Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему *Метод автоматизованого виявлення суїцидальних намірів у повідомленнях користувачів засобами обробки природної мови*

Галузь знань *12 – Інформаційні технології*
Шифр і назва галузі знань

Спеціальність *122 – Комп'ютерні науки*
Шифр і назва спеціальності

Освітня програма *Комп'ютерні науки*
Назва освітньої програми

Виконав: *студент групи КН-22-2* _____ *Павло ШЕВЧУК*
Група виконавця | Підпис | Ім'я, ПРІЗВИЩЕ

Керівник: *асистент кафедри КН* _____ *Валерія КЛІМЕНКО*
Науковий ступінь, посада | Підпис | Ім'я, ПРІЗВИЩЕ

Нормоконтроль: *к.т.н., доц. каф. КН* _____ *Руслан БАГРІЙ*
Науковий ступінь, посада | Підпис | Ім'я, ПРІЗВИЩЕ

До захисту допускаю:
Зав. кафедри КН, д.т.н., професор _____ *Олександр БАРМАК*
Підпис | Ім'я, ПРІЗВИЩЕ

17 червня 2026 р.

Хмельницький 2026

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

ЗАТВЕРДЖУЮ
Завідувач кафедри *комп'ютерних наук*

_____
(підпис)
д.т.н., професор Олександр БАРМАК
«*22*» *січня* *2026* року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

1. Тема кваліфікаційної роботи бакалавра: *«Метод автоматизованого виявлення суїцидальних намірів у повідомленнях користувачів засобами обробки природної мови»*

2. Завдання видано студенту *Павлу Шевчуку*
(Ім'я, прізвище)

3. Керівник роботи *асистент кафедри КН Валерія Кліменко*
(посада, ім'я, прізвище)

4. Затверджено наказом університету від «*20*» *січня* *2026* р. № *7*

5. Дата видачі завдання студенту: «*22*» *січня* *2026* р.

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

Мета роботи – підвищення ефективності процесу виявлення суїцидальних намірів у повідомленнях користувачів засобами обробки природної мови. Для досягнення мети слід виконати такі задачі: провести дослідження предметної області для задачі виявлення суїцидальних намірів у текстових повідомленнях; розробити метод автоматизованого виявлення суїцидальних намірів у повідомленнях користувачів; здійснити програмну реалізацію інтелектуальної системи виявлення суїцидальних намірів за текстовим представленням та провести дослідження ефективності розробленого методу.

7. Календарний план виконання кваліфікаційної роботи бакалавра:

№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	*Вибір напряму дослідження та узгодження теми кваліфікаційної роботи з керівником, складання календарного графіка виконання*	*січень 2026*	*виконано*
2	*Ознайомлення з предметною областю, формулювання мети і задач дослідження, визначення об'єкта та предмета дослідження*	*лютий 2026*	*виконано*
3	*Проєктування методу розв'язання задачі, опис архітектурних рішень, розроблення математичних моделей та алгоритмів.*	*березень 2026*	*виконано*
4	*Обґрунтування інструментарію розробки, програмна реалізація розробленого методу, проведення експериментального тестування та оцінювання ефективності.*	*квітень 2026*	*виконано*
5	*Написання тексту кваліфікаційної роботи, урахування зауважень керівника, оформлення згідно з вимогами*	*травень 2026*	*виконано*
6	*Розроблення презентаційних матеріалів та попередній захист кваліфікаційної роботи*	*травень 2026*	*виконано*
7	*Отримання відгуку керівника, рецензії, перевірка тексту кваліфікаційної роботи на плагіат, нормоконтроль*	*червень 2026*	*виконано*
8	*Підготовка до захисту та захист кваліфікаційної роботи*	*червень 2026*	*виконано*

Виконавець: *студент групи КН-22-2* (Група виконавця) _____ (Підпис) *Павло ШЕВЧУК* (Ім'я, ПРІЗВИЩЕ)

Керівник: *асистент кафедри КН* (Науковий ступінь, посада) _____ (Підпис) *Валерія КЛІМЕНКО* (Ім'я, ПРІЗВИЩЕ)

Анотація

Тема кваліфікаційної роботи бакалавра: *«Метод автоматизованого виявлення суїцидальних намірів у повідомленнях користувачів засобами обробки природної мови»*

Виконавець кваліфікаційної роботи бакалавра: *студент групи КН-22-2 Павло Шевчук*

Керівник кваліфікаційної роботи бакалавра: *асистент кафедри КН Валерія Кліменко*

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
56	18	10	44	2

Мета роботи – підвищення ефективності процесу виявлення суїцидальних намірів у повідомленнях користувачів засобами обробки природної мови.

Напрямком практичного використання розробленого методу та реалізованого на його основі прототипу є автоматизована модерація повідомлень в україномовних онлайн-платформах, соціальних мережах та сервісах психологічної підтримки з метою своєчасного виявлення осіб, які потребують кризової допомоги.

Ключові слова: NLP, велика мовна модель, zero-shot класифікація, структурований вихід, виявлення суїцидальних намірів, модерація тексту.

Виконавець: *студент групи КН-22-2* _____ *Павло ШЕВЧУК*

Група виконавця — Підпис — Ім'я, ПРІЗВИЩЕ

Зміст

Перелік скорочень

Скорочення, термін, позначення	Пояснення
БД	База даних
AI	Artificial Intelligence
API	Application Programming Interface
ASGI	Asynchronous Server Gateway Interface
JSON	JavaScript Object Notation
JWT	JSON Web Token
LLaMA	Large Language Model Meta AI
LLM	Large Language Model
NLP	Natural Language Processing
REST	Representational State Transfer
SQL	Structured Query Language
СКБД	Система керування базами даних.
FN	False Negative
FP	False Positive
LPU	Language Processing Unit

Вступ

Кваліфікаційна робота бакалавра присвячена підвищенню ефективності процесу виявлення суїцидальних намірів у повідомленнях користувачів засобами обробки природної мови.

