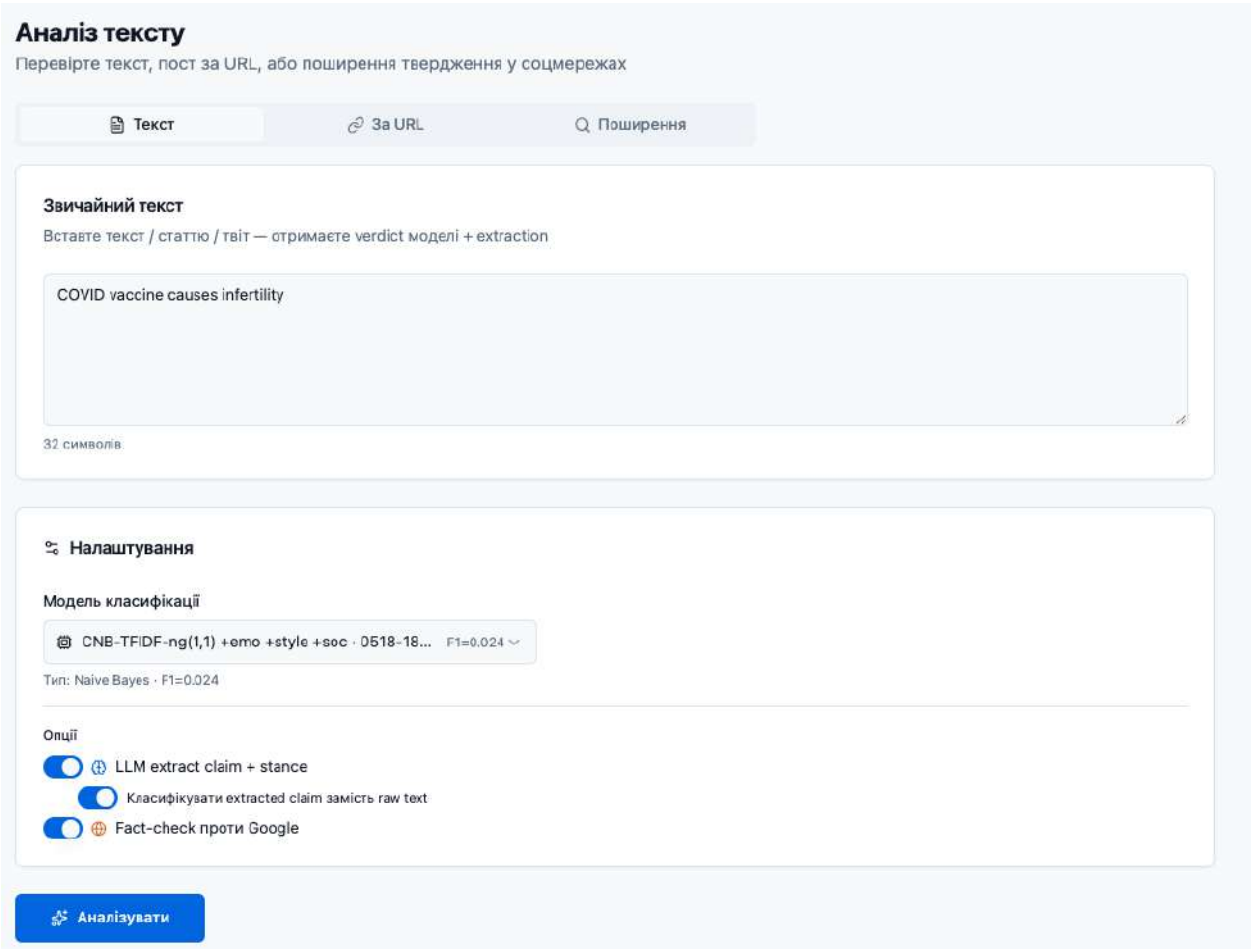

Рисунок 2.19. *Аналіз тексту та вибір параметрів (DistilBERT) на секції «Текст»*

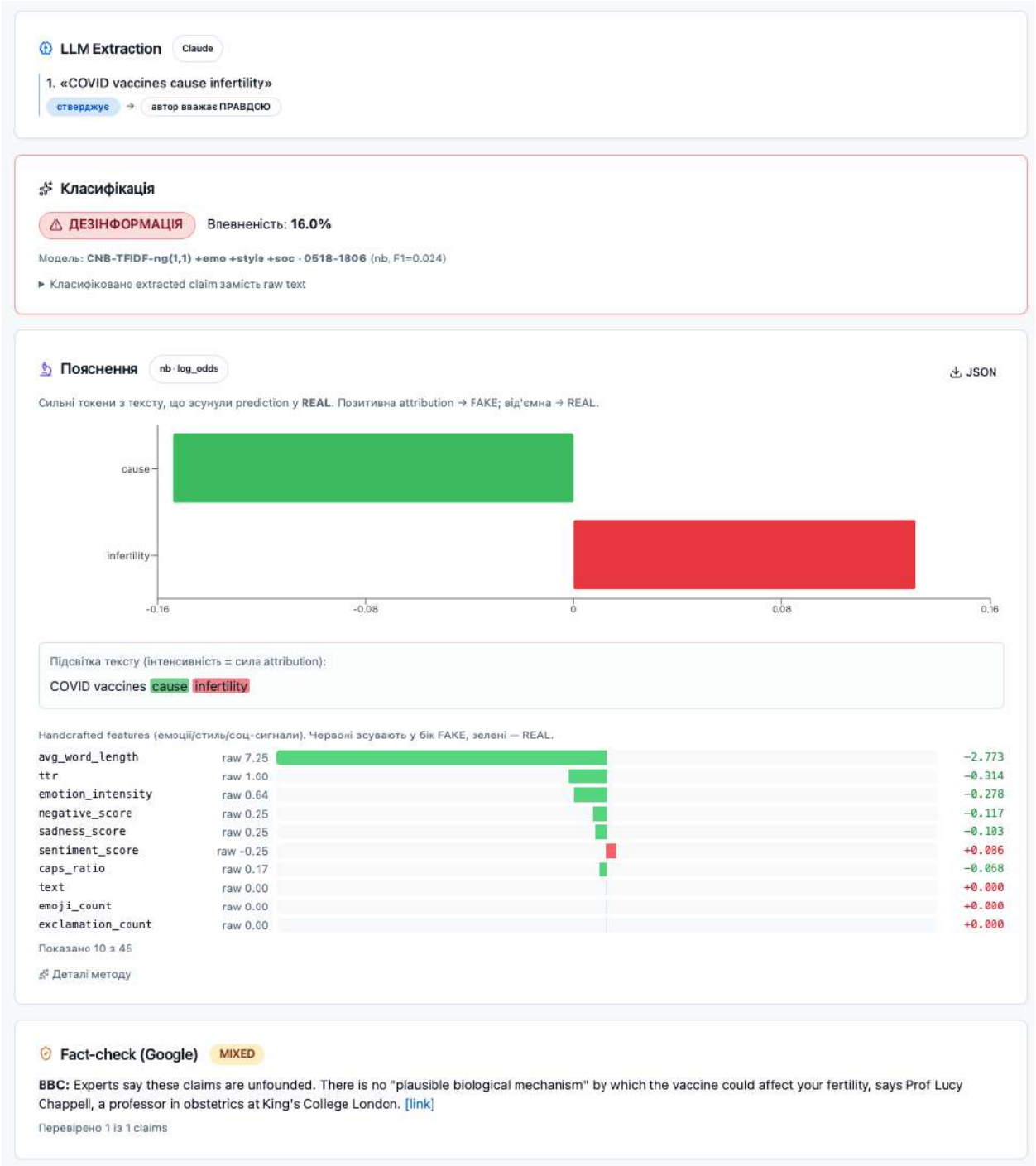

Рисунок 2.20. *Результат аналізу (DistilBERT) на секції «Текст»*

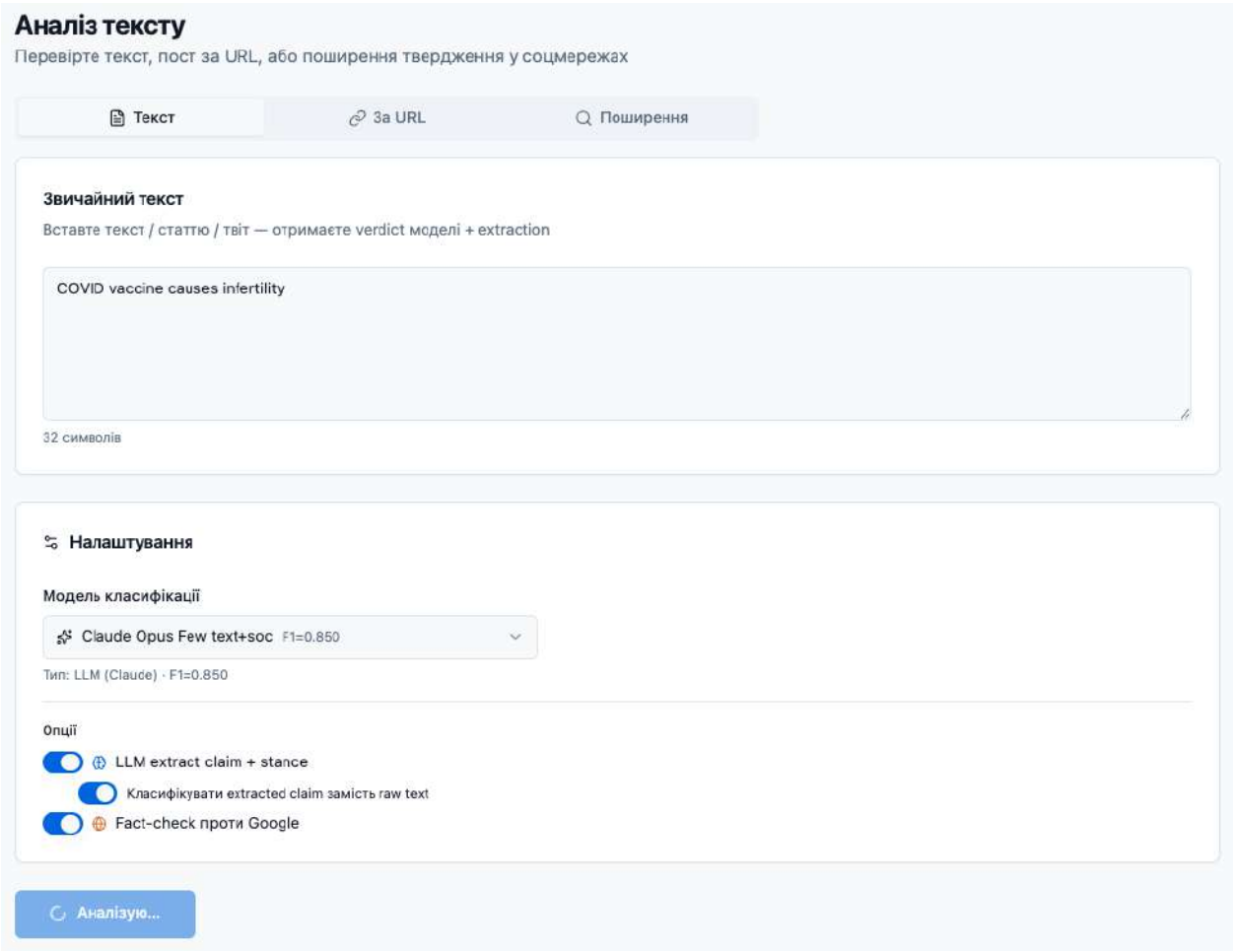

Рисунок 2.21. *Аналіз тексту та вибір параметрів (Claude Opus) на секції «Текст»*

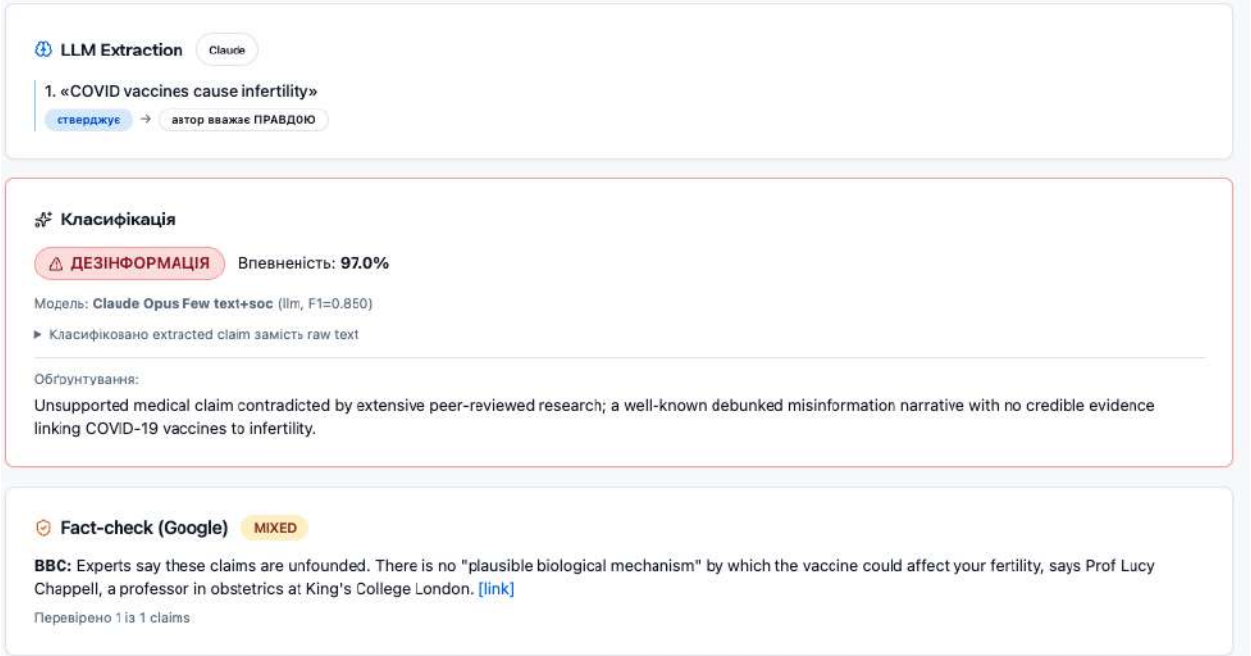

Рисунок 2.22. *Результат аналізу (Claude Opus) на секції «Текст»*

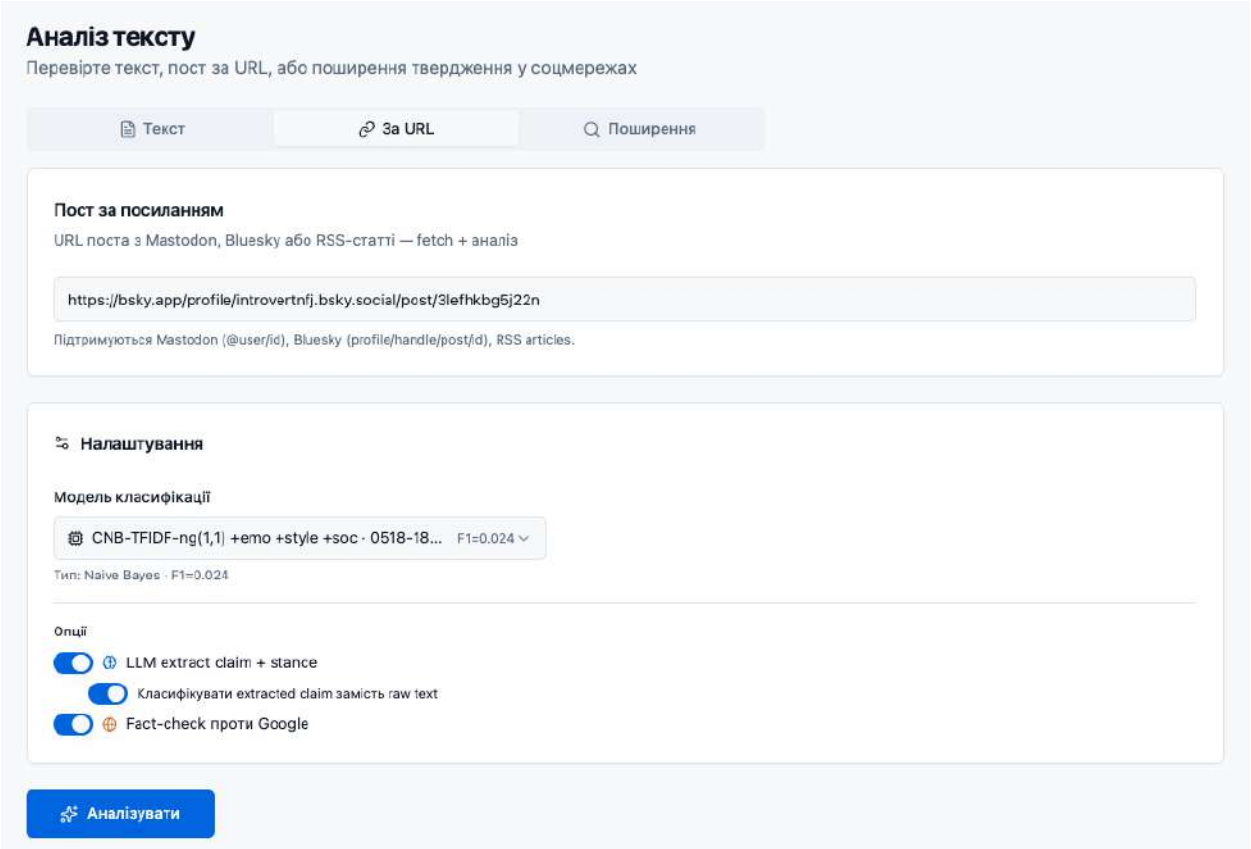

Рисунок 2.23. *Аналіз тексту та вибір параметрів (наївний баєс) на секції «За URL»*

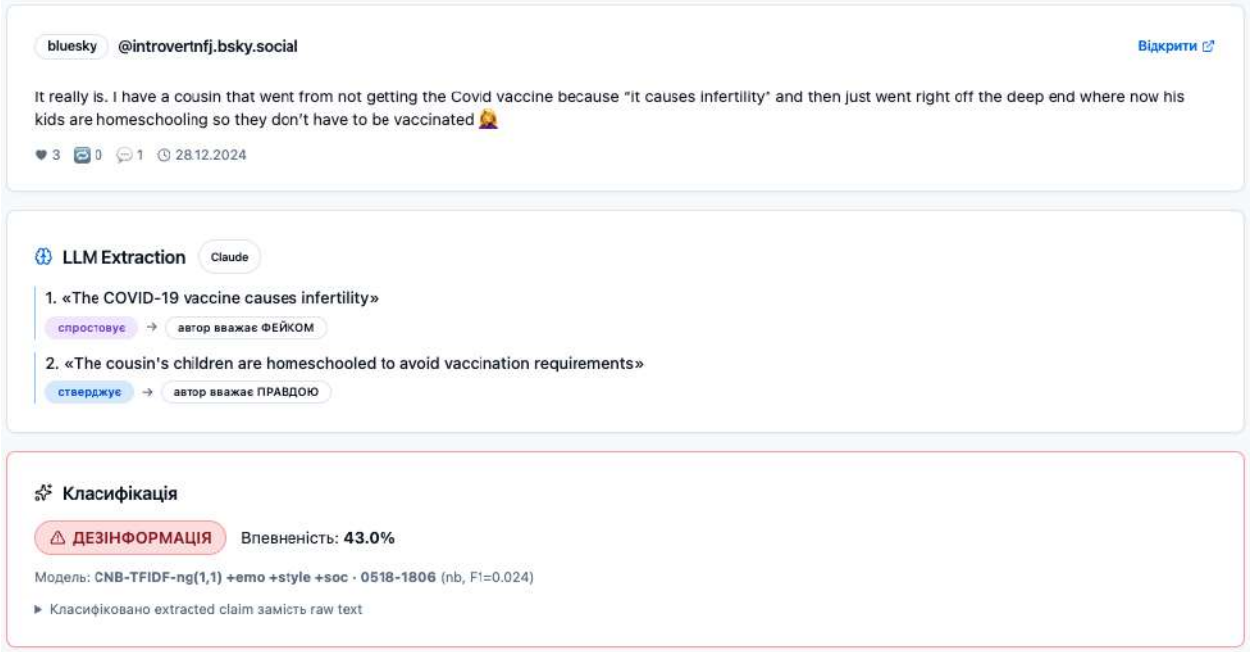

Рисунок 2.24*. Результат аналізу (наївний баєс) на секції «За URL»*

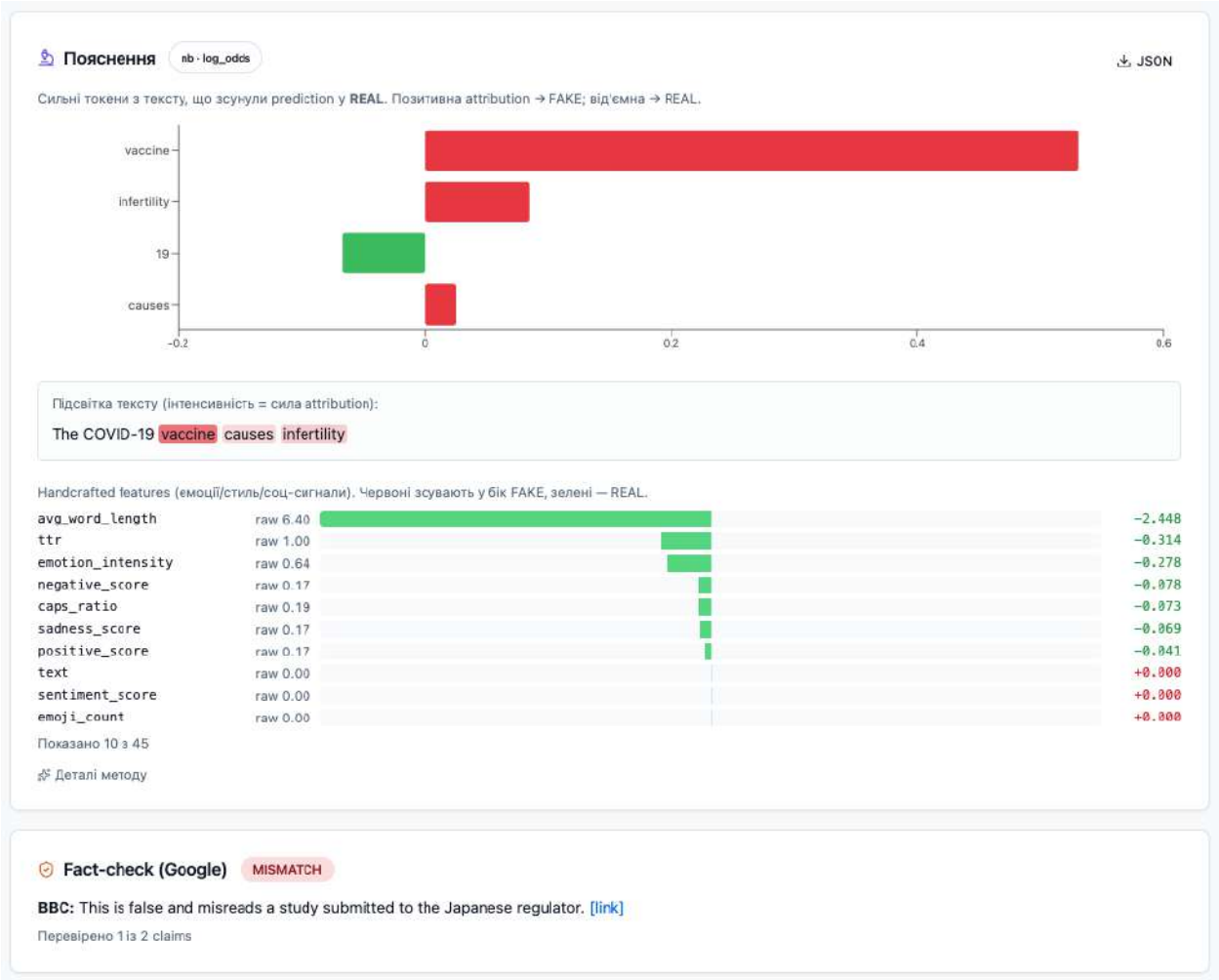

Рисунок 2.25. *Результат аналізу (наївний баєс) на секції «За URL»*

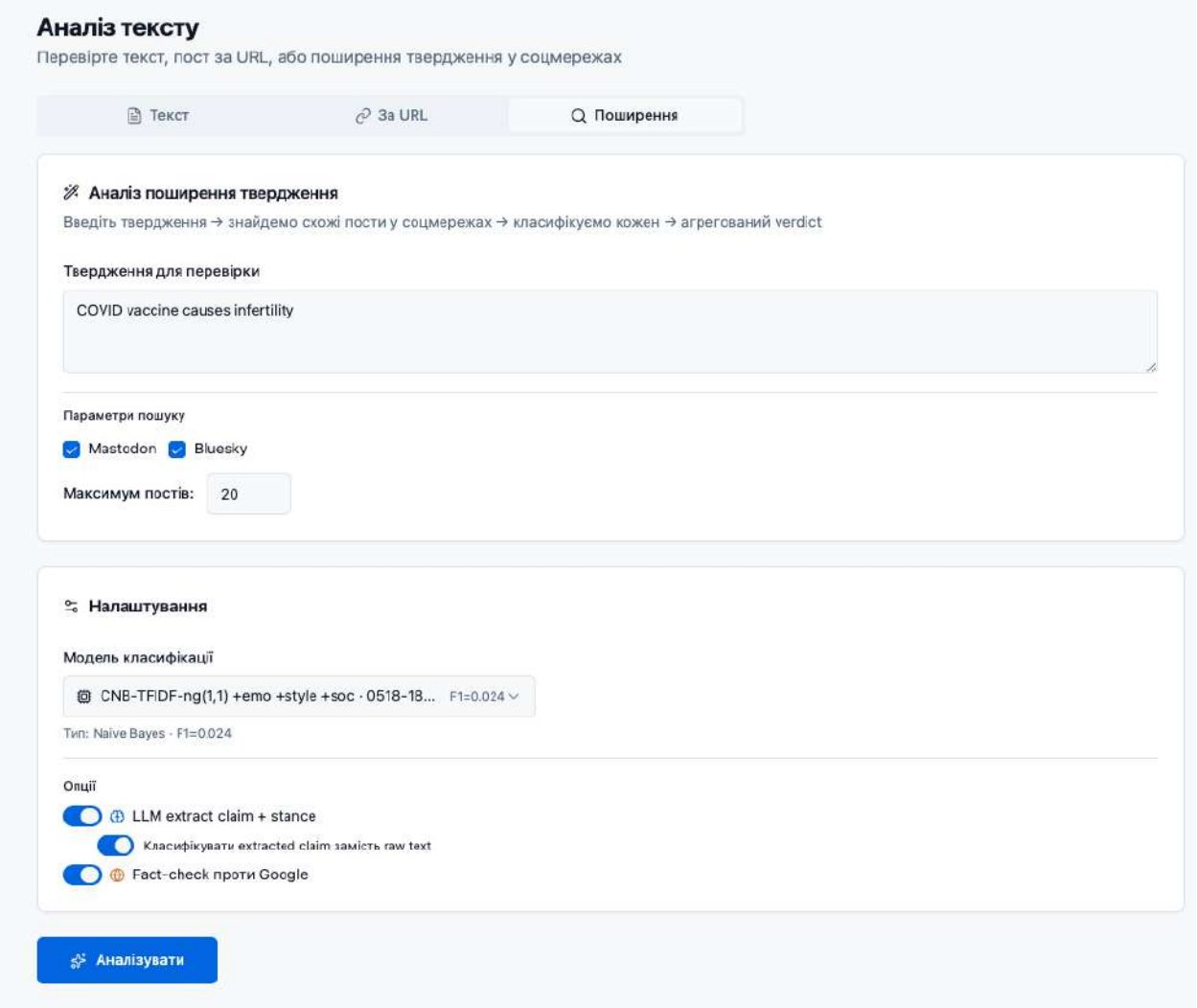

Рисунок 2.26. *Аналіз тексту та вибір параметрів (наївний баєс) на секції «Поширення»*

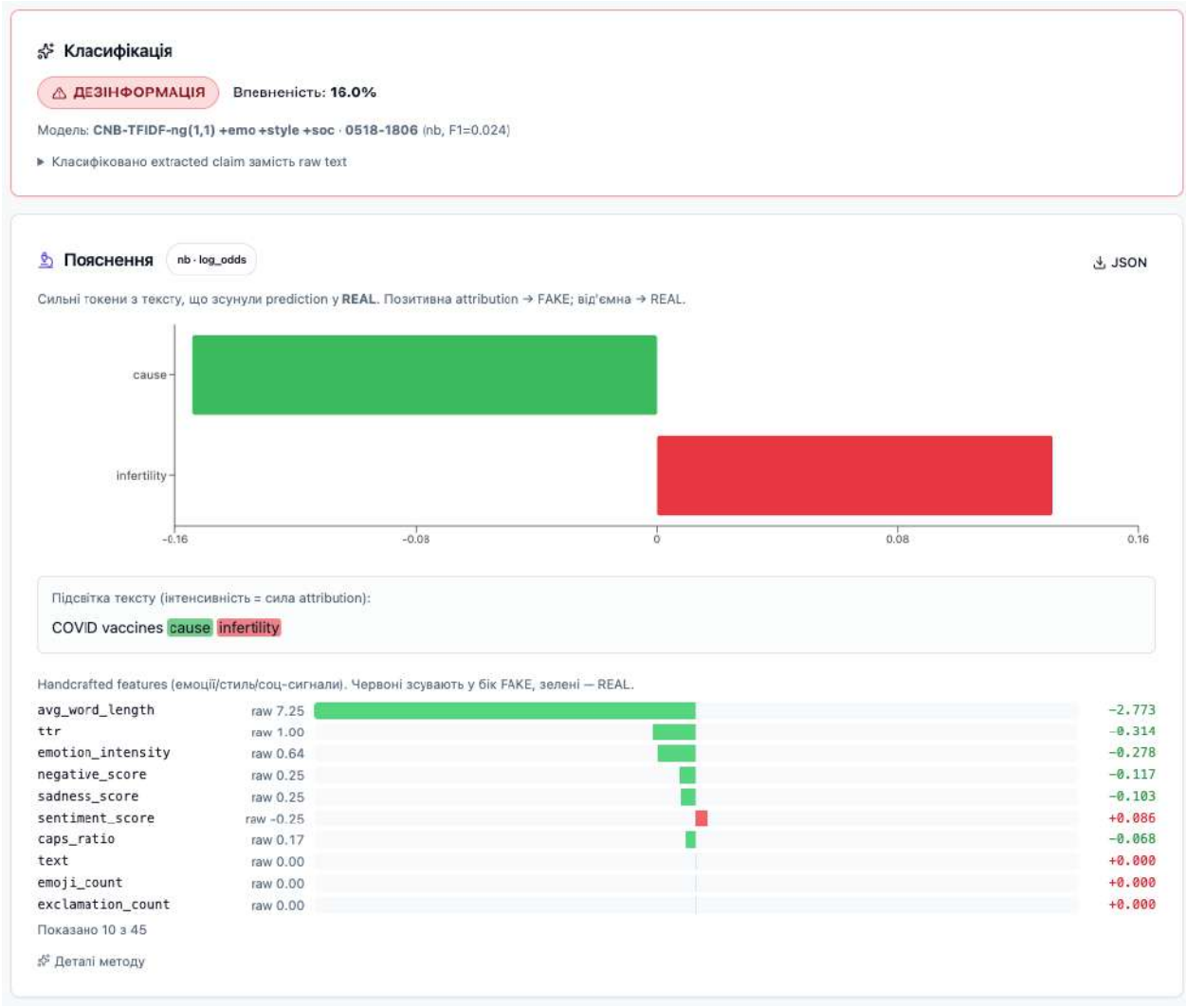

Рисунок 2.27. *Результат аналізу (наївний баєс) на секції «Поширення»*

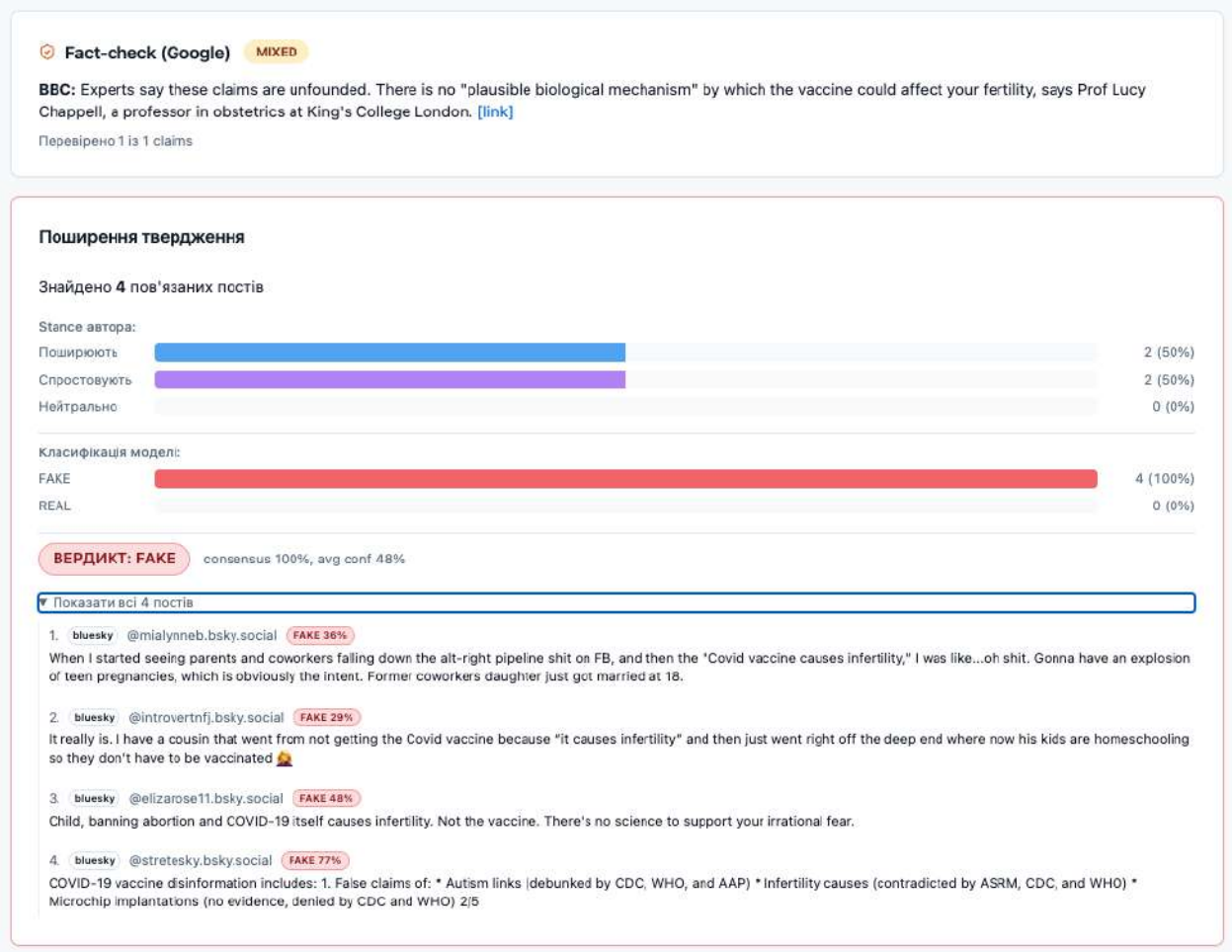

Рисунок 2.28. *Результат аналізу (наївний баєс) на секції «Поширення»*

Сторінка “Реальні дані” надає інтерфейс для аналізу публічних публікацій з соціальних мереж. Форма пошуку (рис. 2.30) містить вибір джерел даних (Bluesky, Mastodon з можливістю активації обох одночасно), поле для введення пошукового запиту (підтримуються ключові слова, хештеги та згадки користувачів), налаштування кількості постів для завантаження та вибір моделі для класифікації з випадного списку. Додатково доступні три режими роботи через вкладки: пошук у соцмережах (за замовчуванням), автоматичні свіжі новини та завантаження з URL-адреси.

Після виконання пошуку система відображає список знайдених публікацій (рис. 2.31) з компактними картками що містять: назву соціальної мережі та статус верифікації автора, текст поста, engagement metrics (кількість фоловерів, постів, взаємодій), час публікації та кнопку "Класифікувати" для запуску аналізу. Клік на

картку відкриває детальну сторінку поста (рис. 2.32) з повним текстом, профілем автора, коментарями та можливістю перейти до оригінальної публікації.

Після натискання "Класифікувати" система виконує повний цикл аналізу обраною моделлю та відображає розгорнуті результати (рис. 2.33): блок LLM Extraction з витягнутими твердженнями та визначенням stance (стверджує/спростовує/нейтральне), блок класифікації з міткою, показником впевненості та текстовим поясненням моделі, блок fact-check з результатами верифікації кожного твердження через Google Fact Check API (включаючи джерело фактчекера, рейтинг та посилання на повну перевірку). Фінальний блок показує консенсус верифікації та порівняння з результатом моделі (збігається/не збігається), що дозволяє оцінити узгодженість машинної класифікації з експертною перевіркою фактів.