Актуальність. У сучасному цифровому середовищі соціальні мережі, месенджери та онлайн-форуми стали основними каналами комунікації для значної частини населення, особливо для молодих людей. Користувачі дедалі частіше використовують ці платформи для вираження емоційних переживань, у тому числі тих, що пов’язані з психологічними кризами.

Однією з ключових проблем сучасної онлайн-модерації є те, що ручний моніторинг великих обсягів повідомлень практично неможливий через їх кількість, а наявні автоматизовані системи здебільшого орієнтовані на виявлення мови ворожнечі, спаму чи образливого контенту, але не на лінгвістичні маркери суїцидальної поведінки. Окремою проблемою є мовна доступність: переважна більшість моделей машинного навчання та інструментів аналізу тональності розроблялися для англомовного контенту, тоді як україномовний сегмент залишається мало охопленим. Це створює прогалину між реальною потребою користувачів україномовних сервісів у безпечному цифровому середовищі та можливостями технічних рішень.

Актуальність теми зумовлена необхідністю системного підходу до забезпечення психологічної безпеки користувачів цифрових сервісів. Зі зростанням обсягу україномовного контенту та поширенням ШІ-сервісів виникає потреба в інструментах, які здатні швидко й точно ідентифікувати приховані сигнали суїцидальної поведінки в текстових повідомленнях, при цьому забезпечуючи прозорість, інтерпретованість і відповідність етичним вимогам.

Об’єкт дослідження – процес нейромрежевого виявлення суїцидальних намірів у текстовому контенті.

Предмет дослідження – методи та засоби обробки природної мови для виявлення суїцидальних намірів у повідомленнях користувачів.

Мета кваліфікаційної роботи бакалавра – підвищення ефективності процесу виявлення суїцидальних намірів у повідомленнях користувачів засобами обробки природної мови.

Завдання кваліфікаційної роботи бакалавра:

– провести аналіз предметної області автоматизованого виявлення суїцидальних намірів у текстових повідомленнях;

– розробити метод автоматизованого виявлення суїцидальних намірів у повідомленнях користувачів засобами NLP;

– реалізувати інтелектуальну систему виявлення суїцидальних намірів у повідомленнях користувачів;

– провести тестування розробленого методу з використанням розробленої інтелектуальної системи.

Розділ 1 Характеристика предметної області: аналіз моделей, методів та реалізацій

1.1 Аналіз предметної області задачі виявлення суїцидальних намірів

Самогубство є однією з найгостріших проблем громадського здоров'я XXI століття. За даними Всесвітньої організації охорони здоров'я, щороку у світі добровільно припиняють життя понад 700 тисяч осіб, а серед причин смерті серед осіб віком 15–29 років самогубство посідає одне з провідних місць [1]. Стрімка цифровізація комунікацій змінила сам характер виявлення кризових станів: якщо раніше така робота була виключно прерогативою клінічних психологів і працівників ліній довіри, то нині значна частина сигналів психоемоційного дистресу породжується безпосередньо в публічному або напівпублічному текстовому контенті – дописах у соціальних мережах, повідомленнях у месенджерах, коментарях на форумах [2].

Предметною областю задачі є текстові повідомлення користувачів україномовних онлайн-платформ – соцмереж, форумів, месенджерів та сервісів психологічної підтримки. Повідомлення можуть бути одиничними висловлюваннями або послідовностями реплік діалогу. Суттєвими є саме ті лінгвістичні характеристики, що відображають психоемоційний дистрес автора – афективна лексика, маркери ізоляції, безнадії, негативні когнітивні викривлення та прямі або непрямі вказівки на наміри припинення життя.

Задачею є автоматичне віднесення повідомлення або послідовності повідомлень до одного з рівнів психоемоційного ризику. У клінічній практиці прийнятою є градація на три рівні: низький (без ознак дистресу), помірний (виявлені маркери емоційного неблагополуччя без прямої артикуляції намірів) та високий (прямі маркери намірів, плану або щойно вчинених дій). Така градація використовується у більшості сучасних автоматизованих систем кризової модерації як компроміс між інформативністю результату та реалізованістю автоматичної класифікації.

Типові лінгвістичні маркери, на які реагують моделі виявлення суїцидальних намірів: афективні – слова з негативною валентністю, лексика безнадії; соціальної ізоляції «ніхто не розуміє», «всі залишили»; когнітивні викривлення катастрофізація, поляризоване мислення; фізичні маркери дистресу безсоння, втрата апетиту; прямі маркери – артикуляція намірів, прощальні формулювання, вказівки на способи.

Для україномовного контенту специфічними є: висока поширеність перемикання між українською, суржиком та англійською мовами; активне використання сленгу, скорочень та емодзі; регіональна варіативність лексики; відсутність відкритих україномовних клінічно валідованих датасетів з розміткою рівнів суїцидального ризику [3].

Розв'язання задачі автоматичного виявлення психоемоційних станів у текстовому контенті традиційно входить до сфери компетенції обробки природної мови (Natural Language Processing, NLP) – підгалузі комп'ютерних наук, що займається моделюванням людської мови засобами обчислювальних методів. У сучасній постановці задача класифікації емоційного стану розглядається як часткова задача класифікації коротких текстів, для якої розроблено широкий спектр методів – від класичних статистичних класифікаторів на основі векторних представлень типу TF-IDF до сучасних нейромережевих архітектур на трансформерах [4, 5].