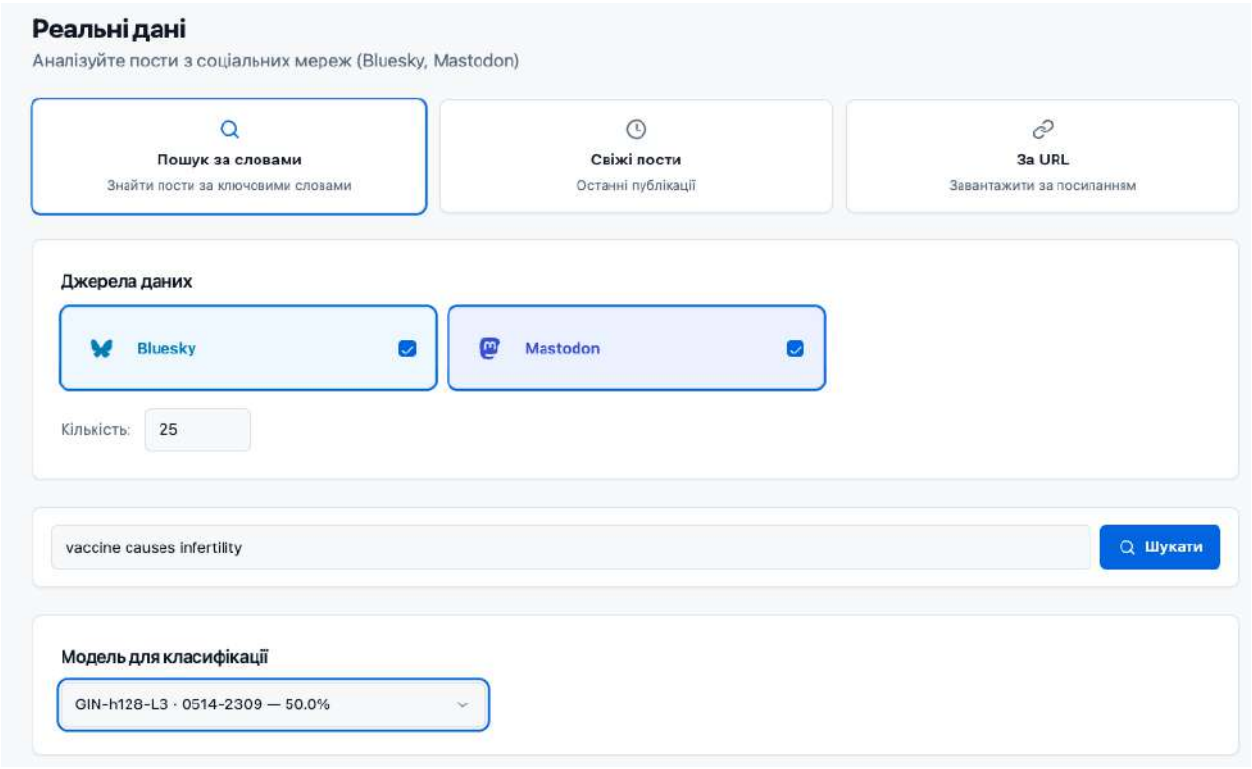

Рисунок 2.29. *Сторінка Реальні дані*

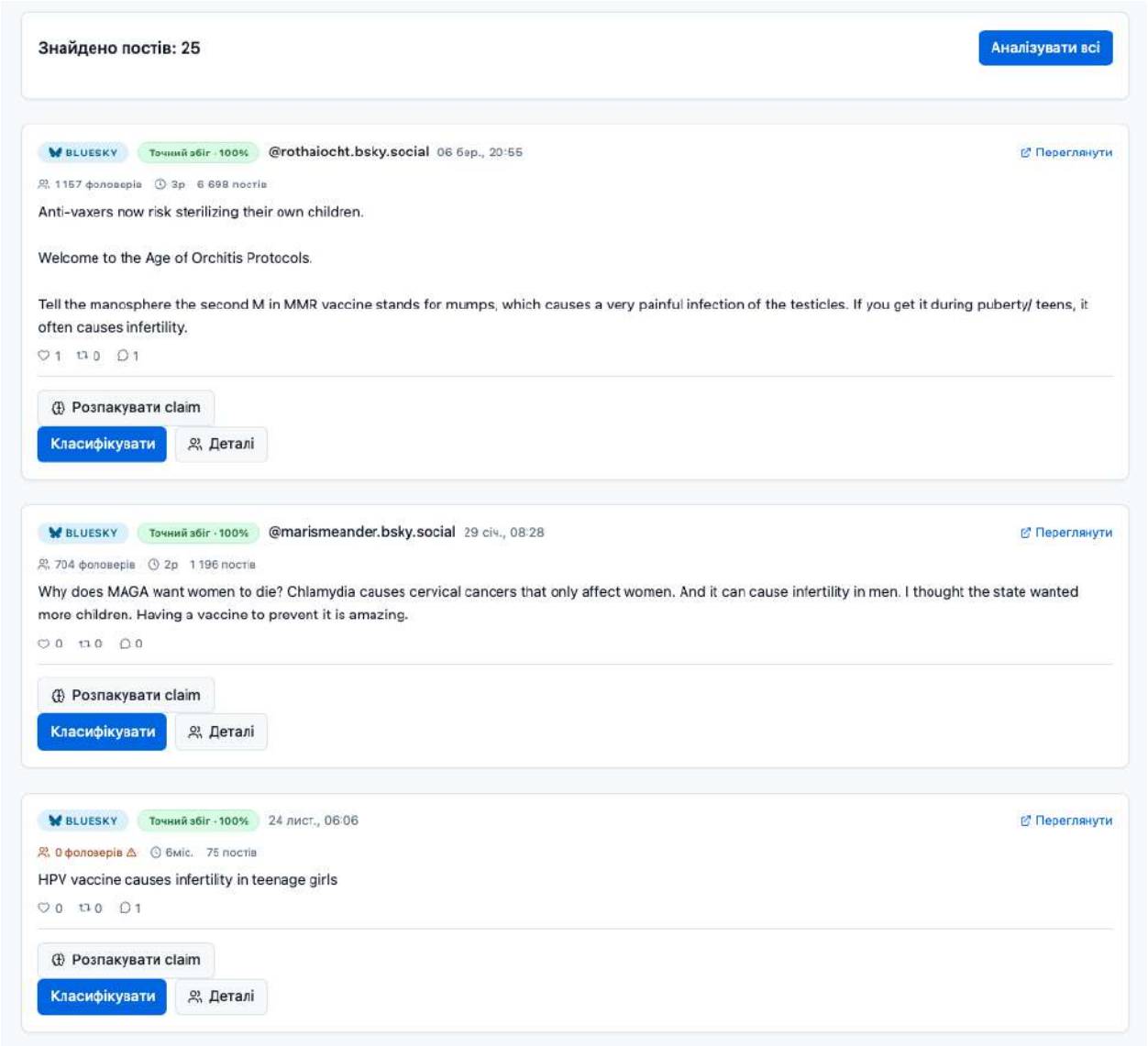

Рисунок 2.30. *Частина результату пошуку*

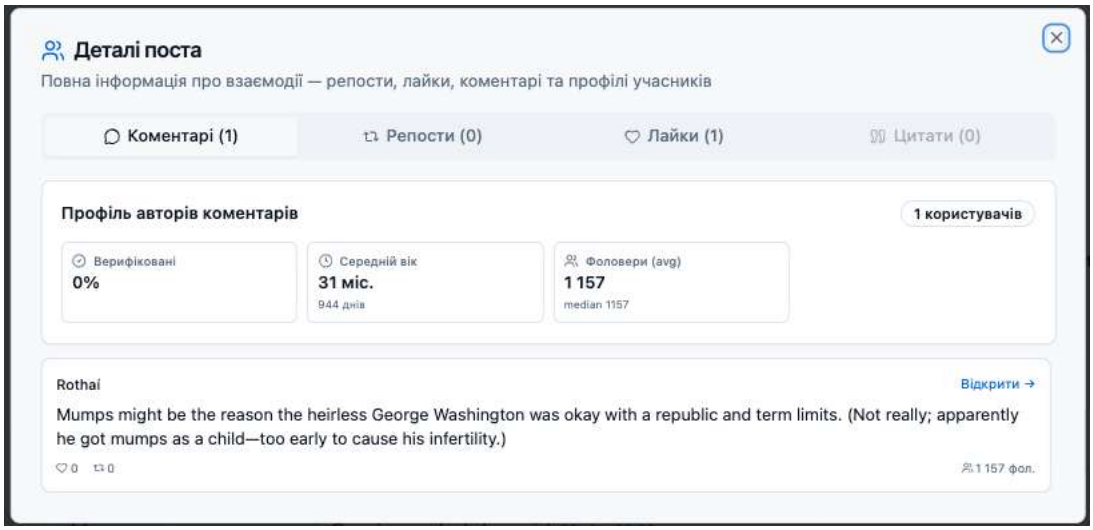

Рисунок 2.31*. Деталі посту*

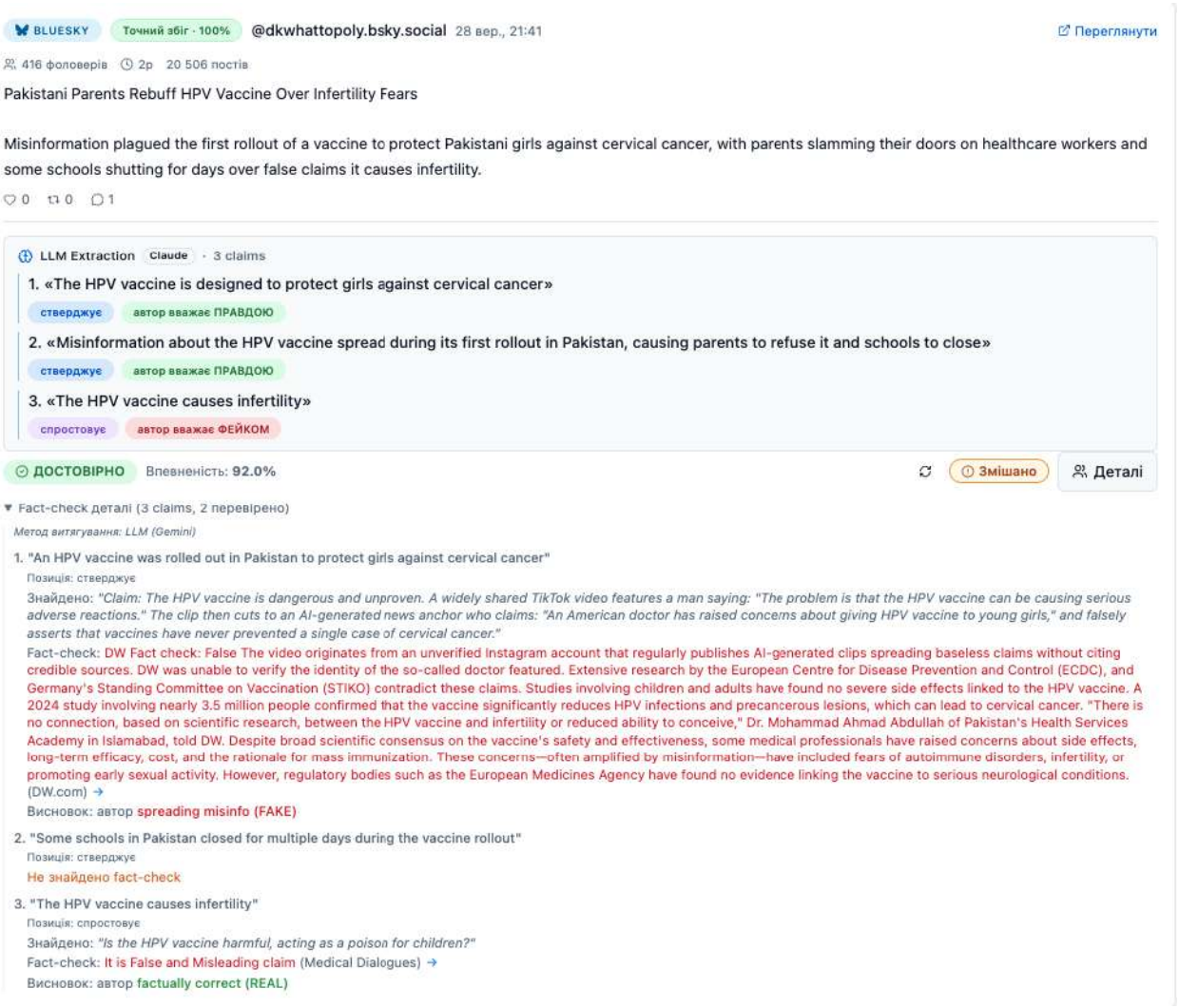

Рисунок 2.32. *Результат екстракції тверджень і класифікація (Claude Opus)*

2.5.2 Управління станом та взаємодія з backend

Усі взаємодії клієнтського застосунку з серверною частиною здійснюються через REST API запити. Для уніфікації використовується сконфігурований інстанс бібліотеки axios з двома interceptor-ами: request-interceptor автоматично додає до кожного запиту заголовок Authorization: Bearer <token>, зчитуючи JWT з localStorage; response-interceptor централізовано перехоплює відповіді зі статусом 401 – у такому разі токен і дані користувача видаляються з localStorage, після чого виконується перезавантаження сторінки, що відкриває екран входу.

Управління станом застосунку побудовано на стандартних хуках React (useState, useEffect). Глобальний стан користувача зберігається у головному

компоненті App та ініціалізується з localStorage при монтуванні; React Context використовується лише для перемикання теми оформлення (світла/темна). Серіалізація тіл запитів у JSON та парсинг відповідей виконуються бібліотекою axios автоматично.

JWT-токен зберігається у localStorage браузера, що забезпечує його збереженість між сесіями. Після успішної автентифікації виконується додатковий запит до /auth/me для отримання повного об'єкта користувача. Перевірка валідності токена відбувається імпліцитно: будь-яка спроба захищеного запиту з невалідним токеном повертає HTTP 401, що автоматично обробляється response-interceptor-ом.

2.6 Висновки до Розділу 2

У цьому розділі розглянуто проєктування та реалізацію програмного прототипу системи виявлення дезінформації. Робота охопила повний цикл практичного втілення: від формулювання функціональних вимог та архітектурних рішень до конкретних деталей реалізації чотирьох парадигм машинного навчання та семи основних сторінок користувацького інтерфейсу.

Ключовими досягненнями реалізованого прототипу є:

Комплексність підходу. Система інтегрує чотири парадигми машинного навчання: статистичну (Naive Bayes з 45 додатковими ознаками), нейромережеву (DistilBERT з контекстним моделюванням послідовностей), графову (GraphSAGE та GIN на графах поширення новин) та підхід на основі великих мовних моделей (Claude у чотирьох режимах промптингу). Це дозволяє провести структуроване порівняння їх ефективності у межах єдиного експериментального середовища. Кожна парадигма має свої переваги: NB забезпечує інтерпретованість та швидкість тренування, DistilBERT – точність через моделювання контекстних залежностей, GNN – врахування структури вірусного поширення публікацій, LLM – здатність до класифікації без попереднього тренування та генерацію текстових пояснень.

Інтеграція з реальним світом. На відміну від більшості академічних робіт, що обмежуються оцінюванням на статичних датасетах, система забезпечує можливість

класифікації реальних публікацій з соціальних мереж Bluesky та Mastodon, а також автоматичної верифікації через Google Fact Check Tools API, що агрегує результати понад 300 фактчекер-організацій. Реалізовано витягування перевірюваних тверджень з тексту через LLM з визначенням позиції автора, нормалізацію рейтингів до єдиної шкали та обчислення консенсусу через мажоритарне голосування. Це дозволяє оцінити ефективність моделей на шумних користувацьких текстах, що принципово відрізняються від відредагованих прикладів тренувального датасету.

Дослідницька гнучкість. Архітектура системи спроектована з урахуванням потреб дослідницького процесу: модульна структура дозволяє легко додавати нові моделі або ознаки, покроковий інтерфейс (wizard) тренування забезпечує повний контроль над гіперпараметрами та наборами ознак, збереження натренованих моделей у Google Drive гарантує відтворюваність експериментів між сесіями. ML-сервер на базі Google Colab надає безкоштовний доступ до GPU T4, що критично важливо для тренування DistilBERT та GNN. Система підтримує створення LLM-пресетів з різними системними промптами та режимами взаємодії, а також побудову ансамблів з трьома стратегіями голосування.

Користувацький досвід. Сім основних сторінок інтерфейсу покривають повний спектр сценаріїв використання: завантаження та перегляд статистики датасетів, налаштування та запуск тренування моделей з візуальним індикатором виконання, перегляд та порівняння метрик натренованих моделей, створення та тестування LLM-пресетів, побудова ансамблів з автоматичним оцінюванням, інтерактивна класифікація текстів з поясненнями рішень, моніторинг та аналіз публікацій соціальних мереж. Інтуїтивний дизайн через компонентну бібліотеку shadcn/ui на Tailwind CSS, візуалізація матриць помилок та хмар найвпливовіших слів роблять систему доступною як для досвідчених дослідників, так і для студентів профільних дисциплін.

У результаті виконаної роботи отримано працюючий наскрізний прототип, що демонструє повний цикл функціональності: від завантаження тренувального датасету через тренування чотирьох типів моделей – до інтерактивної верифікації

публікацій соціальних мереж із автоматичним порівнянням висновків моделей з консенсусом фактчекерів. Усі архітектурні та технологічні рішення обґрунтовано з огляду на специфіку дослідницького завдання. Детальний аналіз ефективності реалізованих моделей – включно з оцінюванням на in-domain та cross-domain сценаріях, дослідженням вкладу ознак та порівнянням парадигм – представлено у наступному розділі.

РОЗДІЛ 3. ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ МОДЕЛЕЙ

3.1 Методологія експериментів

У цьому розділі представлено результати експериментального оцінювання чотирьох парадигм машинного навчання, описаних у Розділі 2, на задачі бінарної класифікації новинних публікацій за категоріями FAKE та REAL. Метою експериментів є порівняння ефективності реалізованих моделей у двох режимах оцінювання – внутрішньодоменному (in-domain) та міждоменному (cross-domain) – а також з'ясування впливу різних груп ознак на якість класифікації через дослідження вкладу ознак.

3.1.1 Опис датасету

Експерименти проведено на датасеті FakeNewsNet, детально описаному у підрозділі 1.2; структуру CSV-файлів наведено у підрозділі 2.3.1.

3.1.2 Розбиття на вибірки

Для оцінювання моделей використано два сценарії розбиття, що відповідають типовим ситуаціям практичного застосування системи. In-domain сценарій імітує випадок коли модель тренується та використовується у межах одного тематичного домену (GossipCop) – оцінює здатність моделей вивчати паттерни дезінформації у конкретному домені. Cross-domain сценарій імітує трансферне навчання – модель тренується на GossipCop, а тестується на PolitiFact без перенавчання – оцінює узагальнюючу здатність моделей та виявляє моделі, що покладаються на domain-specific ознаки. Конкретні розміри вибірок описано у підрозділі 2.3.3. Феномен поширення дезінформації характеризується різними

паттернами у різних тематичних доменах [3] – тому здатність моделі до трансферу є важливим індикатором її практичної придатності.

3.1.3 Метрики оцінювання

Для всебічної оцінки якості класифікації обрано шість метрик.

Accuracy – частка правильно класифікованих публікацій. Хоча ця метрика інтуїтивно зрозуміла, вона може бути оманливою при дисбалансі класів: модель, що завжди прогнозує REAL, отримає accuracy близько 76% на GossipCop попри повну нездатність виявляти дезінформацію. Саме тому необхідно враховувати precision та recall окремо для класу FAKE.

Precision (FAKE) – частка справді фейкових публікацій серед прогнозованих моделлю як FAKE; висока precision означає мало хибних спрацьовувань.

Recall (FAKE) – частка справді фейкових публікацій, що модель виявила; низький recall означає що значна частина дезінформації проходить непоміченою.

F1 (FAKE) – гармонічне середнє між Precision та Recall для класу FAKE; є основною метрикою порівняння моделей у цій роботі.

F1-macro – середнє арифметичне F1 для обох класів, що враховує також якість класифікації достовірних новин як збалансована оцінка загальної ефективності.

ROC-AUC – площа під ROC-кривою (Receiver Operating Characteristic), що характеризує здатність моделі ранжувати приклади за ймовірністю належності до класу FAKE незалежно від порогу класифікації. У межах подальшого аналізу F1 (FAKE) використовується як основна метрика порівняння моделей, ROC-AUC – як додаткова метрика, що виявляє потенціал для покращення через підбір порогу класифікації.

3.1.4 Експериментальне середовище

Усі експерименти проведено на платформі Google Colab з використанням графічного процесора NVIDIA Tesla T4 (16 ГБ відеопам'яті). Програмний стек включає Python 3.11 та такі основні бібліотеки: scikit-learn для реалізації Naive Bayes та обчислення метрик, transformers від HuggingFace для DistilBERT [16], PyTorch та PyTorch Geometric для графових моделей GraphSAGE [9] та GIN [10], sentence-transformers для генерації MiniLM ембедингів вузлів графів, Claude Code CLI для взаємодії з моделями сімейства Claude [20].

Кожна модель тренувалася та оцінювалася один раз на фіксованому розбитті даних (train/val/test). Це не дозволяє оцінити статистичну значущість різниці між моделями та варіативність результатів; оцінювання статистичної значущості відмінностей залишається за межами поточного дослідження.

У цьому підрозділі представлено результати оцінювання чотирьох парадигм класифікації на сценарії in-domain, де тренувальна, валідаційна та тестова вибірки походять з одного домену (GossipCop). Цей сценарій оцінює здатність моделей вивчати паттерни дезінформації у межах конкретного тематичного домену – у даному випадку, новин про знаменитостей та індустрію розваг. Для LLM-підходу (Claude) внутрішньодоменне оцінювання не проводилося, оскільки великі мовні моделі не тренуються на специфічному датасеті і завжди працюють у режимі zero-shot або few-shot – результати Claude представлені у підрозділі 3.5 на тестовій вибірці PolitiFact.

3.2 Базові результати моделей у внутрішньодоменному сценарії

3.2.1 Базова конфігурація Naive Bayes

Базова конфігурація наївного баєсівського класифікатора використовує лише текстові ознаки: TF-IDF векторизацію з unigrams та лематизацією, alpha=1.0 для Лапласового згладжування. Результати оцінювання представлені у таблиці 3.1.

Таблиця 3.1. *Результати Naive Bayes з базовою текстовою конфігурацією на in-domain*

Метрика	Значення
Accuracy	0.832
Precision (FAKE)	0.955
Recall (FAKE)	0.298
F1 (FAKE)	0.454
F1-macro	0.677
ROC-AUC	0.821

Аналіз метрик виявляє характерну для класичного NB поведінку – сильний дисбаланс між precision (0.955) та recall (0.298). При accuracy 83.2% модель здатна виявити лише близько 30% усієї дезінформації у тестовій вибірці. Цей дисбаланс пояснюється консервативною природою NB: модель присвоює мітку FAKE лише при високій впевненості, наслідком чого є висока надійність позитивних прогнозів (95.5% precision) ціною пропуску значної частини дезінформації. Високий показник ROC-AUC (0.821) свідчить що модель добре розрізняє класи на рівні апостеріорних ймовірностей, проте стандартний поріг класифікації 0.5 виявляється недостатньо чутливим. Детальне дослідження впливу різних груп ознак на якість NB наведено у підрозділі 3.3.

3.2.2 DistilBERT з різними конфігураціями

Для DistilBERT проведено п'ять експериментів з різними конфігураціями гіперпараметрів: комбінаціями параметра freeze_base (заморозка базових шарів), максимальної довжини послідовності токенів max_length та кількості епох тренування. Результати представлені у таблиці 3.2.

Таблиця 3.2. *Результати DistilBERT з різними конфігураціями на in-domain*

Конфігурація	Accuracy	Precision	Recall	F1 (FAKE)	F1-macro	ROC-AUC
1 ep, max=128, freeze=True	0.855	0.785	0.523	0.628	0.769	0.872

1 ep, max=128, freeze=False	0.871	0.780	0.627	0.695	0.807	0.908
1 ep, max=256, freeze=True	0.858	0.804	0.525	0.636	0.774	0.874
3 ep, max=256, freeze=True	0.875	0.787	0.643	0.708	0.814	0.904
5 ep, max=256, freeze=True	**0.877**	0.777	**0.668**	**0.718**	**0.820**	**0.910**

Аналіз результатів дозволяє зробити три ключових висновки.

По-перше, повне тонке налаштування (freeze_base=False) суттєво перевершує заморожений режим при одній епосі – F1 зростає з 0.628 до 0.695 (приріст 6.7 п.п.). Це свідчить що адаптація глибших шарів DistilBERT під специфіку домену GossipCop додає значущу інформацію до базових ембедингів. Заморожування базової моделі зменшує час тренування але обмежує здатність до адаптації за обмеженої кількості епох.

По-друге, збільшення max_length зі 128 до 256 токенів при заморожених шарах та одній епосі майже не впливає на якість (F1 змінюється з 0.628 до 0.636). Це означає що для більшості статей GossipCop перші 128 токенів містять достатньо інформації для коректної класифікації – заголовок та початок тексту вже несуть основний сигнал. Однак збільшення max_length подвоює час тренування та споживання пам'яті.

По-третє, збільшення кількості епох тренування дає істотний приріст якості. Конфігурація freeze=True з 1 епохою досягає F1=0. 636, а з 5 епохами – F1=0.718.

Найкраща конфігурація DistilBERT (freeze=True, max=256, 5 епох) досягає F1=0.718, що суттєво перевищує найкращий результат NB з текстом (F1=0.454). Це підтверджує перевагу контекстних ембедингів трансформерів [16] над bag-of-words представленням TF-IDF для задачі виявлення дезінформації.

3.2.3 Графові нейронні мережі (GraphSAGE та GIN)

Для графових моделей обрано конфігурацію з двома шарами згортки, прихованою розмірністю 128, dropout 0.3, learning rate 0.001 та 50 епохами тренування. GraphSAGE використовує mean aggregator для агрегації представлень сусідів. Результати представлені у таблиці 3.3..

Таблиця 3.3. *Результати GraphSAGE та GIN на in-domain*

Конфігурація	**Accuracy**	**Precision**	**Recall**	**F1 (FAKE)**	**F1-macro**	**ROC-AUC**
GIN (sum aggregator)	**0.928**	**0.859**	**0.830**	**0.844**	**0.899**	0.945
GraphSAGE (mean aggregator)	0.918	0.830	0.817	0.824	0.885	**0.960**

GIN досягає найкращого результату серед усіх натренованих на GossipCop моделей (F1=0.844), GraphSAGE демонструє близьку, але дещо нижчу ефективність (F1=0.824). Обидві архітектури суттєво перевершують як NB (F1=0.454), так і найкращу конфігурацію DistilBERT (F1=0.718). F1 для GIN порівняно з DistilBERT свідчить про істотну цінність графової структури як додаткового джерела сигналу понад текст. Це підтверджує тезу теоретичного огляду (підрозділ 1.5.3) про комплементарність текстової та графової інформації [12].

При порівнянні двох архітектур між собою спостерігається цікавий компроміс: GIN має дещо вищий F1 (0.844 проти 0.824), що відповідає його теоретично вищій виразності [10], тоді як GraphSAGE демонструє вищий ROC-AUC (0.960 проти 0.945) – кращу здатність ранжувати приклади за ймовірностями належності до класу FAKE. На практиці обидві архітектури демонструють порівнянну якість, з невеликою перевагою GIN за основною метрикою F1.

3.2.4 Порівняльний аналіз парадигм на in-domain

Зведена таблиця 3.4 представляє найкращі конфігурації кожної парадигми у порядку зростання F1 для класу FAKE.

Таблиця 3.4. *Порівняння найкращих моделей кожної парадигми на in-domain*

Модель	Accuracy	F1 (FAKE)	F1-macro	ROC-AUC
Naive Bayes (text only)	0.832	0.454	0.677	0.821
DistilBERT (best)	0.877	0.718	0.820	0.910
GraphSAGE	0.918	0.824	0.885	**0.960**
GIN	**0.928**	**0.844**	**0.899**	0.945

Результати демонструють чітку ієрархію ефективності парадигм на in-domain сценарії: графові моделі > трансформери > наївний баєс. Кожен наступний крок ієрархії додає принципово нове джерело інформації: DistilBERT додає контекстне розуміння тексту порівняно з bag-of-words NB, а GNN додають структуру соціального поширення порівняно з ізольованим текстом DistilBERT.

Візуалізація порівняння наведена на рисунку 3.1.

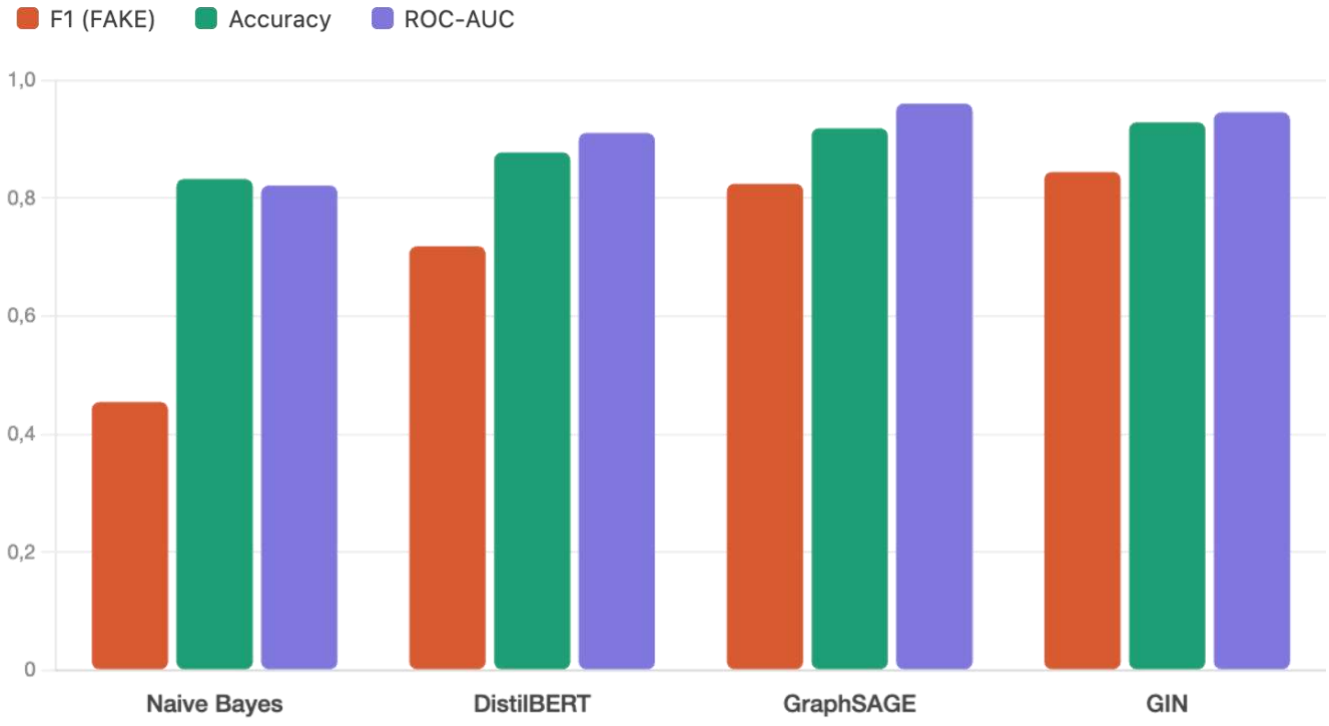

Рисунок 3.1. *Порівняння F1 (FAKE) найкращих моделей кожної парадигми на in-domain GossipCop*

Однак цей рейтинг має суттєве застереження: усі моделі тренувалися та тестувалися на одному домені. Як буде показано у підрозділі 3.4, при перенесенні на інший домен ієрархія кардинально змінюється – модель з найвищою in-domain якістю не обов'язково є найкращою для практичного застосування.

3.3 Дослідження вкладу ознак для Naive Bayes

Базовий результат Naive Bayes на in-domain (F1=0.454) суттєво відстає від трансформерних та графових моделей. Виникає природне запитання: чи можна покращити NB через додавання додаткових ознак, описаних у підрозділі 2.3 – емоційних, стилістичних, соціальних? Для відповіді проведено систематичне дослідження вкладу ознак з дев'ятьма конфігураціями, що покривають різні комбінації груп ознак. Усі експерименти виконано на in-domain тестовій вибірці GossipCop з однаковими гіперпараметрами (ComplementNB, alpha=1.0, TF-IDF unigrams з лематизацією для текстової складової).

3.3.1 Конфігурації експериментів

Розглянуто чотири групи ознак: текстові (TF-IDF unigrams), емоційні (14 ознак з NRC EmoLex [23]), стилістичні (8 ознак за моделлю Хорна та Адалі [24]), соціальні (23 ознаки профілів та поширення). Конфігурації побудовано двома стратегіями: інкрементальним додаванням груп до базової текстової моделі та оцінюванням окремих груп та їх комбінацій без тексту. Повний перелік конфігурацій з результатами представлено у таблиці 3.5.

Таблиця 3.5. *Результати дослідження вкладу ознак для Naive Bayes на in-domain GossipCop*

Конфігурація	**Accuracy**	**Precision**	**Recall**	**F1 (FAKE)**	**F1-macro**	**ROC-AUC**
text only	0.832	0.955	0.298	0.454	0.677	0.821

text + emo	0.818	0.971	0.231	0.373	0.633	0.836
text + emo + sty	0.812	0.973	0.203	0.336	0.613	0.849
text + emo + sty + soc	0.817	1.000	0.221	0.362	0.628	0.943
emo + sty + soc	**0.841**	0.628	0.794	**0.701**	**0.797**	0.897
emo + sty	0.611	0.296	0.477	0.365	0.542	0.598
emo	0.573	0.272	0.488	0.349	0.516	0.559
sty	0.645	0.300	0.382	0.336	0.547	0.592
soc	0.836	0.617	0.794	0.694	0.791	**0.896**

Візуалізація F1 (FAKE) для усіх дев'яти конфігурацій наведена на рисунку 3.2.

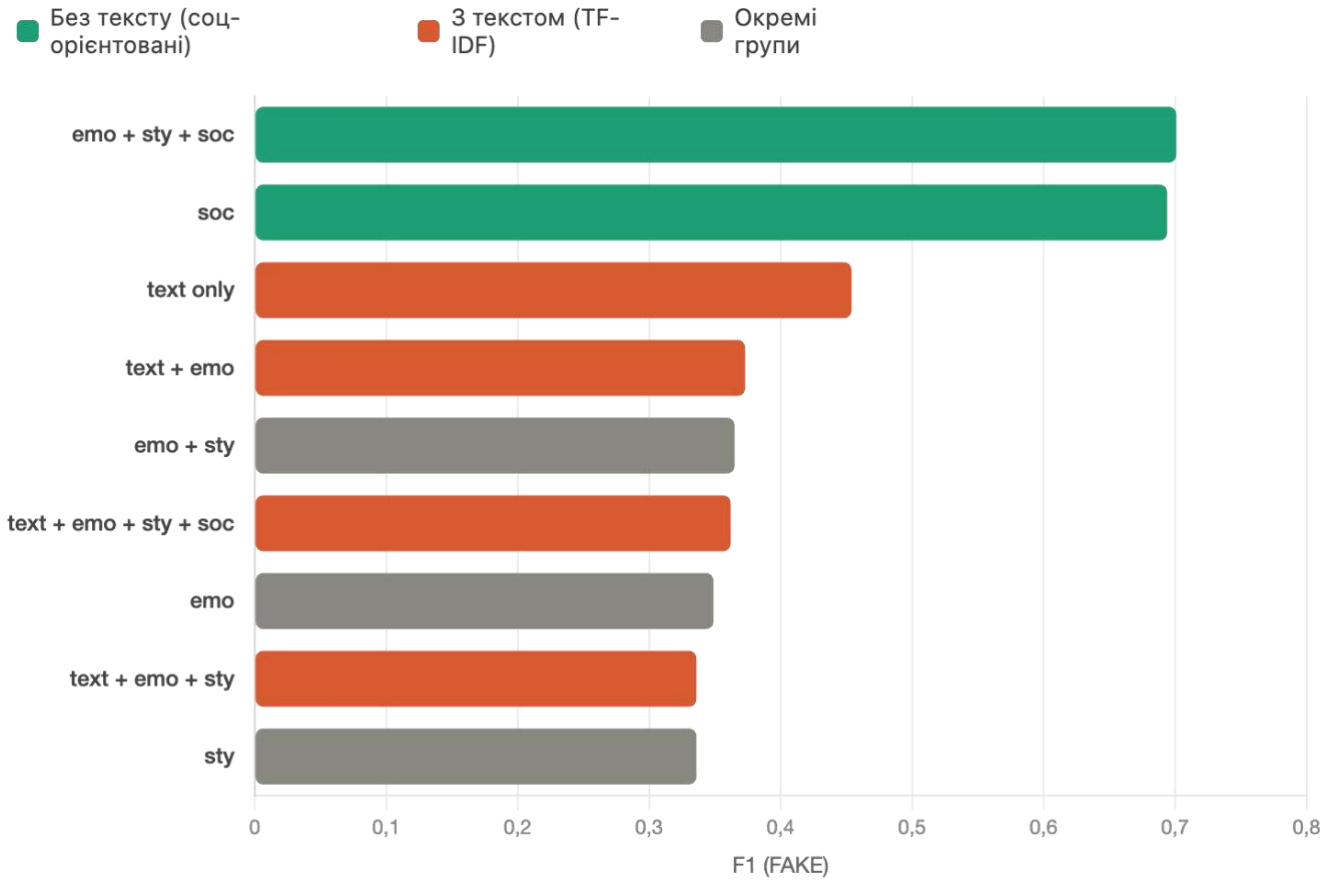

Рисунок 3.2. *F1 (FAKE) для різних комбінацій груп ознак Naive Bayes на in-domain GossipCop*

3.3.2 Парадокс текстових ознак

Результатом дослідження вкладу ознак є наступне спостереження: інкрементальне додавання додаткових груп ознак до текстової моделі поступово знижує F1, а не підвищує його. Базова текстова модель досягає F1=0.454, додавання емоційних ознак знижує F1 до 0.373, додавання стилістичних – до 0.336, додавання соціальних – лише незначно піднімає до 0.362. Конфігурація що містить усі чотири групи ознак показує гірший F1 ніж конфігурація лише з текстом.

Натомість конфігурація без текстових ознак – лише емоційні, стилістичні та соціальні – досягає F1=0.701, що у 1.5 рази перевищує найкращу конфігурацію з текстом. Ще більш разюче: одні лише соціальні ознаки (всього 23 ознаки) дають F1=0.694 – практично таку ж якість як комбінація трьох груп без тексту. Це означає що соціальні ознаки є домінантним джерелом інформації для класифікації GossipCop, тоді як текстові ознаки не лише не додають інформації, але й активно шкодять моделі.

Запропоновано наступне пояснення цього явища. TF-IDF векторизатор створює sparse матрицю з понад п'ятдесятьма тисячами ознак, тоді як кількість додаткових ознак становить лише 45. Через припущення Naive Bayes про незалежність ознак та однакову їх обробку модель не може компенсувати таку диспропорцію вимірності – масивна текстова складова "глушить" сигнал від чисельно невеликих, але дуже інформативних соціальних ознак. Крім того, текстові ознаки GossipCop виявляються слабко дискримінативними: лексика плітковинного контенту схожа у статтях обох класів – обидва часто згадують імена знаменитостей, події у шоу-бізнесі, емоційно забарвлені висловлювання. Натомість соціальні ознаки (структура авторитету акаунту, патерни поширення) виявляються набагато більш дискримінативними: дезінформація поширюється відмінним типом користувачів та з різною інтенсивністю порівняно з достовірними новинами.

3.3.3 Аналіз компромісу precision/recall

Друге важливе спостереження стосується принципово різного характеру компромісу між precision та recall у текстових та соціальних конфігураціях. Текстові конфігурації характеризуються екстремально високим precision (0.955–1.000) при низькому recall (0.203–0.298): модель майже завжди права коли каже FAKE, але пропускає 70–80% дезінформації. Це поведінка занадто консервативної моделі що класифікує як FAKE лише найбільш очевидні випадки.

Натомість конфігурації з соціальними ознаками (emo+sty+soc та soc only) демонструють збалансований профіль: precision ~0.62 при recall ~0.79. Така модель помиляється приблизно у третині випадків коли каже FAKE, але виявляє 80% усієї дезінформації. Для практичної системи виявлення дезінформації другий профіль є набагато бажанішим – пропуск дезінформації нівелює саму мету системи, тоді як помилкові спрацьовування можуть бути відфільтровані додатковими механізмами (наприклад, фактчек-верифікацією, описаною у підрозділі 2.1).