З погляду інженерії програмних систем, задача автоматизованого виявлення суїцидальних намірів вимагає реалізації повноцінного обчислювального конвеєра, що включає прийом вхідних даних через мережеве API, валідацію формату повідомлень, обробку засобами мовної моделі, парсинг структурованої відповіді, збереження результатів у реляційній базі даних та інтеграцію з зовнішніми сервісами сповіщення. Кожен з цих компонентів формує окрему інженерну задачу з власними вимогами до продуктивності, надійності та відмовостійкості.

Сучасні дослідження у сфері виявлення суїцидальних намірів демонструють поступовий перехід від класичних методів машинного навчання,

що вимагають великих розмічених датасетів, до підходів на основі великих мовних моделей у режимі few-shot або zero-shot класифікації. Цей перехід зумовлений принциповою перевагою сучасних LLM – здатністю розв'язувати нові задачі без донавчання, лише на основі текстової інструкції, що особливо важливо для україномовного домену через відсутність розмічених датасетів у відкритому доступі [3].

З технічного боку задача формулюється як багатокласова класифікація коротких текстів з трьома цільовими класами (low, medium, high) та опціональним додатковим виходом – переліком ключових фраз і текстовим обґрунтуванням. Специфікою є асиметрична ціна помилок: помилка типу false negative (пропуск кризової ситуації) має суттєво вищу ціну порівняно з false positive (хибне спрацювання), оскільки в першому випадку йдеться про потенційну загрозу життю людини.

1.2 Огляд моделей глибокого навчання для задачі виявлення суїцидальних намірів

Розв’язання задачі спирається на сучасні моделі глибокого навчання, що працюють з природною мовою. Розглянуто сім моделей, які потенційно можуть бути використані для цієї задачі.

BERT [4] – пресемінальна енкодерна модель сімейства трансформерів з багатоголовим механізмом самоприкладання, попередньо навчена в режимі masked language modeling [5]. Перевагою є висока якість контекстно-залежних представлень. Недолік – англомовна орієнтація; для україномовної задачі необхідно використовувати mBERT з нижчою якістю та донавчанням на розміченому датасеті, який для домену суїцид-детекції відсутній.

RoBERTa – покращена версія BERT з оптимізованою процедурою попереднього навчання. На англомовних датасетах суїцид-детекції регулярно входить до переліку моделей з найвищими показниками F1. Недоліки збігаються з BERT – переважно англомовна орієнтація та необхідність донавчання.

XLM-RoBERTa – багатомовна версія RoBERTa, навчена на корпусі зі 100 мов включно з українською. Перевага – нативна підтримка україномовного контенту з якістю, помітно вищою за mBERT. Недоліки: вимагає донавчання; обмежений розмір контекстного вікна (512 токенів) ускладнює аналіз довгих діалогів.

LLaMA [6] – сімейство великих авторегресивних мовних моделей з відкритими вагами, розроблених Meta. На відміну від BERT-подібних енкодерів, LLaMA – декодерна модель. Перевагою для задачі суїцид-детекції є здатність виконувати класифікацію в режимі zero-shot або few-shot prompting без донавчання [7], що вирішує проблему відсутності україномовного датасету. LLaMA-3 версії 70b забезпечує високу якість роботи з україномовним текстом за рахунок широкого багатомовного корпусу. Недолік – висока обчислювальна вартість локального запуску, що робить необхідним використання хмарних сервісів інференсу.

Сімейство gpt-oss від OpenAI [9] – відкриті ваги великих мовних моделей з декодерною архітектурою та механізмом експертної маршрутизації Mixture-of-Experts (MoE) [10]. Представлені версії: gpt-oss-20b (20 млрд параметрів) та gpt-oss-120b (120 млрд). Перевагами є висока якість багатомовної zero-shot класифікації та підтримка примусово структурованого виходу через параметр response_format = json_object.

Mistral та Mixtral – серія великих мовних моделей з відкритими вагами, що поєднує декодерну архітектуру з механізмом групованого self-attention та (у Mixtral-варіантах) з MoE. Перевага – компактність та швидкість інференсу. Недолік – нижча якість багатомовного покриття для українського контексту.

PaLM та її наступник Gemini від Google – закриті комерційні великі мовні моделі. Перевагою є висока загальна якість. Недоліки – закритість ваг, залежність від єдиного провайдера API, відсутність гарантованого структурованого виходу у всіх версіях.

Порівняльну характеристику моделей наведено в таблиці 1.1.

Таблиця 1.1 – Порівняльна характеристика моделей глибокого навчання для задачі

Модель	**Архітек-тура**	**Українська підтримка**	**Режим застосування**	**Ключове обмеження**
BERT	Encoder	Слабка (mBERT)	Донавчання	Потребує розміченого датасету
RoBERTa	Encoder	Відсутня	Донавчання	Неприйнятна для українського
XLM-RoBERTa	Encoder	Помірна	Донавчання	Потребує датасету; контекст 512
LLaMA-70b	Decoder	Висока	Zero-shot / few-shot	Висока обчислювальна вартість
gpt-oss-20b/120b	Decoder+ MoE	Висока	Zero-shot з JSON-виходом	Залежність від хмарного провайдера
Mistral / Mixtral	Decoder (+MoE)	Помірна	Zero-shot / few-shot	Слабше багатомовне покриття
PaLM / Gemini	Decoder (закриті)	Висока	Zero-shot через API	Закриті ваги, єдиний провайдер

Отже, для україномовної задачі за відсутності розміченого датасету найбільш доцільним є використання великих мовних моделей у режимі zero-shot класифікації. Оптимальним є LLaMA-70b як дефолтна модель з високою якістю україномовного покриття при прийнятній латентності, доповнена альтернативними моделями gpt-oss (20b, 120b) для дослідницького режиму порівняння. Усі моделі надаються через єдиний API Groq Cloud [11, 12].