Унікальною є конфігурація text+emo+sty+soc що досягає precision 1.000 – модель не помилилася жодного разу зі своїх прогнозів FAKE. Однак recall лише 0.221 робить таку модель практично непридатною: вона детектує лише п'яту частину дезінформації. Високий ROC-AUC цієї конфігурації (0.943) свідчить що модель добре розрізняє класи на рівні ймовірностей – проблема не у моделі, а у занадто високому пороговому значенні класифікації за замовчуванням.

3.3.4 Висновки дослідження вкладу ознак

Проведене дослідження дозволяє сформулювати три практичних висновки для застосування Naive Bayes у задачі виявлення дезінформації.

По-перше, для домену GossipCop соціальні ознаки є істотно більш інформативним джерелом сигналу ніж текстовий контент. Конфігурація лише з 23 соціальними ознаками досягає F1=0.694, що перевищує конфігурацію з текстом у 1.5 рази. Це не означає, що текст не несе інформації – лише, що у поєднанні з обмеженнями NB (припущення про незалежність, TF-IDF матриця) текстові ознаки шкодять моделі.

По-друге, нагромадження різних груп ознак без врахування їх взаємодії не завжди покращує результат. Інтуїтивне очікування "більше ознак – краще" не справджується для NB: оптимальною є компактна модель без текстових ознак з акцентом на соціальний контекст.

По-третє, низький recall є системною проблемою NB що не вирішується додаванням ознак. Перехід до моделей з контекстними щільними представленнями (DistilBERT) та з графовою структурою поширення (GraphSAGE/GIN) є необхідним для досягнення задовільного recall (підрозділи 3.2.2 та 3.2.3).

3.4 Міждоменні експерименти (GossipCop → PolitiFact)

Cross-domain експерименти оцінюють здатність моделей переносити знання з тренувального домену до семантично відмінного цільового домену без перенавчання. У цьому сценарії моделі тренуються на 14 300 публікаціях GossipCop (новини про знаменитостей) та оцінюються на 590 публікаціях PolitiFact (політичні новини) – два домени, що суттєво відрізняються за тематикою, словником, типом джерел і характером поширення у соціальних мережах.

3.4.1 Очікування та постановка

Перенесення між тематично віддаленими доменами є складною задачею для будь-якої моделі машинного навчання, що ґрунтується на вивченні розподілу тренувальних даних. Це особливо актуально для виявлення дезінформації, де паттерни хибних та достовірних публікацій можуть бути domain-specific: політична дезінформація має іншу структуру, ніж дезінформація про знаменитостей. Метою цього експерименту є з'ясування – які парадигми вивчають універсальні патерни (що переносяться між доменами), а які – специфічні для домену патерни (що залишаються прив'язаними до GossipCop). Результати усіх натренованих моделей на cross-domain split представлено у таблиці 3.6.

Таблиця 3.6. *Результати моделей на cross-domain (GossipCop → PolitiFact)*

Модель	**Accuracy**	**Precision**	**Recall**	**F1 (FAKE)**	**F1-macro**	**ROC-AUC**
NB (text only)	0.439	0.353	0.018	0.035	0.319	0.488
NB (text + emo)	0.442	0.200	0.003	0.006	0.309	0.507
NB (text + emo + sty)	0.442	0.000	0.000	0.000	0.306	0.497
NB (text + emo + sty + soc)	0.442	0.400	0.012	0.024	0.317	0.517
NB (emo + sty + soc)	0.578	0.580	0.880	0.670	0.500	0.510
NB (soc only)	0.581	0.581	0.874	**0.699**	0.507	0.513
DistilBERT (freeze, max=128)	0.531	0.713	0.252	0.372	0.499	0.600
DistilBERT (unfrozen, max=128)	0.559	0.717	0.334	0.456	0.543	0.618
DistilBERT (freeze, max=256)	0.550	0.736	0.291	0.417	0.526	0.610
DistilBERT (freeze, max=256, 3ep)	0.506	0.578	0.395	0.467	0.504	0.520
DistilBERT (freeze, max=256, 5ep)	0.500	0.547	0.457	0.500	0.500	0.500
GIN	0.476	0.523	0.583	0.552	0.461	0.472
GraphSAGE	0.475	0.518	0.696	0.594	0.425	0.407

3.4.2 Naive Bayes: парадокс соціальних ознак

Результати NB на cross-domain виявляють разючу полярність: текстові конфігурації демонструють катастрофічну деградацію, тоді як конфігурації з соціальними ознаками практично зберігають якість. Ця полярність наочно

ілюструється на рисунку 3.3, де представлено зміну F1 кожної з дев'яти конфігурацій NB при переході від in-domain GossipCop до cross-domain PolitiFact.

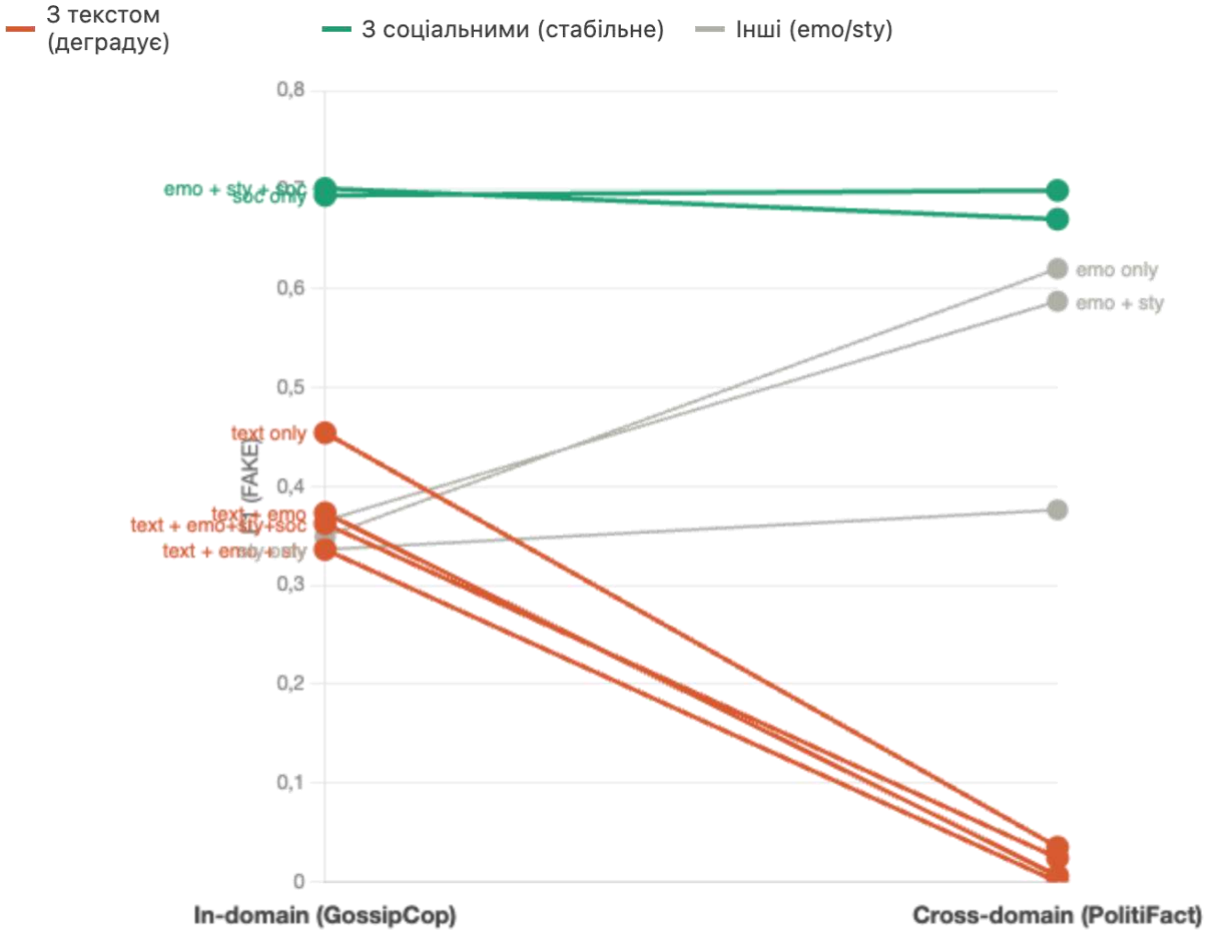

Рисунок 3.3. *Зміна F1 (FAKE) конфігурацій Naive Bayes при перенесенні з GossipCop на PolitiFact*

На графіку виразно проступають три патерни поведінки. Перша група – конфігурації що містять текстові ознаки (виділені теракотовим кольором) – демонструє різке падіння лінії вниз: базова текстова модель з F1=0.454 на in-domain падає до F1=0.035 на cross-domain (деградація 92%), а конфігурація text+emo+sty взагалі досягає F1=0.000 – модель не змогла коректно класифікувати жодної FAKE публікації у тестовій вибірці PolitiFact. Друга група – конфігурації з соціальними ознаками (зелений колір) – демонструє практично горизонтальні лінії: soc only показує F1=0.699 на cross-domain – практично ідентичний результат до F1=0.694 на in-domain (приріст 0.5%), а конфігурація emo+sty+soc – лише незначне падіння з 0.701 до 0.670 (-4.4%). Це означає, що соціальні ознаки кодують універсальний

сигнал дезінформації – патерни поведінки авторів та поширення публікацій є подібними у обох доменах, попри тематичну відмінність.

Несподіваним є поведінка третьої групи – окремі конфігурації emo та sty (сірий колір), що не містять ані тексту, ані соціальних ознак. На in-domain ці моделі є найслабкішими (F1=0.336–0.365), однак при перенесенні на PolitiFact їхня якість парадоксально зростає: F1 для emo піднімається з 0.349 до 0.620, для emo+sty – з 0.365 до 0.587. Це можна пояснити тим, що політичний дискурс PolitiFact містить більш виражені емоційно-стилістичні маркери ніж відносно нейтральний за стилем плітковинний контент GossipCop – політичні публікації систематично використовують більш заряджену лексику, апеляції до емоцій та характерні риторичні прийоми. На GossipCop ці маркери є слабко дискримінативними (новини про знаменитостей загалом емоційні незалежно від достовірності), тоді як на PolitiFact вони стають корисним сигналом.

Пояснення основного результату – катастрофічної деградації текстових конфігурацій – ґрунтується на природі ознак TF-IDF. Текстові ознаки представляють модель розподілу слів у тренувальному корпусі, причому у GossipCop домінують лексичні маркери шоу-бізнесу: імена знаменитостей, тематичні терміни індустрії розваг, специфічна для таблоїдної журналістики стилістика. У PolitiFact ці слова майже відсутні, натомість домінує політичний дискурс. Як наслідок, переважна більшість TF-IDF ознак натренованої на GossipCop моделі не активується для жодного слова PolitiFact публікації, а ті що активуються – пов'язані з найзагальнішою лексикою, що не несе дискримінативного сигналу.

Соціальні ознаки натомість мають універсальну природу. Кількість фоловерів, вік акаунту, статус верифікації, патерни engagement, глибина та ширина каскаду поширення – ці показники описують поведінкові патерни користувачів та структуру вірусного розповсюдження, що не залежить від теми новини. Дезінформація поширюється "далі, швидше, глибше та ширше" незалежно від тематичної категорії [3] – і саме цей універсальний патерн фіксують соціальні ознаки.

3.4.3 DistilBERT

DistilBERT демонструє суттєву, але не катастрофічну деградацію при перенесенні. Найкраща in-domain конфігурація (5 епох, max=256, freeze=True) знижує F1 з 0.718 до 0.500 – деградація 30%. Це проміжний результат між повним крахом NB з текстом та стабільністю NB з соціальними ознаками. Загальна картина деградації для всіх п'яти конфігурацій DistilBERT представлена на рисунку 3.4.

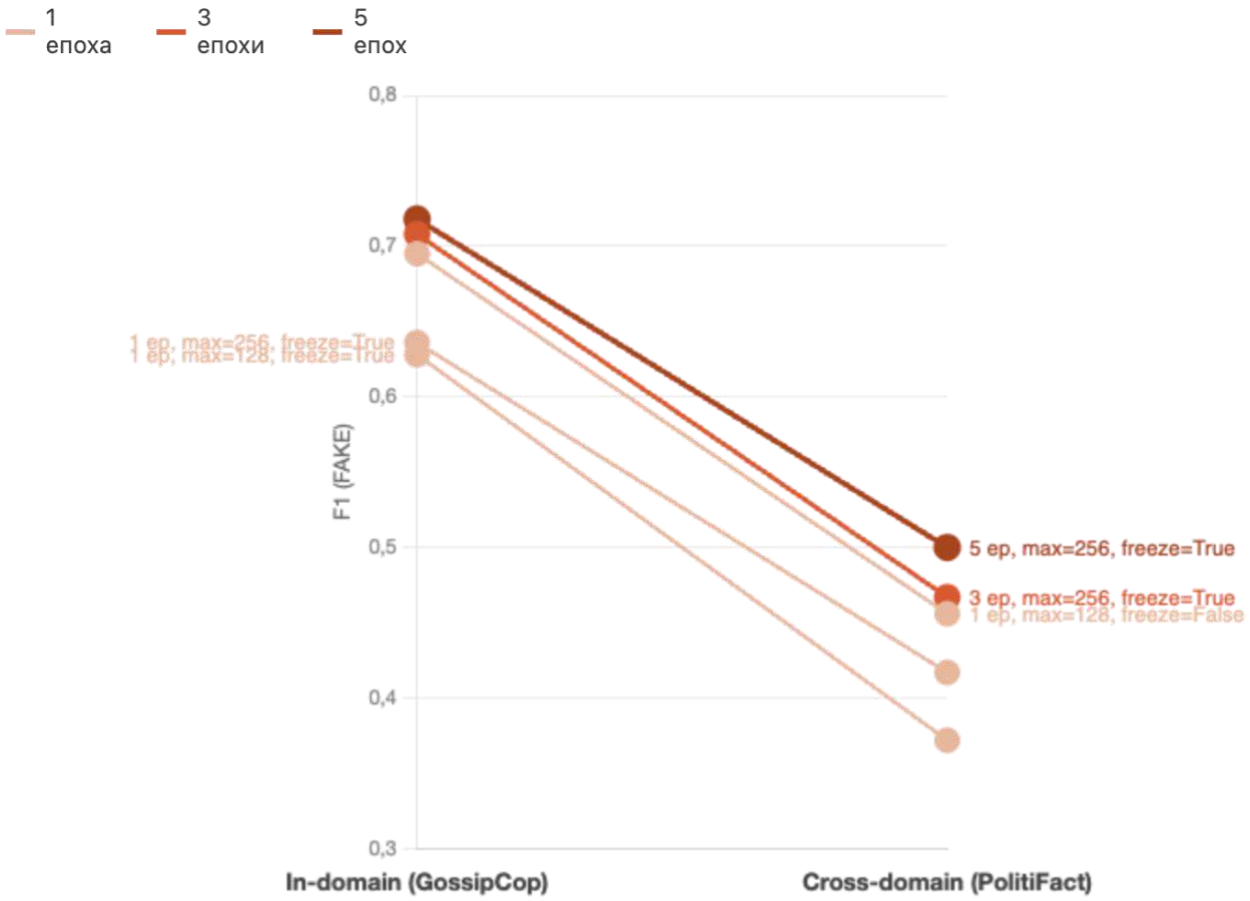

Рисунок 3.4. *Деградація F1(FAKE) конфігурацій DistilBERT при перенесенні з GossipCop на PolitiFact*

На відміну від конфігурацій NB з соціальними ознаками (рисунок 3.3), жодна з п'яти конфігурацій DistilBERT не зберігає якість при перенесенні – усі лінії мають негативний нахил приблизно однакової стрімкості, а абсолютна деградація F1 коливається у вузькому діапазоні 0.218–0.256 балів. Це означає, що жодна з варіацій гіперпараметрів (заморозка шарів, довжина послідовності, кількість епох)

принципово не покращує здатність моделі до domain transfer. Деградація пояснюється тим, що під час fine-tuning трансформер адаптує контекстні представлення до специфіки GossipCop – вивчає характерні фрази, синтаксичні конструкції та стилістичні маркери шоу-бізнесу, які при перенесенні на політичний домен стають швидше шкідливими, ніж корисними.

Особливо показовим є експеримент із поступовим збільшенням кількості епох у конфігураціях з замороженою базовою моделлю та max_length=256. На перший погляд, збільшення епох однозначно корисне: F1 зростає з 0.417 (1 епоха) до 0.500 (5 епох). Однак детальніший аналіз метрик у таблиці 3.7 виявляє небезпечну тенденцію, приховану за цим зростанням

Таблиця 3.7. *Динаміка cross-domain метрик DistilBERT при збільшенні кількості enox (freeze=True, max=256)*

Епох	**F1 (FAKE)**	**Precision**	**Recall**	**ROC-AUC**
1	0.417	0.736	0.291	0.610
3	0.467	0.578	0.395	0.520
5	0.500	0.547	0.457	0.500

F1 послідовно зростає з 0.417 до 0.500, проте ROC-AUC дзеркально симетрично падає з 0.610 до 0.500 – рівня випадкового вибору. Це означає, що модель з 5 епохами тренування фактично втрачає здатність ранжувати приклади за ймовірністю належності до класу FAKE. Уявне покращення F1 пояснюється феноменом overfitting у поєднанні зі зсувом порогу класифікації: з кожною епохою модель глибше адаптується до GossipCop, precision падає, recall зростає, а гармонічне середнє F1 – артефактно збільшується. ROC-AUC, що не залежить від конкретного порогу, об'єктивніше відображає реальну деградацію.

Таким чином, попри найвищий cross-domain F1 саме у конфігурації з 5 епохами, оптимальною є конфігурація з 1 епохою (F1=0.417, ROC-AUC=0.610) – менший F1 компенсується надійнішим ранжуванням та кращою придатністю до калібрування порогу під конкретний use case.

3.4.4 Графові мережі: деградація попри найкращі in-domain результати

GNN демонструють суттєву деградацію попри найвищі результати на in-domain. GIN падає з F1=0.844 до 0.552 (-35%), GraphSAGE – з 0.824 до 0.594 (-28%). Графові моделі, що показали себе найкращими у внутрішньодоменному оцінюванні, при перенесенні стають лише третіми після LLM (підрозділ 3.5) та NB з соціальними ознаками. Масштаб деградації найкраще видно при одночасному розгляді F1 та ROC-AUC, представленому на рисунку 3.5.

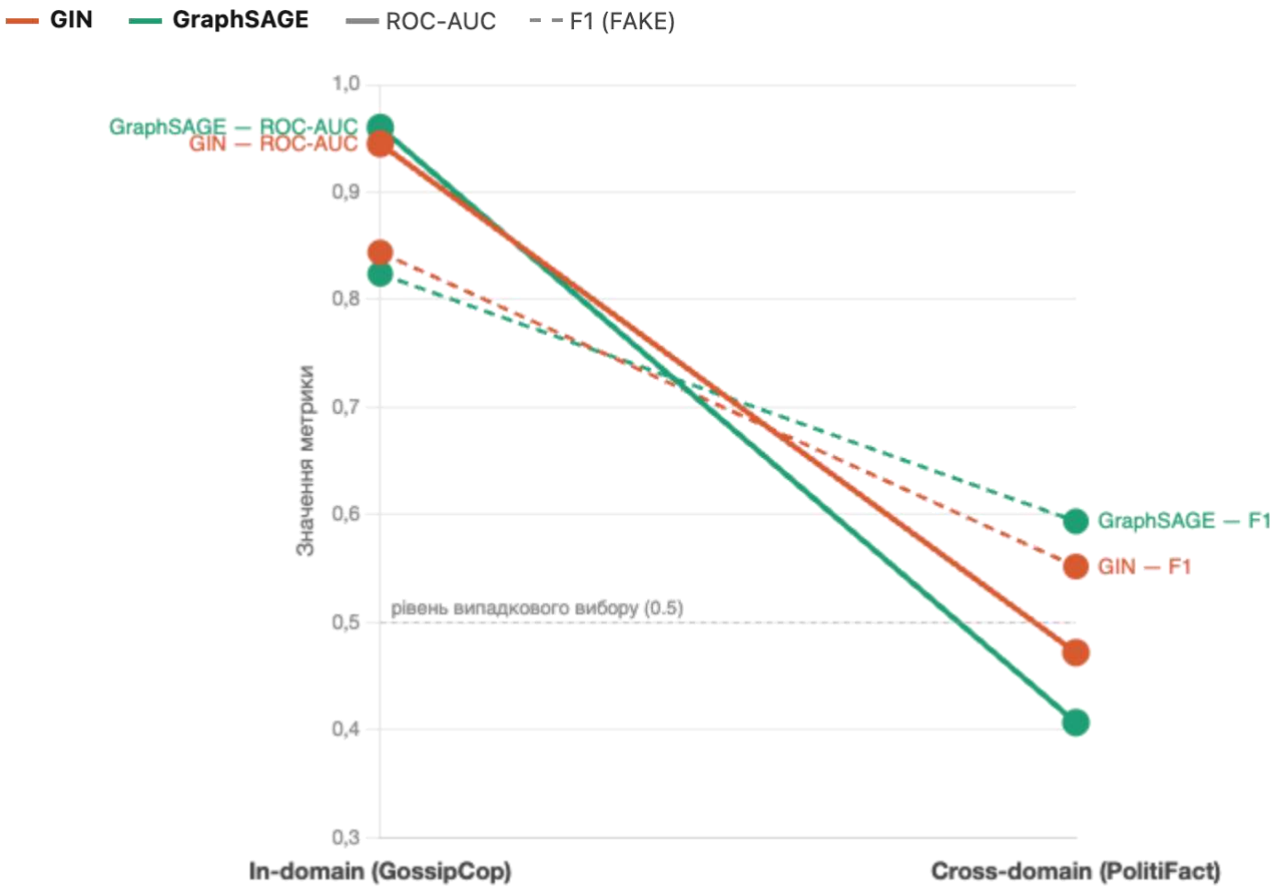

Рисунок 3.5. *Деградація F1 та ROC-AUC моделей GIN та GraphSAGE при перенесенні з GossipCop на PolitiFact*

На графіку виокремлюється поведінка ROC-AUC (суцільні лінії): обидві архітектури демонструють падіння з рівня близько 0.95 на in-domain до 0.472 для GIN та 0.407 для GraphSAGE на cross-domain – нижче пунктирної лінії випадкового

вибору (0.5). Це означає, що моделі ранжують приклади гірше за випадковий вибір – апостеріорна ймовірність FAKE від GraphSAGE для конкретного PolitiFact приклада статистично менш інформативна, ніж результат підкидання монетки. Це сильний сигнал, що домен PolitiFact знаходиться поза розподілом тренувальних даних – модель робить систематично невірні висновки на новому домені.

Причина такої деградації є подвійною. По-перше, MiniLM ембедінги вузлів графа – це текстові представлення твітів, ретвітів та відповідей. При перенесенні на PolitiFact ці ембедінги представляють зовсім іншу мовну дистрибуцію (політичні дискусії проти обговорення знаменитостей), що ускладнює класифікацію навіть якщо графова структура відповідає тій, що бачилася при тренуванні. По-друге, сама структура графів поширення у двох доменах може суттєво відрізнятися: політичні новини мають інший характер каскадів – інші патерни ретвітів, інша глибина обговорень, інша швидкість поширення. У сукупності ці два фактори створюють подвійний distribution shift – і за змістом вузлів, і за топологією графа – що пояснює чому GNN, найсильніші моделі на in-domain, виявляються одними з найслабкіших на cross-domain.

3.4.5 Порівняння деградації між парадигмами

Зведена таблиця 3.8 порівнює ключові конфігурації кожної парадигми у in-domain та cross-domain сценаріях.

Таблиця 3.8. *Деградація моделей при cross-domain transfer*

Модель	**In-domain F1**	**Cross-domain F1**	**Деградація**
NB (text only)	0.454	0.035	-92%
NB (soc only)	0.694	0.699	+1%
NB (emo + sty + soc)	0.701	0.670	-4%
DistilBERT (best)	0.695	0.456	-34%
GraphSAGE	0.824	0.594	-28%
GIN	0.844	0.552	-35%

Візуалізація порівняння наведена на рисунку 3.6.

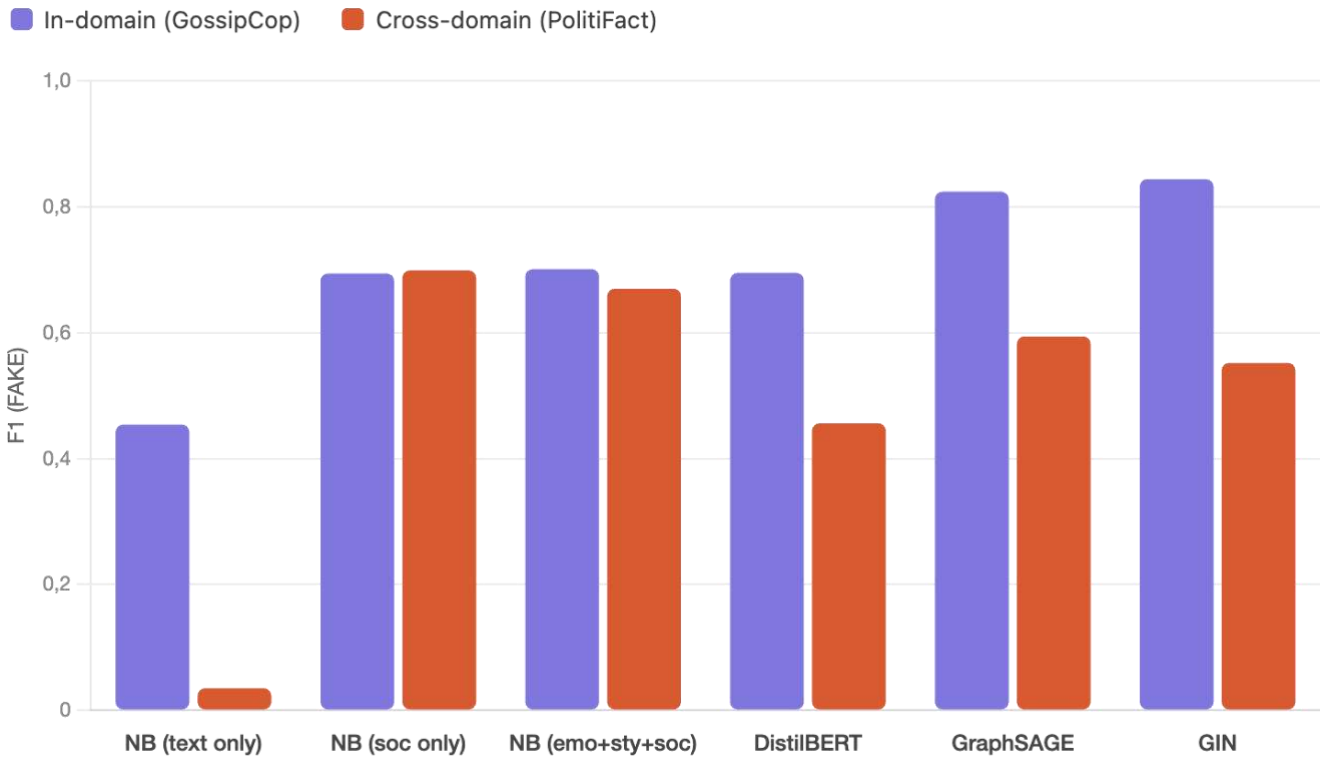

Рисунок 3.6. *Порівняння F1 (FAKE) на in-domain GossipCop та cross-domain PolitiFact для найкращих конфігурацій кожної парадигми*

Цей аналіз дозволяє сформулювати ключовий висновок: жодна з натренованих парадигм не забезпечує задовільної якості при перенесенні на новий домен без перенавчання. Найкращий результат cross-domain серед натренованих моделей – F1=0.699 для NB з соціальними ознаками – суттєво поступається in-domain результатам GIN (F1=0.844). При цьому моделі з найвищою in-domain якістю (GIN, GraphSAGE, DistilBERT) деградують найсильніше, тоді як NB з компактним набором соціальних ознак виявляється найбільш стійкою. Ця "інверсія рейтингу" має важливе практичне значення: при виборі моделі для production системи з можливими *domain shifts* слід орієнтуватися не на максимальний результат на одному датасеті, а на стійкість при тестуванні на множинних доменах.

Альтернативою тренованим моделям, що принципово не страждають від *domain shifts*, є LLM-підхід, який не вимагає тренувального етапу – результати Claude розглядаються у наступному підрозділі.

3.5 LLM-based підхід: Claude як zero-shot класифікатор

LLM-підхід принципово відрізняється від трьох інших парадигм: модель не тренується на GossipCop, а отримує промпт з інструкцією класифікувати текст безпосередньо у режимі inference. У результаті проблема domain shift, що катастрофічно вплинула на треновані моделі (підрозділ 3.4), для LLM не виникає за побудовою – модель завжди використовує свої pre-trained знання незалежно від домену тестових даних. Експерименти проведено лише на cross-domain тестовій вибірці PolitiFact (590 публікацій), оскільки в LLM-парадигмі немає окремого тренувального етапу – результат не залежить від того, який датасет ми називаємо "тренувальним".

3.5.1 Постановка експерименту

Для оцінювання використано три моделі сімейства Claude [20] у порядку зростання обчислювальної потужності: Haiku, Sonnet та Opus. Кожна модель тестувалася у чотирьох конфігураціях за двома осями: режим промптингу (zero-shot або few-shot [18]) та наявність соціального контексту у промпті (text only або text + social context). У zero-shot конфігурації модель отримує лише системний промпт з інструкцією класифікації; у few-shot – додатково 5 прикладів класифікованих публікацій з поясненням прийнятих рішень.

Соціальний контекст подається до моделі у трирівневій структурі. Перший рівень – загальний огляд каскаду поширення: кількість твітів, ретвітів і відповідей, тривалість дискусії, глибина гілок реплаїв, час піку активності, частка верифікованих акаунтів та загальний тон обговорення (підтримка, скептицизм, нейтральна реакція). Другий рівень – топ-5 найвпливовіших твітів з повним текстом, інформацією про автора (нік, статус верифікації, кількість фоловерів, професійна роль, якщо її вдалося визначити) та найпопулярнішою відповіддю під кожним. Впливовість обчислюється як зважена сума показників залученості (ретвіти, лайки, реплаї) та авторитетності джерела. Третій рівень – голоси

меншості: реакції, що не потрапили до топу, з підрахунком таких користувачів (окремо для звичайних та верифікованих акаунтів) та прикладами найдовших подібних твітів.

Повні результати представлено у таблиці 3.9.

Таблиця 3.9. *Результати моделей Claude на cross-domain PolitiFact*

Модель	**Режим**	**Контекст**	**F1 (FAKE)**	**Accuracy**	**Precision**	**Recall**	**ROC-AUC**
Haiku	zero-shot	text	0.810	0.822	0.969	0.696	0.912
Haiku	zero-shot	text+soc	0.876	0.871	0.937	0.822	0.908
Haiku	few-shot	text	0.817	0.821	0.959	0.712	0.866
Haiku	few-shot	text+soc	**0.877**	0.873	0.943	0.819	0.884
Sonnet	zero-shot	text	0.856	0.858	0.969	0.767	0.909
Sonnet	zero-shot	text+soc	0.860	0.854	0.914	0.813	0.895
Sonnet	few-shot	text	0.854	0.854	0.955	0.773	**0.917**
Sonnet	few-shot	text+soc	0.860	0.854	0.917	0.810	0.895
Opus	zero-shot	text	0.826	0.824	0.908	0.758	0.879
Opus	zero-shot	text+soc	0.843	0.832	0.875	0.813	0.884
Opus	few-shot	text	0.821	0.817	0.895	0.758	0.879

Opus	few-shot	text+soc	0.853	0.842	0.880	0.828	0.895

Візуалізація F1 для всіх конфігурацій наведена на рисунку 3.7.

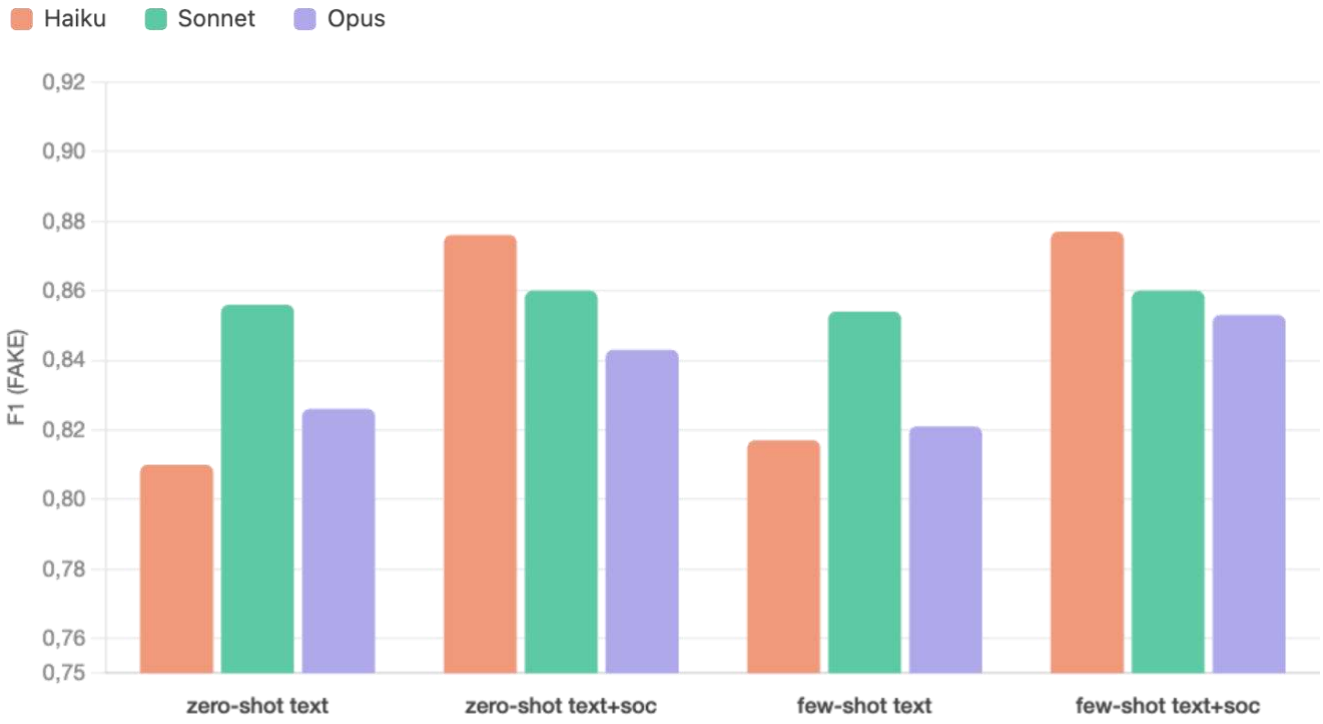

Рисунок 3.7. *F1 (FAKE) для всіх конфігурацій Claude на cross-domain PolitiFact*

3.5.2 Вплив соціального контексту

Систематичним патерном у результатах є позитивний вплив додавання соціального контексту, проте його величина суттєво залежить від моделі. Найсильніший ефект спостерігається для Claude Haiku: додавання соціального контексту підвищує F1 з 0.810 до 0.876 у zero-shot та з 0.817 до 0.877 у few-shot. Це означає що для меншої моделі додаткова інформація про автора та поширення є істотним сигналом, що компенсує обмежену здатність до семантичного аналізу.

Для Claude Sonnet вплив соціального контексту значно слабший: F1 зростає з 0.856 до 0.860 у zero-shot. Сонет, маючи більшу здатність до глибокого аналізу тексту, отримує менше додаткової інформації з соціальних метрик – модель і без

них коректно класифікує більшість прикладів. Для Opus ефект знаходиться посередині: приріст з 0.826 до 0.843 у zero-shot та з 0.821 до 0.853 у few-shot.

Цікавим побічним ефектом додавання соціального контексту є зміщення компромісу precision/recall. У всіх трьох моделях додавання контексту знижує precision (для Haiku з 0.969 до 0.937) і одночасно підвищує recall (з 0.696 до 0.822). Модель стає менш консервативною – соціальні сигнали (наприклад, низький engagement, новий акаунт, велика кількість ретвітів) схиляють модель до класифікації як FAKE навіть у випадках коли текст сам по собі виглядає нейтрально. Цей зсув загалом покращує F1 (гармонічне середнє precision та recall), хоча трохи знижує надійність позитивних прогнозів.

3.5.3 Парадокс розміру моделі

Найбільш показовим результатом експерименту є те, що Claude Haiku – найменша та найдешевша модель сімейства – досягає найкращих результатів. Зокрема, Haiku few-shot text+soc дає F1=0.877, тоді як Opus few-shot text+soc – лише F1=0.853; Sonnet знаходиться між ними з F1=0.860. Це інверсія загальноприйнятого очікування "більша модель – кращий результат" для конкретної задачі бінарної класифікації новинного контенту.

Запропоновано наступне пояснення цього явища. У режимі text only (без соціального контексту) всі три моделі демонструють схожу високу precision (0.908–0.969) при помірному recall (0.696–0.767) – тобто всі вони є консервативними класифікаторами, що рідко помилково ставлять мітку FAKE. На цьому рівні Sonnet навіть дещо випереджає Haiku (F1=0.856 проти 0.810), що відповідає загальному очікуванню переваги більших моделей. Принципова зміна відбувається при додаванні соціального контексту до промпту: recall Haiku зростає з 0.696 до 0.819 (+0.123), тоді як у Sonnet – лише з 0.767 до 0.810 (+0.043), а у Opus – з 0.758 до 0.828 (+0.070). Тобто Haiku виявилася найбільш чутливою до додаткового сигналу від соціального контексту, що компенсує її потенційно слабші аналітичні здібності у режимі чистого тексту.

Одна з можливих причин такого ефекту – більш рішучий характер класифікаційних рішень Haiku порівняно з більш обережними Sonnet та Opus. Останні частіше враховують нюанси та контекстуальні застереження, що може призводити до неоднозначних відповідей у пограничних випадках. Sonnet має найвищий ROC-AUC (0.917 у few-shot text only) – отже, найкраща калібрація ймовірностей серед усіх моделей, – але при бінарному порозі рішучіша Haiku виграє за F1.

3.5.4 Незначний вплив режиму few-shot

Очікувалося, що додавання прикладів класифікованих публікацій (few-shot) суттєво покращить якість порівняно з zero-shot – це класичний результат для більшості NLP задач [18]. Однак наші експерименти не підтверджують цього очікування для задачі виявлення дезінформації на тестовому датасеті. Для Claude Haiku text+soc різниця між zero-shot (F1=0.876) та few-shot (F1=0.877) становить лише 0.1 п.п. – пренебрежно мала варіація. Для Sonnet обидва режими дають однаковий результат як на text only (F1=0.856 vs 0.854), так і на text+soc (обидва 0.860). Лише для Opus спостерігається обмежений ефект – на конфігурації text+soc few-shot покращує F1 з 0.843 до 0.853 (+0.01), тоді як на text only few-shot навпаки незначно погіршує результат (з 0.826 до 0.821).