1.3 Огляд наукових публікацій з тематики автоматизованого виявлення суїцидальних намірів

Тематика автоматизованого виявлення суїцидальних намірів у текстовому контенті активно розвивається з другої половини 2010-х років і набула

актуальності з поширенням великих мовних моделей у 2022–2024 роках. У роботі [13] представлено систематичний огляд методів машинного навчання для виявлення суїцидальних намірів. Автори класифікують підходи за поколіннями моделей – від класичних методів на TF-IDF до сучасних трансформерних архітектур – та фіксують зростання якості з кожним поколінням. Для BERT та RoBERTa на стандартних англомовних датасетах (Reddit SuicideWatch, CLPsych Shared Task) типові показники F1 у бінарній постановці перевищують 0,90, але результати погіршуються при перенесенні на нові домени та мови.

У роботі [14] представлено дослідження детекції суїцидальної ідеації в соцмережах із застосуванням комбінованих підходів глибинного навчання та NLP. Підтверджено, що моделі трансформерів суттєво переважають класичні методи у розпізнаванні прихованих маркерів дистресу.

У дослідженні [15] зосереджено увагу на ідентифікації суїцидального ризику в соцмережах із різними типами векторних представлень. Запропоновано порівняння BERT-ембедингів з ансамблевими методами агрегації векторів речень для коротких повідомлень.

Систематичний огляд застосування великих мовних моделей у сфері психічного здоров’я представлено у роботі [16]. Автори фіксують зростання частки досліджень, що використовують zero-shot та few-shot prompting замість traditional fine-tuning, і пов’язують це з труднощами збору розмічених даних у чутливій предметній області. Великі мовні моделі демонструють якість класифікації межових випадків, порівнянну зі спеціалізованими моделями, та забезпечують додаткову перевагу – природне формулювання обґрунтування.

У роботі [17] представлено спеціалізовану адаптацію LLaMA-моделі під задачі інтерпретованого аналізу психічного здоров’я у соцмережах. Описано підхід донавчання на широкому наборі суміжних задач (депресія, тривожні розлади, ПТСР, суїцидальний ризик) з обов’язковим формуванням людиночитного обґрунтування.

Емпіричну оцінку моделей ChatGPT у задачах NLP, пов’язаних з психічним здоров’ям, представлено у роботі [18]. Автори показують, що ChatGPT у zero-shot досягає показників якості, порівнянних зі спеціалізованими

fine-tuned моделями, при значно нижчих витратах на впровадження. Зазначено схильність моделі до «обережного» патерну на сенситивних запитах, що може спричиняти зміщення в бік false negative.

У роботі [19] представлено систематичну оцінку великих мовних моделей у задачах прогнозування психічного здоров’я на основі онлайн-текстів. Автори показують, що сучасні відкриті моделі LLaMA та gpt-oss у zero-shot забезпечують показники, які наближаються до закритих комерційних моделей, при цьому дозволяючи локальне розгортання та повний контроль над пайплайном.

Узагальнюючи, найбільш перспективним напрямом для україномовної задачі є застосування великих мовних моделей у режимі zero-shot класифікації зі структурованим виходом. Цей підхід забезпечує якість, порівнянну з традиційним донавчанням трансформерів, без потреби у розміченому датасеті. Україномовний сегмент задачі залишається малодослідженим, так само як практичні аспекти побудови end-to-end систем кризової модерації з програмними інтерфейсами інтеграції [3].

1.4 Аналіз існуючих програмних засобів автоматизованої модерації

З метою визначення поточного рівня розвитку технологій у цій сфері, а також виявлення їхніх архітектурних та функціональних обмежень, було проведено порівняльний аналіз сучасного програмного забезпечення. Нижче розглянуто чотири найбільш репрезентативні програмні засоби, що реалізують функцію автоматизованого виявлення кризових станів, деструктивних маніпулятивних стратегій або шкідливого контенту в текстових повідомленнях. Розглянемо чотири найбільш репрезентативні програмні засоби, що реалізують функцію автоматизованого виявлення кризових станів або шкідливого контенту в текстових повідомленнях.

Crisis Text Line [20] – некомерційна служба психологічної підтримки з власною алгоритмічною системою пріоритизації звернень. Обробляє вхідні повідомлення в реальному часі, оцінює рівень ризику за лінгвістичними маркерами і просуває звернення з найвищою ймовірністю кризи в верх черги

волонтерів-консультантів. Сильна сторона – інтеграція в реальний робочий процес. Слабка – закритість, відсутність україномовної версії, непридатність для звичайних онлайн-платформ.

OpenAI Moderation API [21] – комерційний сервіс модерації контенту з класифікатором за категоріями (мова ворожнечі, насильство, самопошкодження). Перевага – зрілість сервісу та інтеграція з екосистемою OpenAI. Недоліки: категорія «self-harm» – бінарна оцінка без диференціації рівня; відсутнє текстове обґрунтування; нижча якість на україномовному контенті.

Perspective API [22] – безкоштовний сервіс Google Jigsaw для оцінки токсичності коментарів. Перевага – простота інтеграції. Недолік: переважно бінарна постановка, категорії самопошкодження представлені узагальнено; українська підтримується частково.

Azure AI Content Safety [23] – комерційний сервіс модерації від Microsoft з аналізом за чотирма категоріями небезпечного контенту з рівнями 0–7. Перевага – гранулярна шкала. Недоліки: категорія самопошкодження не диференціює суїцидальні наміри; немає текстового обґрунтування; залежність від екосистеми Azure. Порівняння цих рішень за ключовими характеристиками наведено в таблиці 1.2.