Цей результат свідчить, що сучасні великі мовні моделі Claude мають достатньо передзнань про феномен дезінформації, щоб коректно класифікувати тексти без додаткових прикладів. Системний промпт з інструкцією та критеріями класифікації є достатнім сигналом – додавання прикладів не суттєво розширює "знання" моделі про задачу. З практичної точки зору це означає, що zero-shot режим є оптимальним: він простіший у конфігурації, споживає менше токенів промпту та не потребує підбору якісних прикладів для few-shot частини.

3.5.5 Порівняння з тренованими моделями

Зведена таблиця 3.10 порівнює найкращу конфігурацію Claude (Haiku few-shot text+soc, F1=0.877) з найкращими тренованими моделями на cross-domain.

Таблиця 3.10. *Порівняння LLM з тренованими моделями на cross-domain*

Модель	**F1 (FAKE)**	**Accuracy**	**ROC-AUC**	**Відносна якість**
Claude Haiku (few-shot, text+soc)	**0.877**	**0.873**	0.884	+25.5% vs NB best
Claude Sonnet (few-shot, text only)	0.854	0.854	**0.917**	+22.2% vs NB best
NB (soc only)	0.699	0.581	0.513	–
GraphSAGE	0.594	0.475	0.407	-15.0%
GIN	0.552	0.476	0.472	-21.0%
DistilBERT (5ep)	0.500	0.500	0.500	-28.5%

LLM-підхід забезпечує найкращий результат cross-domain серед усіх досліджених парадигм – Claude Haiku перевершує найкращу треновану модель (NB з соціальними ознаками) на 25.5% за F1. Більше того, навіть найгірша конфігурація Claude (Haiku zero-shot text only з F1=0.810) перевершує найкращу треновану модель. Це переконливо демонструє практичну цінність LLM-підходу для задач де очікується суттєвий domain shift або де неможливо передбачити характеристики тестових даних.

Однак LLM-підхід має і свої обмеження. По-перше, кожна класифікація вимагає звернення до зовнішнього API, що додає мережеву затримку та робить систему залежною від доступності API провайдера. По-друге, навіть найдешевша Claude Haiku коштує $1.00 за мільйон вхідних токенів та $5.00 за мільйон вихідних, що при масштабних обсягах класифікації стає істотним фактором (особливо при використанні соціального контексту, що збільшує кількість токенів). По-третє, моделі надаються "як є" – Anthropic, на відміну від ряду інших провайдерів, не пропонує загальнодоступного API для тонкого налаштування Claude під специфічні домени. Тому LLM-підхід оптимальний для сценаріїв з помірними

обсягами класифікації (десятки тисяч публікацій на день), очікуваним domain diversity та критичністю якості класифікації; для масових сценаріїв (мільйони публікацій) з фіксованим доменом тонко налаштовані спеціалізовані моделі залишаються економічно ефективнішим вибором.

3.6 Порівняльний аналіз парадигм

Експерименти підрозділів 3.2–3.5 дозволяють провести систематичне порівняння чотирьох парадигм виявлення дезінформації за двома осями: ефективність на in-domain сценарії та стійкість при перенесенні до іншого домену. Цей підрозділ синтезує отримані результати, визначає сильні та слабкі сторони кожної парадигми та формулює практичні рекомендації для різних сценаріїв застосування.

3.6.1 Зведене порівняння

Таблиця 3.11 містить ключові конфігурації кожної парадигми; повна таблиця результатів усіх 26 конфігурацій у обох сценаріях оцінювання наведена у Таблиці А.1 (Додаток А).

Таблиця 3.11. *Ключові конфігурації моделей у in-domain та cross-domain сценаріях"*

Парадигма	**Найкраща конфігурація**	**In-domain F1**	**Cross-domain F1**	**Деградація**
Naive Bayes	text only	0.454	0.035	-92%
Naive Bayes	soc only	0.694	0.699	+1%
Naive Bayes	emo + sty + soc	0.701	0.670	-4%
DistilBERT	freeze=True, max=256, 5ep	0.718	0.500	-30%
GraphSAGE	optimized	0.824	0.594	-28%
GIN	optimized	**0.844**	0.552	-35%

Claude (LLM)	Haiku few-shot text+soc	–	**0.877**	–

Візуалізація відношення in-domain та cross-domain якості представлена на рисунку 3.8.

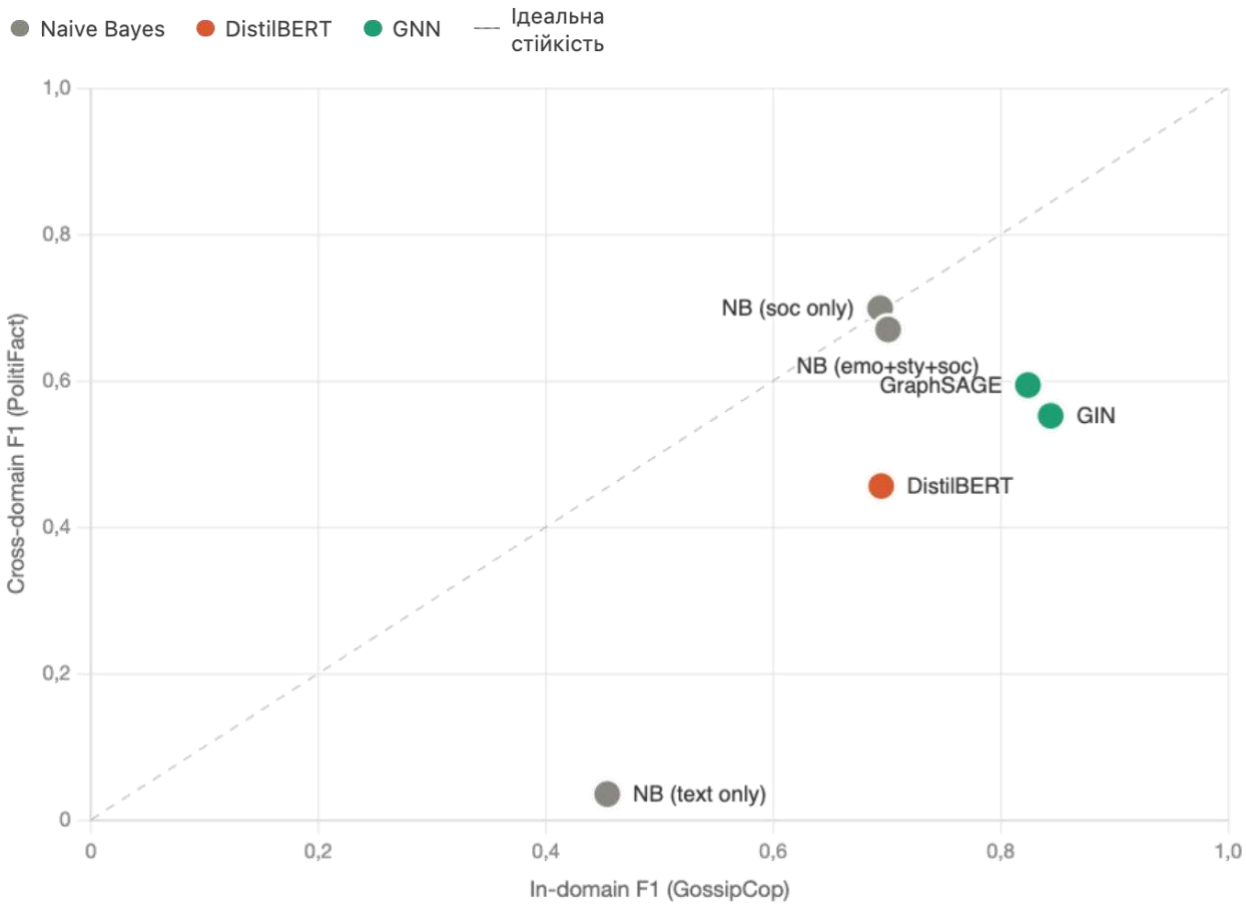

Рисунок 3.8. *Стійкість моделей до domain shift: F1 на in-domain GossipCop проти F1 на cross-domain PolitiFact*

З аналізу таблиці випливає ключове спостереження: жодна модель не є оптимальною у обох сценаріях одночасно. Модель з найвищою in-domain якістю (GIN, F1=0.844) демонструє значну деградацію (35%) при перенесенні. Найбільш стабільна між доменами модель – NB з соціальними ознаками (деградація лише 4%) – не досягає максимумів ані in-domain (де переважає GIN), ані cross-domain (де переважає Claude).

Це означає, що вибір моделі для практичного застосування є не однопараметричною оптимізацією, а компромісом між кількома критеріями:

очікуваним обсягом domain diversity, наявністю якісного тренувального датасету для цільового домену, обчислювальними ресурсами, вимогами до інтерпретованості та допустимою вартістю одиничної класифікації.

3.6.2 Сильні та слабкі сторони парадигм

Naive Bayes має дві принципово різні модальності залежно від набору ознак. У текстовій конфігурації NB виявляється практично непридатним для виявлення дезінформації: низький recall на in-domain (0.298) та повний крах на cross-domain (F1=0.035). У соціальній конфігурації NB перетворюється на найбільш стабільну модель серед усіх досліджених – деградація лише 4%. Сильні сторони соціальної NB: обчислювальна простота (тренування за хвилини на CPU), інтерпретованість через витягування топ-ознак, мала вимога до даних (всього 23 ознаки), стабільність між доменами. Слабка сторона: помірна абсолютна якість, що не дотягує до сучасних трансформерних та LLM-підходів.

DistilBERT забезпечує контекстне розуміння тексту, що відсутнє у bag-of-words NB. Сильні сторони: висока in-domain якість (F1=0.718) при помірних обчислювальних вимогах, широка доступність pretrained ваг, добре документований процес fine-tuning [16]. Слабкі сторони: суттєва деградація при domain shift (−30%), небезпека overfitting при збільшенні кількості епох (підрозділ 3.4.3), необхідність обмежувати максимальну довжину послідовності для довгих статей.

Графові нейронні мережі (GraphSAGE, GIN) використовують структуру каскаду поширення новини як додаткове джерело сигналу понад текст. Сильні сторони: найкраща in-domain якість (F1=0.844 для GIN), здатність моделювати соціальний контекст у його повній структурній складності, теоретично обґрунтована виразність GIN у сенсі WL-тесту [10]. Слабкі сторони: суттєва деградація при domain shift (−28% до −35%) попри найкращі in-domain результати, складність побудови графів (вимагає кількох пов'язаних CSV-таблиць з налаштованими foreign keys), різниця мовної дистрибуції між тренувальним та

цільовим доменами призводить до зниження якості ембедингів вузлів на новому домені, складність інтерпретації – графова структура важко зводиться до зрозумілого людині пояснення.

LLM-підхід (Claude) принципово відрізняється відсутністю тренувального етапу та використанням знань pretrained моделі. Сильні сторони: найкраща cross-domain якість (F1=0.877 для Haiku), неможливість виникнення проблеми domain shift через відсутність тренувального етапу, текстове пояснення кожного рішення, можливість швидкого адаптування системи через зміну промпту без перенавчання моделі. Слабкі сторони: залежність від зовнішнього API Anthropic, мережева затримка на запит, вартість API викликів, що стає істотною при масштабних обсягах класифікації, неможливість гарантованого відтворення поведінки моделі при оновленні версій провайдером.

3.6.3 Практичні рекомендації

Спираючись на проведений аналіз, можна сформулювати наступні рекомендації для вибору моделі залежно від сценарію застосування.

Сценарій 1: Закритий стабільний домен з великим тренувальним корпусом. Якщо система розгортається для конкретної тематичної ніші (наприклад, новини про охорону здоров'я, фінансові новини, спортивна журналістика) і доступний достатній за обсягом domain-specific тренувальний датасет – оптимальним є вибір графових моделей GIN або GraphSAGE. Вони забезпечують найвищу in-domain якість завдяки використанню структури соціального поширення. Спрощена альтернатива – DistilBERT, що працює лише з текстом статті і не вимагає побудови графів соціального поширення.

Сценарій 2: Відкритий домен з очікуваним domain diversity. Якщо система має обробляти публікації з різних тематичних доменів, або якщо цільові домени відомі тільки частково – рекомендується LLM-підхід через Claude API. Дослідження показало, що Claude Haiku забезпечує найкращий результат cross-

domain (F1=0.877) серед усіх досліджених підходів. Додавання соціального контексту у промпт є істотно корисним, особливо для меншої моделі Haiku.

Сценарій 3: Production-середовище з обмеженими обчислювальними ресурсами. Якщо існують жорсткі обмеження на обчислювальні ресурси (відсутність GPU) та вимоги до низької затримки класифікації, при цьому домен фіксований – оптимальним є вибір Naive Bayes з конфігурацією emo + sty + soc. Модель тренується за хвилини на CPU, класифікація працює без GPU з низькою затримкою, є інтерпретовною, та забезпечує помірно стабільну якість (F1=0.670–0.701) як для in-domain, так і для cross-domain сценаріїв.

Сценарій 4: Висока критичність якості з можливістю поєднання підходів. Якщо висока ціна помилкових класифікацій (як false positive – помилкова класифікація достовірних публікацій як FAKE, так і false negative – пропуск дезінформації) і доступні ресурси для гібридних підходів – рекомендується побудова ансамблю з LLM та однією або кількома тонко налаштованими моделями. У внутрішньодоменному сценарії ансамбль може складатися з GIN + Claude, у cross-domain – з NB (soc) + Claude. Описана у Розділі 2 система підтримує побудову ансамблів з трьома стратегіями голосування (hard, soft, weighted), що дозволяє експериментально знаходити оптимальні комбінації.

3.6.4 Загальне спостереження про вплив соціального контексту

Окрім різниці між парадигмами, дослідження виявило що соціальний контекст є наскрізно корисним додаванням для більшості моделей. У Naive Bayes соціальні ознаки роблять модель практично domain-invariant. У Claude додавання соціального контексту покращує F1 для всіх трьох моделей сімейства. У GNN соціальний контекст є невід'ємною частиною графової структури. Винятком є DistilBERT, що у поточній реалізації використовує лише текст; інтеграція соціальних метаданих у трансформерну архітектуру через multimodal механізми становить напрям для подальших досліджень.

Цей результат узгоджується з моделлю Tri-Relationship[2], яка постулює одночасне врахування контенту, соціального контексту та часової динаміки для ефективного виявлення дезінформації. Наші експерименти підтверджують цю модель емпірично: жоден з підходів що ігнорує соціальний контекст не досягає топ-результатів ані in-domain ані cross-domain. Дезінформація не існує у вакуумі тексту – вона невід'ємно пов'язана з поведінкою її поширювачів та структурою її розповсюдження у соціальній мережі.

3.7 Висновки до Розділу 3

У цьому розділі представлено результати експериментального оцінювання чотирьох парадигм виявлення дезінформації – статистичної (Naive Bayes), нейромережевої (DistilBERT), графової (GraphSAGE, GIN) та підходу на основі великих мовних моделей (Claude) – на датасеті FakeNewsNet у двох сценаріях: внутрішньодоменному (GossipCop → GossipCop) та міждоменному (GossipCop → PolitiFact). Проведено систематичні експерименти, що включають базові порівняння моделей, дослідження вкладу ознак для Naive Bayes з дев'ятьма конфігураціями груп ознак та оцінювання дванадцяти конфігурацій.

Ключові результати дослідження можна сформулювати у вигляді чотирьох головних знахідок.

Перша знахідка: ієрархія парадигм радикально змінюється при перенесенні доменів. На in-domain GossipCop спостерігається послідовна ієрархія від найгіршої до найкращої моделі: NB text only (F1=0.454) < DistilBERT (**F1=0.718**) < GraphSAGE (F1=0.824) < GIN (F1=0.844). Графові моделі забезпечують найвищу якість завдяки використанню структури соціального поширення новин, що недоступна іншим парадигмам. Однак при перенесенні до PolitiFact ієрархія інвертується: GIN та GraphSAGE деградують на 28–35% (до F1=0.552–0.594), DistilBERT – на 30% (до F1=0.500), тоді як NB з соціальними ознаками демонструє практично нульову деградацію (F1=0.699 проти 0.694 на in-domain). Найкращий

результат cross-domain (F1=0.877) забезпечує Claude Haiku – модель, що не зазнає впливу domain shift через відсутність тренувального етапу.

Друга знахідка: соціальний контекст є наскрізно стабільним сигналом дезінформації. Дослідження вкладу ознак Naive Bayes (підрозділ 3.3) виявило парадоксальний результат: інкрементальне додавання текстових ознак до соціальних знижує F1 з 0.701 до 0.336–0.362. Конфігурація лише з 23 соціальними ознаками досягає F1=0.694 на in-domain – у 1.5 рази вище за найкращу конфігурацію з текстом. Більше того, ця ж конфігурація переноситься без втрати якості на PolitiFact (F1=0.699). Це означає, що поведінкові патерни поширювачів дезінформації (профіль авторитету, структура engagement, характеристики каскаду) є універсальними маркерами, що працюють незалежно від тематичного домену.

Третя знахідка: LLM без тренування перевершує всі треновані моделі на cross-domain. Claude Haiku у режимі few-shot з соціальним контекстом досягає F1=0.877 на PolitiFact, що на 25.5% перевищує найкращу треновану модель (NB soc only) та на 75% перевищує найкращу конфігурацію DistilBERT. Несподіваним є те, що менша модель Haiku забезпечує кращі результати, ніж потужніші Sonnet та Opus – ймовірно, через більшу чутливість Haiku до сигналу соціального контексту у промпті (підрозділ 3.5.3).

Четверта знахідка: текстові ознаки в Naive Bayes принципово не переносяться між доменами. Базова конфігурація NB з TF-IDF на GossipCop показує F1=0.454, але на PolitiFact падає до F1=0.035 – деградація 92%. Конфігурація text+emo+sty взагалі досягає F1=0.000 на cross-domain – модель не класифікує коректно жодного FAKE прикладу. Пояснення полягає у vocabulary mismatch: TF-IDF словник з близько 50 000 ознак, натренованих на лексиці контенту про знаменитостей, майже не активується для політичних публікацій. Це обмеження стосується лише NB-подібних моделей, що обробляють текст як bag-of-words без контекстного розуміння; трансформерні моделі (DistilBERT) використовують pretrained ембединги, які частково компенсують vocabulary shift, а LLM повністю розв'язує проблему завдяки масштабним pretrained знанням.

Експерименти підтверджують, що задача виявлення дезінформації не має універсального розв'язку: оптимальний підхід залежить від характеристик цільового домену, доступних обчислювальних ресурсів та вимог до інтерпретованості. Розроблений у Розділі 2 програмний прототип, що інтегрує усі чотири парадигми у єдину систему з підтримкою ансамблів, забезпечує необхідну гнучкість для адаптації до різноманітних практичних сценаріїв та слугує платформою для подальших досліджень.

ВИСНОВКИ

У межах магістерської роботи проведено комплексне порівняльне дослідження ефективності чотирьох парадигм машинного навчання у задачі автоматизованого виявлення дезінформації у соціальних мережах та розроблено інтегрований програмний прототип системи, що об'єднує ці парадигми у єдиному дослідницькому середовищі. Поставлені у вступі завдання виконано у повному обсязі.