Таблиця 1.2 – Порівняльний аналіз існуючих програмних засобів модерації

Засіб	**Доступ**	**Гранулярність ризику**	**Інтерпретованість**	**Українська мова**
Crisis Text Line	Закритий	Висока (внутр.)	Висока (оператору)	Відсутня
OpenAI Moderation API	Публічний API	Бінарна	Відсутня	Часткова
Perspective API	Публічний API	Бінарна	Відсутня	Часткова
Azure AI Content Safety	Публічний API	8 рівнів	Низька	Часткова
Розроблюваний прототип	Open-source	3 рівні + score	Висока (key_phrases + reasoning)	Повна

Жоден з існуючих засобів не задовольняє одночасно чотирьом ключовим вимогам для модерації – повноцінній підтримці багатомовності, диференціації рівня ризику з гранулярною шкалою, інтерпретованості рішень та відкритій архітектурі. Це обґрунтовує доцільність розробки власного рішення на основі великої мовної моделі у zero-shot режимі з примусово структурованим JSON-виходом, що містить як числову оцінку, дискретну мітку, так і ключові фрази та обґрунтування.

1.5 Мета та завдання дослідження

У результаті проведеного аналізу предметної області, огляду моделей глибокого навчання та існуючих програмних засобів встановлено, що задача автоматизованого виявлення суїцидальних намірів у україномовних текстових повідомленнях є актуальною та практично значущою, але не має задовільного рішення серед наявних інструментів. Натомість сучасні великі мовні моделі з відкритими вагами у режимі zero-shot класифікації з примусово структурованим виходом дозволяють побудувати таке рішення без розміченого датасету та трудомісткого донавчання.

Мета роботи – підвищення ефективності процесу виявлення суїцидальних намірів у повідомленнях користувачів засобами обробки природної мови.

Для досягнення поставленої мети необхідно вирішити такі завдання:

– провести аналіз предметної області автоматизованого виявлення суїцидальних намірів у текстових повідомленнях;

– розробити метод автоматизованого виявлення суїцидальних намірів у повідомленнях користувачів засобами NLP;

– реалізувати інтелектуальну систему виявлення суїцидальних намірів у повідомленнях користувачів;

– провести тестування розробленого методу з використанням розробленої інтелектуальної системи.

Розділ 2 Метод автоматизованого виявлення суїцидальних намірів у повідомленнях користувачів засобами обробки природної мови

2.1 Формалізація предметної області задачі виявлення суїцидальних намірів

Для коректної постановки задачі необхідно ввести формальний апарат, що описує вхідні дані, очікувані виходи моделі та процес перетворення вхідної інформації у вихідну.

Вхідним об'єктом задачі є послідовність текстових повідомлень діалогу, що утворюють контекст комунікації одного користувача:

$$M = \{m_1, m_2, \ldots, m_n\}, n \geq 1, \quad (2.1)$$

де $m_i = (r_i, c_i)$ – окреме повідомлення, $r_i \in \{user, assistant, system\}$ – роль автора, $c_i \in \Sigma^*$ – текстовий зміст. Така структура дозволяє моделювати як одиничні висловлювання (n = 1), так і повноцінні діалогові обміни.

Вихідним об'єктом задачі є кортеж класифікаційного рішення:

$$R = (s, l, K, \rho), \quad (2.2)$$

де $s \in [0, 1]$ – probability_score; $l \in L = \{low, medium, high\}$ – дискретна мітка рівня ризику; K – множина ключових фраз; $p \in \Sigma^*$ – текстове обґрунтування (reasoning).

Метою задачі є побудова відображення $F: M \rightarrow R$, що для довільної припустимої послідовності повертає коректне класифікаційне рішення. Коректність оцінюється за метриками з п. 2.5.

На послідовність M накладаються обмеження: $|M| \leq 50$ (розмір контекстного вікна моделі); $\sum|c_i| \leq 32000$ (обмеження API провайдера); $\forall i: |c_i| \geq 1;$ мова повідомлень – переважно українська з допустимими вкрапленнями інших мов.

Для розв'язання задачі задіюється параметризована функція класифікації, реалізована великою мовною моделлю (LLM):

$$r = \mathrm{f}_{\mathrm{LLM}}(P;\theta), \theta \in \Theta, \quad (2.3)$$

де P – вхідний промпт; θ – фіксований набір параметрів моделі (ваги нейронної мережі), отриманих у результаті попереднього навчання на корпусі багатомовних текстів; $r \in \Sigma^*$ – вихідний текст моделі у форматі валідного JSON за рахунок використання режиму structured output.

Функція формування промпту φ_prompt будує запит до моделі:

$$P = \varphi_{\mathrm{prompt}}(M, S) = S \oplus M, \quad (2.4)$$

де $S \in \Sigma^*$ – системний промпт із фіксованою інструкцією та JSON-схемою очікуваної відповіді; $\oplus$ – операція впорядкованої конкатенації повідомлень за полем role.

Функція парсингу відповіді *φ_parse* перетворює JSON-текст моделі у внутрішню структуру:

$$(s, K, \rho) = \varphi_{\mathrm{parse}}(r) = \pi_{\mathrm{score,keys,reasoning}}\big(\mathrm{json.loads}(r)\big). \quad (2.5)$$

Дискретна мітка виводиться з числової оцінки через функцію *δ:* [0, 1] → L:

$$l = \delta(s) = \begin{cases} \mathrm{high}, s \geq \alpha_2 \\ \mathrm{medium}, \alpha_1 \leq s < \alpha_2 \\ \mathrm{low}, s < \alpha_1 \end{cases} \quad (2.6)$$

з порогами $\alpha_1 = 0{,}4$ та $\alpha_2 = 0{,}7$. Обґрунтування цих значень наведено в п. 2.3.

Сукупно метод реалізується композицією:

$$F(M) = (s, \delta(s), K, \rho), \text{де } (s, K, \rho) = \varphi_{\mathrm{parse}}\Big(\mathrm{f}_{\mathrm{LLM}}\big(\varphi_{\mathrm{prompt}}(M, S); \theta\big)\Big) \quad (2.7)$$

Особливістю запропонованої формалізації є те, що параметри моделі θ є фіксованими і не підлягають оптимізації – модель використовується у режимі zero-shot без донавчання. Структурований JSON-вихід гарантовано забезпечується API провайдера через примусове обмеження response_format = json_object, що повністю усуває потребу в ймовірнісному парсингу вільнотекстової відповіді.

2.2 Опис архітектури моделі

Функція f_LLM реалізується великою мовною моделлю з відкритими вагами сімейства gpt-oss та llama-70b, доступ до якої надається через Groq Cloud [25]. Моделі належать до класу авторегресивних трансформерних мовних моделей із декодерною архітектурою (decoder-only Transformer) та доповнені механізмом експертної маршрутизації Mixture-of-Experts [24]. Прикладом сучасної моделі з механізмом MoE поза сімейством gpt-oss є Mixtral [25], що використовує аналогічний підхід до маршрутизації токенів між кількома експертами.

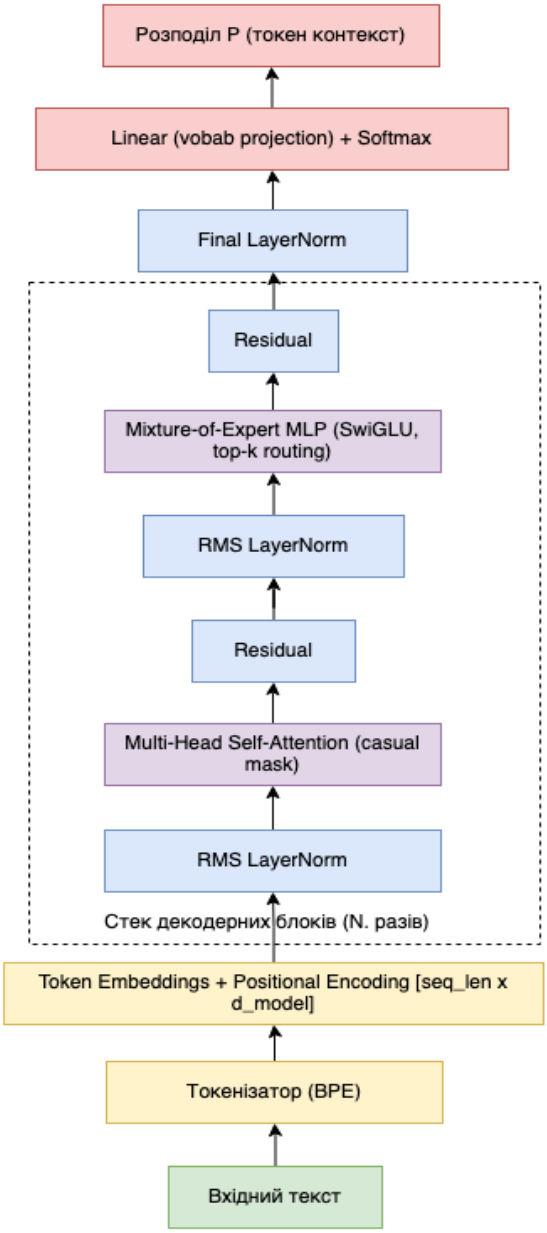

Рисунок 2.1 – Узагальнена архітектура моделі gpt-oss

Основу архітектури складає стек з N однотипних декодерних блоків (для gpt-oss-20b N = 24, для gpt-oss-120b N = 36). Кожен блок включає підблок багатоголового самоприкладання з причинною маскою та підблок експертної мережі прямого поширення. Схему архітектури наведено на рисунку 2.1.

Вхідний текст проходить через токенізацію Byte-Pair Encoding (BPE), що задає відображення $\tau: \Sigma^* \rightarrow V^*$, де V – словник токенів (близько 200 тисяч позицій). Кожному токену зіставляється ембединг розмірності d_model (для gpt-oss-20b d_model = 2880):

$$E_i = W_{\text{emb}}[t_i] + \text{PE}_i, i = 1, 2, \ldots, L, \tag{2.8}$$

де W_emb – навчена матриця ембедингів; PE_i – позиційне кодування. У gpt-oss застосовується обертальне позиційне кодування RoPE [26].

Центральним блоком декодера є механізм багатоголового самоприкладання. Для матриці прихованих станів H обчислюються проекції запитів, ключів та значень:

$$Q = H \cdot W^Q,\ K = H \cdot W^K,\ V = H \cdot W^V, \tag{2.9}$$

де W^Q, W^K, W^V – навчені вагові матриці. Операція уваги виконується за класичною формулою [3]:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{Q \cdot K^T}{\sqrt{d_k}} + M_{\text{causal}}\right) \cdot V, \tag{2.10}$$

де *d_k = d_model / h* – розмірність окремої голови; h – кількість голів; M_causal – нижньотрикутна маска, що забороняє позиції i звертатися до позицій j > i. Багатоголовий механізм виконує h таких операцій паралельно:

$$\text{MultiHead}(H) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h) \cdot W^O. \tag{2.11}$$

Після блоку уваги застосовується підблок Mixture-of-Experts MLP. Замість єдиної мережі прямого поширення використовується ансамбль експертів E_1, …, E_K, серед яких для кожного токена обирається підмножина top-k активних:

$$\text{MoE}(x) = \sum_{e \in \text{TopK}(\text{G}(x))} \text{G}(x)_e \cdot E_e(x), \tag{2.12}$$

де G(x) – вектор ваг експертів, обчислений gating-мережею; кожен експерт реалізується як MLP з активацією SwiGLU. Підхід дозволяє наростити ємність моделі без пропорційного зростання обчислювальних витрат на інференс.