Здійснено огляд літератури з виявлення дезінформації, проаналізовано теоретичні основи та практичні обмеження чотирьох парадигм машинного навчання: класичних статистичних методів на прикладі наївного баєсівського класифікатора, трансформерних архітектур (BERT, DistilBERT), графових нейронних мереж (GraphSAGE, GIN) та підходу на основі великих мовних моделей (сімейство Claude). Розглянуто тривимірне представлення дезінформації за моделлю Tri-Relationship, а також підходи до інженерії ознак (TF-IDF, NRC EmoLex, стилістичні маркери, соціальні ознаки). Окремо проаналізовано існуючі датасети та обґрунтовано вибір FakeNewsNet як основи для практичної частини роботи завдяки наявності тематично відмінних піддатасетів (GossipCop та PolitiFact) та повного соціального контексту у вигляді Twitter-каскадів.

Спроектовано та реалізовано програмний прототип системи з трирівневою архітектурою: рівень клієнта (React + TypeScript), рівень бізнес-логіки (FastAPI з SQLite через SQLAlchemy) та рівень машинного навчання (Flask на Google Colab з GPU T4). Розроблено уніфікований інтерфейс класифікації, що абстрагує специфіку кожної парадигми та дозволяє побудову ансамблів з трьома стратегіями голосування. Реалізовано інтеграцію з зовнішніми сервісами: Bluesky та Mastodon для доступу до публічних публікацій, Google Fact Check Tools API для незалежної верифікації результатів, моделями сімейства Claude для класифікації через CLI-інструмент Claude Code.

Проведено систематичні експерименти у двох сценаріях оцінювання: внутрішньодоменному (GossipCop → GossipCop) та міждоменному (GossipCop →

PolitiFact). Загалом оцінено 26 конфігурацій моделей: дев'ять конфігурацій Naive Bayes з різними групами ознак, п'ять конфігурацій DistilBERT з варіюванням гіперпараметрів, дві графові архітектури та дванадцять конфігурацій Claude. Оцінювання виконано за шістьма метриками: Accuracy, Precision(FAKE), Recall(FAKE), F1 (FAKE), F1-macro та ROC-AUC.

Головні наукові результати роботи можна сформулювати у вигляді чотирьох знахідок.

Першою знахідкою є інверсія ієрархії ефективності парадигм при перенесенні між тематичними доменами. На in-domain GossipCop спостерігається послідовна ієрархія від найгіршої до найкращої моделі: Naive Bayes з текстовими ознаками (F1=0.454) поступається DistilBERT (F1=0.718), який, у свою чергу, поступається графовим мережам – GraphSAGE (F1=0.824) та GIN (F1=0.844). Однак при перенесенні до PolitiFact ця ієрархія кардинально змінюється: моделі з найвищою внутрішньодоменною якістю (GIN та GraphSAGE) демонструють найбільшу деградацію (28–35%), DistilBERT втрачає 30% якості, тоді як Naive Bayes з соціальними ознаками демонструє практично нульову деградацію. Цей результат має принципове практичне значення: при виборі моделі для production-системи з можливими domain shifts слід орієнтуватися не на максимальний результат на одному датасеті, а на стійкість до зміни домену.

Другою знахідкою є виявлення універсального характеру соціальних ознак як сигналу дезінформації. Дослідження вкладу ознак Naive Bayes продемонструвало парадоксальний результат: інкрементальне додавання текстових ознак до соціальних знижує F1, а не підвищує його; конфігурація лише з 23 соціальними ознаками досягає F1=0.694 на in-domain – у 1.5 рази вище за найкращу конфігурацію з текстом, та переноситься без втрати якості на PolitiFact (F1=0.699). Це означає, що поведінкові патерни поширювачів дезінформації (профіль авторитету акаунту, структура engagement, характеристики каскаду) є універсальними маркерами, які працюють незалежно від тематичного домену.

Третьою знахідкою є перевага LLM-підходу через Claude над усіма натренованими моделями у міждоменному сценарії за відсутності тренувального

етапу. Claude Haiku у режимі few-shot з соціальним контекстом досягає F1=0.877 на PolitiFact, що на 25.5% перевищує найкращу треновану модель (Naive Bayes з соціальними ознаками) та на 75% – найкращу конфігурацію DistilBERT. Несподіваним аспектом цього результату є те, що менша та найдешевша модель сімейства Haiku забезпечує кращі результати, ніж потужніші Sonnet та Opus – ймовірно, через більшу чутливість Haiku до сигналу соціального контексту у промпті.

Четвертою знахідкою є непереносність текстових TF-IDF ознак між тематичними доменами для статистичних моделей. Базова конфігурація Naive Bayes з TF-IDF на GossipCop демонструє F1=0.454, але на PolitiFact падає до F1=0.035 – деградація сягає 92%. Конфігурація з усіма текстовими, емоційними та стилістичними ознаками взагалі досягає F1=0.000, не класифікуючи коректно жодного FAKE прикладу. Це обмеження стосується лише моделей, що обробляють текст як bag-of-words без контекстного розуміння; трансформерні моделі (DistilBERT) частково компенсують vocabulary shift через pretrained ембединги, а великі мовні моделі повністю розв'язують проблему завдяки масштабним pretrained знанням.

На основі отриманих результатів сформульовано практичні рекомендації для вибору моделі залежно від сценарію застосування: для закритого стабільного домену з великим тренувальним корпусом оптимальними є графові моделі GIN/GraphSAGE; для відкритого домену з очікуваною різноманітністю – LLM-підхід через Claude; для production-середовища з обмеженими обчислювальними ресурсами – Naive Bayes з соціальними ознаками.

Практичне значення отриманих результатів полягає у розробці програмного прототипу системи виявлення дезінформації, що підтримує тренування та порівняння моделей чотирьох парадигм, ансамблеві стратегії об'єднання прогнозів та інтеграцію з зовнішніми фактчекер-сервісами. Сформульовані практичні рекомендації можуть бути використані розробниками систем медіа-моніторингу, інформаційно-аналітичних платформ та інструментів підтримки журналістської роботи. Розроблений прототип також може слугувати дослідницьким середовищем

для подальшого вивчення феномену дезінформації – зокрема, для перевірки нових методів інженерії ознак, тестування додаткових LLM-моделей або експериментів з гібридними архітектурами.

Перспективними напрямами подальших досліджень є: розширення експериментів на інші тематичні домени для перевірки універсальності виявлених закономірностей; дослідження динамічних графових моделей (Temporal GNN) для врахування часової еволюції каскадів поширення; вивчення гібридних архітектур типу ARG (Adaptive Rationale Guidance), що поєднують переваги тонко налаштованих моделей та LLM; розширення джерел соціального контексту через інтеграцію додаткових платформ (Discord, Reddit, Facebook); адаптація системи для виявлення дезінформації в українськомовному медіа-просторі – задачі, що набуває особливої актуальності в умовах інформаційного протистояння.

Експерименти переконливо демонструють, що задача виявлення дезінформації у соціальних мережах не має універсального розв'язку: оптимальний підхід залежить від характеристик цільового домену, доступних обчислювальних ресурсів, вимог до інтерпретованості та допустимої вартості одиничної класифікації. Систематичне порівняння чотирьох парадигм машинного навчання у межах єдиного експериментального середовища, проведене у даній роботі, заповнює існуючий пробіл у літературі та надає об'єктивну основу для прийняття обґрунтованих рішень при побудові реальних систем виявлення дезінформації.

ДОДАТКИ

Додаток А. Повна таблиця результатів експериментів

Таблиця А.1. Повні результати оцінювання всіх моделей на in-domain (GossipCop→ GossipCop) та cross-domain (GossipCop → PolitiFact) сценаріях

#	Модель	Конфігурація	Accuracy	Precision (FAKE)	Recall (FAKE)	F1 (FAKE)	F1-macro	ROC-AUC
In-domain (GossipCop → GossipCop)								
1	NB	text only (1,1 lem, α=1)	0.832	0.955	0.298	0.454	0.677	0.821
2	NB	text + emo (1,1 lem, α=1)	0.818	0.971	0.231	0.373	0.633	0.836
3	NB	text + emo + sty (1,1 lem, α=1)	0.812	0.973	0.203	0.336	0.613	0.849
4	NB	text + emo + sty + soc (1,1 lem, α=1)	0.817	1.000	0.221	0.362	0.628	0.943
5	NB	emo + sty + soc (α=1)	0.841	0.628	0.794	0.701	0.797	0.897
6	NB	emo + sty (α=1)	0.611	0.296	0.477	0.365	0.542	0.598
7	NB	emo (α=1)	0.573	0.272	0.488	0.349	0.516	0.559
8	NB	sty (α=1)	0.645	0.300	0.382	0.336	0.547	0.592
9	NB	soc (α=1)	0.836	0.617	0.794	0.694	0.791	0.896
10	DistilBERT	1 ep, max=128, freeze=True	0.855	0.785	0.523	0.628	0.769	0.872
11	DistilBERT	1 ep, max=128, freeze=False	0.871	0.780	0.627	0.695	0.807	0.908

12	DistilBERT	1 ep, max=256, freeze=True	0.858	0.804	0.525	0.636	0.774	0.874
13	DistilBERT	3 ep, max=256, freeze=True	0.875	0.787	0.643	0.708	0.814	0.904
14	DistilBERT	5 ep, max=256, freeze=True	**0.877**	0.777	0.668	**0.718**	**0.820**	**0.910**
15	GIN	hidden=128, 2 шари, dropout=0.3, lr=0.001, 50 ep, sum	**0.928**	**0.859**	**0.830**	**0.844**	**0.899**	0.945
16	GraphSAGE	hidden=128, 2 шари, dropout=0.3, lr=0.001, 50 ep, mean	0.918	0.830	0.817	0.824	0.885	**0.960**
Cross-domain (GossipCop → PolitiFact)								
17	NB	text only (1,1 lem, α=1)	0.439	0.353	0.018	0.035	0.319	0.488
18	NB	text + emo (1,1 lem, α=1)	0.442	0.200	0.003	0.006	0.309	0.507
19	NB	text + emo + sty (1,1 lem, α=1)	0.442	0.000	0.000	0.000	0.306	0.497
20	NB	text + emo + sty + soc (1,1 lem, α=1)	0.442	0.400	0.012	0.024	0.317	0.517
21	NB	emo + sty + soc (α=1)	0.578	0.580	0.880	0.670	0.500	0.510
22	NB	emo + sty (α=1)	0.518	0.558	0.619	0.587	0.504	0.473
23	NB	emo (α=1)	0.554	0.586	0.660	0.620	0.540	0.575

24	NB	sty (α=1)	0.415	0.459	0.319	0.376	0.413	0.446
25	NB	soc (α=1)	0.581	0.581	0.874	0.699	0.507	0.513
26	DistilBERT	1 ep, max=128, freeze=True	0.531	0.713	0.252	0.372	0.499	0.600
27	DistilBERT	1 ep, max=128, freeze=False	0.559	0.717	0.334	0.456	0.543	0.618
28	DistilBERT	1 ep, max=256, freeze=True	0.550	0.736	0.291	0.417	0.526	0.610
29	DistilBERT	3 ep, max=256, freeze=True	0.506	0.578	0.395	0.467	0.504	0.520
30	DistilBERT	5 ep, max=256, freeze=True	0.500	0.547	0.457	0.500	0.500	0.500
31	GIN	hidden=128, 2 шари, dropout=0.3, lr=0.001, 50 ep, sum	0.476	0.523	0.583	0.552	0.461	0.472
32	GraphSAGE	hidden=128, 2 шари, dropout=0.3, lr=0.001, 50 ep, mean	0.475	0.518	0.696	0.594	0.425	0.407
33	Claude Haiku	zero-shot, text only	0.822	0.969	0.696	0.810	0.822	0.912
34	Claude Haiku	zero-shot, text + soc	0.871	0.937	0.822	0.876	0.871	0.908
35	Claude Haiku	few-shot, text	0.821	0.959	0.712	0.817	0.821	0.866
36	Claude Haiku	few-shot, text + soc	**0.873**	0.943	0.819	**0.877**	**0.873**	0.884

37	Claude Sonnet	zero-shot, text only	0.858	0.969	0.767	0.856	0.858	0.909
38	Claude Sonnet	few-shot, text only	0.854	0.955	0.773	0.854	0.854	**0.917**
39	Claude Sonnet	zero-shot, text + soc	0.854	0.914	0.813	0.860	0.854	0.895
40	Claude Sonnet	few-shot, text + soc	0.854	0.917	0.810	0.860	0.854	0.895
41	Claude Opus	zero-shot, text only	0.824	0.908	0.758	0.826	0.824	0.879
42	Claude Opus	few-shot, text only	0.817	0.895	0.758	0.821	0.817	0.879
43	Claude Opus	zero-shot, text + soc	0.832	0.875	0.813	0.843	0.831	0.884
44	Claude Opus	few-shot, text + soc	0.842	0.880	0.828	0.853	0.842	0.895

Примітки:

- Жирним виділено найкращі значення метрик у відповідних групах
- "lem" – попередня обробка з лематизацією
- "(1,1)" – n-gram range (тільки unigrams)
- "α=1" – параметр Лапласового згладжування Naive Bayes
- "emo" – емоційні ознаки (14), "sty" – стилістичні ознаки (8), "soc" – соціальні ознаки (23)
- "freeze" – заморозка базових шарів DistilBERT
- "max" – максимальна довжина послідовності токенів
- Розміри вибірок: in-domain test – 3066 публікацій (GossipCop), cross-domain test – 590 публікацій (PolitiFact)

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Wardle C., Derakhshan H. Information Disorder: Toward an Interdisciplinary Framework for Research and Policy Making. Strasbourg: Council of Europe, 2017. 107 p.
2. Shu K., Sliva A., Wang S., Tang J., Liu H. Fake News Detection on Social Media: A Data Mining Perspective. ACM SIGKDD Explorations Newsletter. 2017. Vol. 19, № 1. P. 22–36. DOI: 10.1145/3137597.3137600.
3. Vosoughi S., Roy D., Aral S. The spread of true and false news online. Science. 2018. Vol. 359, № 6380. P. 1146–1151. DOI: 10.1126/science.aap9559.
4. 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: ACL, 2017. P. 422–426. DOI: 10.18653/v1/P17-2067.
5. Ahmed H., Traore I., Saad S. Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments. Lecture Notes in Computer Science, vol. 10618. Cham: Springer, 2017. P. 127–138. DOI: 10.1007/978-3-319-69155-8_9.
6. Shu K., Mahudeswaran D., Wang S., Lee D., Liu H. FakeNewsNet: A Data Repository with News Content, Social Context, and Spatiotemporal Information for Studying Fake News on Social Media. Big Data. 2020. Vol. 8, № 3. P. 171–188. DOI: 10.1089/big.2020.0062.
7. Cui L., Lee D. CoAID: COVID-19 Healthcare Misinformation Dataset. arXiv preprint arXiv:2006.00885. 2020. URL: https://arxiv.org/abs/2006.00885.
8. Granik M., Mesyura V. Fake News Detection Using Naive Bayes Classifier. 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON). Kyiv: IEEE, 2017. P. 900–903. DOI: 10.1109/UKRCON.2017.8100379.
9. Hamilton W., Ying Z., Leskovec J. Inductive Representation Learning on Large Graphs. Proceedings of the 31st International Conference on Neural Information

Processing Systems (NIPS 2017). Long Beach, CA, USA: Curran Associates Inc., 2017. P. 1024–1034.

10. Xu K., Hu W., Leskovec J., Jegelka S. How Powerful are Graph Neural Networks? Proceedings of the 7th International Conference on Learning Representations (ICLR 2019). New Orleans, LA, USA, 2019. URL: https://openreview.net/forum?id=ryGs6iA5Km.
11. Sharma K., Zhang Y., Ferrara E., Liu Y. Identifying Coordinated Accounts on Social Media through Hidden Influence and Group Behaviours. Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21). Virtual Event, Singapore: ACM, 2021. P. 1441–1451. DOI: 10.1145/3447548.3467391.
12. Monti F., Frasca F., Eynard D., Mannion D., Bronstein M. M. Fake News Detection on Social Media using Geometric Deep Learning. arXiv preprint arXiv:1902.06673. 2019. URL: https://arxiv.org/abs/1902.06673.
13. Kazemi S. M., Goel R., Jain K., et al. Representation Learning for Dynamic Graphs: A Survey. Journal of Machine Learning Research. 2020. Vol. 21, № 70. P. 1–73.
14. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser L., Polosukhin I. Attention Is All You Need. Advances in Neural Information Processing Systems 30 (NIPS 2017). Long Beach, CA, USA: Curran Associates Inc., 2017. P. 5998–6008.
15. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (NAACL-HLT 2019). Minneapolis, MN, USA: Association for Computational Linguistics, 2019. P. 4171–4186. DOI: 10.18653/v1/N19-1423.
16. Sanh V., Debut L., Chaumond J., Wolf T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing (NeurIPS 2019). Vancouver, BC, Canada, 2019. arXiv:1910.01108. URL: https://arxiv.org/abs/1910.01108.

17. Patel J., Bhatt C. M., Trivedi H., Nguyen T. T. Misinformation Detection using Large Language Models with Explainability. arXiv preprint arXiv:2510.18918. 2025. URL: https://arxiv.org/abs/2510.18918.

18. Brown T. B., Mann B., Ryder N., Subbiah M., Kaplan J., Dhariwal P., Neelakantan A., Shyam P., Sastry G., Askell A., et al. Language Models are Few-Shot Learners. Advances in Neural Information Processing Systems 33 (NeurIPS 2020). Vancouver, BC, Canada: Curran Associates Inc., 2020. P. 1877–1901.

19. Wei J., Wang X., Schuurmans D., Bosma M., Ichter B., Xia F., Chi E. H., Le Q. V., Zhou D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. Advances in Neural Information Processing Systems 35 (NeurIPS 2022). New Orleans, LA, USA: Curran Associates Inc., 2022. P. 24824–24837.

20. Anthropic. Introducing the next generation of Claude. Anthropic Blog. March 4, 2024. URL: https://www.anthropic.com/news/claude-3-family.

21. Hu B., Sheng Q., Cao J., Shi Y., Li Y., Wang D., Qi P. Bad Actor, Good Advisor: Exploring the Role of Large Language Models in Fake News Detection. Proceedings of the AAAI Conference on Artificial Intelligence. 2024. Vol. 38, № 20. P. 22105–22113. DOI: 10.1609/aaai.v38i20.30214.

22. Salton G., Buckley C. Term-weighting approaches in automatic text retrieval. Information Processing & Management. 1988. Vol. 24, № 5. P. 513–523. DOI: 10.1016/0306-4573(88)90021-0.

23. Mohammad S. M., Turney P. D. Crowdsourcing a Word-Emotion Association Lexicon. Computational Intelligence. 2013. Vol. 29, № 3. P. 436–465. DOI: 10.1111/j.1467-8640.2012.00460.x.

24. Horne B. D., Adali S. This Just In: Fake News Packs a Lot in Title, Uses Simpler, Repetitive Content in Text Body, More Similar to Satire than Real News. Proceedings of the 11th International AAAI Conference on Web and Social Media (ICWSM 2017). Montréal, Québec, Canada: AAAI Press, 2017. P. 759–766.

25. International Fact-Checking Network (IFCN). The Commitments of the Code of Principles. Poynter Institute, 2015. URL: https://ifcncodeofprinciples.poynter.org/.

26. Google. Fact Check Tools API. Google Developers Documentation. URL: https://developers.google.com/fact-check/tools/api.
27. Hassan N., Arslan F., Li C., Tremayne M. Toward Automated Fact-Checking: Detecting Check-worthy Factual Claims by ClaimBuster. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17). Halifax, NS, Canada: ACM, 2017. P. 1803–1812. DOI: 10.1145/3097983.3098131.