Навколо обох підблоків застосовуються залишкові з'єднання та нормалізація RMS LayerNorm [27][28]:

$$h' = h + \mathrm{MultiHead}\big(\mathrm{RMSNorm}(h)\big);\ h'' = h' + \mathrm{MoE}\big(\mathrm{RMSNorm}(h')\big). \quad (2.13)$$

Після проходження всіх N блоків та фінальної нормалізації прихований стан останнього токена проектується на словник через лінійний шар та softmax:

$$P(t_{L+1}|t_1, \dots, t_L) = \mathrm{softmax}(W_{\mathrm{lm}} \cdot h_L^N). \quad (2.14)$$

Параметри моделі θ є результатом попереднього навчання на корпусі понад трильйона токенів з цільовою функцією крос-ентропійної втрати:

$$L_{\mathrm{pretrain}}(\theta) = -\sum_t log\ P(t_{t+1}|t_{\le t};\ \theta). \quad (2.15)$$

У межах поточної роботи параметри θ не оптимізуються – модель використовується у чистому zero-shot режимі. Завдяки інфраструктурі, спеціалізованій під інференс трансформерних моделей [26], час відгуку моделі залишається в межах, прийнятних для інтерактивної модерації.

2.3 Кроки методу автоматизованого виявлення суїцидальних намірів

На основі формальної постановки задачі та опису архітектури моделі розроблено метод автоматизованого виявлення суїцидальних намірів. Метод реалізує перетворення $F: M \rightarrow R$ (2.7) у вигляді п'ятикрокового пайплайну. Логіку методу представлено у вигляді математичного псевдокоду в Алгоритмі 2.1.

Алгоритм 2.1 – Метод автоматизованого виявлення суїцидальних намірів

```
Вхід:    M = {m_1, ..., m_n} – послідовність повідомлень
         θ – фіксовані параметри моделі f_LLM
         S – системний промпт із JSON-схемою
         α_1 = 0.4, α_2 = 0.7 – пороги дискретизації
Вихід:   R = (s, l, K, ρ) – класифікаційне рішення

 1.  V(M) ← валідація_структури(M)            // формула (2.1)
 2.  if V(M) = false then return HTTP 422
 3.  P ← ϕ_prompt(M, S) = S ⊕ M               // формула (2.4)
 4.  r ← f_LLM(P; θ)                          // формула (2.3)
 5.  if r = ⊥ then R ← (0, error, ∅, 'API failure')
 6.  (s, K, ρ) ← ϕ_parse(r)                   // формула (2.5)
 7.  l ← δ(s):
         high     if s ≥ α_2
         medium   if α_1 ≤ s < α_2            // формула (2.6)
         low      if s < α_1
```

```
 8.  R ← (s, l, K, ρ)
 9.  записати R до бази даних (moderation_logs)
10.  return R
```

Графічне відображення пайплайну з посиланнями на формули наведено на рисунку 2.2.

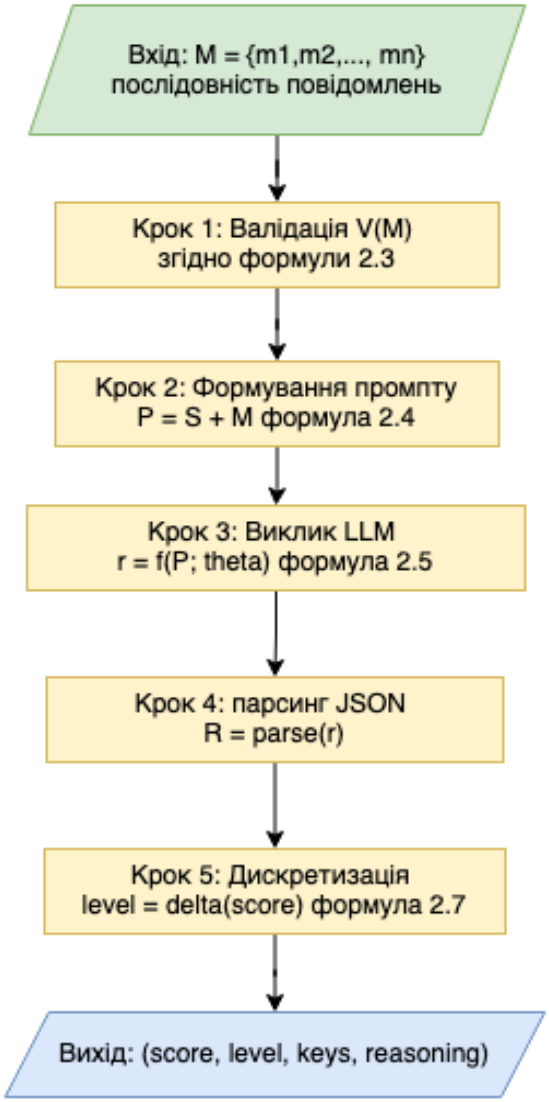

Рисунок 2.2 – Схема пайплайну методу

Крок 1 – валідація вхідної послідовності за обмеженнями з (2.1). У разі невиконання умов послідовність відхиляється, виклик моделі не виконується.

Крок 2 – формування промпту згідно функції φ_prompt (2.4). Системний промпт S містить три компоненти: формулювання ролі моделі як експерта з лінгвістичного аналізу психоемоційного стану; JSON-схему очікуваної відповіді; правила прийняття класифікаційних рішень. Метод реалізує чистий zero-shot

підхід – у промпт не включаються жодні ілюстративні приклади [29]. Сучасні великі мовні моделі демонструють властивості zero-shot reasoning без явних прикладів, що дозволяє розв'язувати складні задачі класифікації лише на основі текстової інструкції [30].

Крок 3 – виклик моделі f_LLM (2.3). Виклик виконується через HTTP-API Groq Cloud з параметром response_format = json_object, що примусово обмежує вихід форматом валідного JSON. Це усуває потребу в ймовірнісному парсингу. Дефолтною моделлю обрано llama-70b як таку, що демонструє найкращий баланс між якістю класифікації та латентністю.

Крок 4 – парсинг JSON-відповіді (2.5). Оскільки модель повертає валідний JSON, парсинг зводиться до виклику json.loads з подальшою проєкцією на цільові поля – probability_score, key_phrases, reasoning. Поля K та ρ є складовими структурованого виходу самої моделі.

Крок 5 – дискретизація s у мітку l (2.6). Пороги $\alpha_1 = 0{,}4$ та $\alpha_2 = 0{,}7$ є нерівновіддаленими: інтервал «low» становить [0; 0,4), «medium» – [0,4; 0,7), «high» – [0,7; 1]. Асиметрія обумовлена різною ціною помилок: false negative має суттєво вищу ціну. Тому нижній поріг навмисно зміщений вниз – повідомлення з probability_score у проміжному діапазоні класифікується як «medium» і виноситься на додатковий розгляд модератора.

Окремо передбачено технічний статус «error» для випадків збою API (таймаут, перевищення квоти, некоректна відповідь).

Після основних кроків виконується технічна операція збереження R до таблиці moderation_logs реляційної бази даних, що забезпечує повний аудиторський слід і можливість зворотного зв'язку модератора через мітки human_label.

2.4 Опис тестового набору повідомлень для оцінювання методу

Оцінювання якості методу вимагає тестового набору з еталонною розміткою. Принциповим обмеженням теми є відсутність на момент виконання

роботи відкритих україномовних клінічно валідованих датасетів з розміткою рівнів суїцидального ризику [44]. Англомовні датасети (Reddit SuicideWatch, CLPsych Shared Task) не можуть бути використані безпосередньо через мовну невідповідність.

Для оцінювання методу сформовано власний експертно розмічений тестовий набір з 60 україномовних повідомлень, розподілених порівну за трьома класами (по 20 на клас). Розмір обрано як компроміс між статистичною репрезентативністю та практичною можливістю якісної ручної розмітки. Загальну характеристику набору наведено в таблиці 2.1.

Таблиця 2.1 – Загальна характеристика тестового набору

Клас	**Кількість, шт.**	**Середня довжина, симв.**	**Стандартне відхилення, симв.**	**Частка, %**
low	20	78	32	33,3
medium	20	112	41	33,3
high	20	94	37	33,3
Усього	60	95	39	100,0

Розподіл за класами є рівномірним, що уникає проблеми незбалансованості та дозволяє коректно інтерпретувати метрики precision, recall та F1. Середня довжина 95 символів типова для коротких повідомлень у соцмережах. Дещо вища середня довжина для класу medium (112 символів) пояснюється тим, що повідомлення цього класу містять змішані емоційні маркери.

Тематичний розподіл повідомлень охоплює основні предметні області, у яких типово виникають сигнали дистресу, і наведений у таблиці 2.2.

Розмітка повідомлень здійснювалася з опором на лінгвістичні маркери з фахової літератури з клінічної психології. Для класу low характерні позитивні афективні маркери, конкретні плани, згадки соціальної підтримки. Для medium –

маркери соціальної ізоляції, втрати мотивації, когнітивних викривлень, фізичних симптомів дистресу без прямої артикуляції намірів.

Таблиця 2.2 – Тематичний розподіл повідомлень тестового набору

Тематична група	**Low**	**Medium**	**High**	**Усього**
Соціальні відносини	5	6	5	16
Академічно-професійна сфера	5	6	4	15
Повсякденні переживання	7	5	3	15
Пряма артикуляція станів	3	3	8	14
Усього	20	20	20	60

Для high – пряма артикуляція суїцидальних намірів, прощальні формулювання, вказівки на конкретні плани або засоби. Приклади повідомлень кожного класу наведено в таблиці 2.3.

Таблиця 2.3 – Приклади повідомлень тестового набору

Клас	**Текст повідомлення**	**Розмітка**
low	«Сьогодні був чудовий день, ходив гуляти з друзями, отримав листа з університету»	low
low	«Зустріч пройшла нормально, дякую за підтримку, до завтра»	low
medium	«Вже тиждень не виходжу з кімнати, не бачу сенсу нічого робити, всі мене залишили»	medium
medium	«Не сплю третю ніч, навіщо все це, ніхто навіть не помітить»	medium
high	«Я вирішив, що сьогодні останній день. Прощавайте, хто мене любив»	high
high	«Все, я не можу більше. Завтра вже не побачимось. Дякую за все»	high

Для забезпечення достовірності розмітки межові випадки вирішувалися на користь більш консервативної, вищої категорії з огляду на асиметрію ціни помилок. Для оцінки інтер-розмічувальної узгодженості було проведено повторну розмітку через 7 днів; рівень узгодженості за коефіцієнтом Cohen's kappa [31] склав 0,89, що відповідає «substantial agreement» за шкалою Лендіса–Коха [32].

Обмеження тестового набору: розмір у 60 повідомлень малий за стандартами сучасних NLP-досліджень; експертна розмітка одним автором не замінює клінічну; не охоплено усю варіативність повідомлень. Результати слід інтерпретувати як попередню валідацію методу, що потребує підтвердження на ширшому клінічно валідованому наборі.

2.5 Метрики оцінювання якості виявлення суїцидальних намірів

Оцінювання якості методу виконується з використанням стандартних метрик задач багатокласової класифікації [33]. Базовим інструментом є матриця плутанини $C \in \mathbb{N}^{|L|\times|L|}$:

$$C_{i,j} = \left|\left\{x \in X_{\text{test}} : y_{\text{true}}(x) = i, y_{\text{pred}}(x) = j\right\}\right|. \quad (2.16)$$

Діагональні елементи $C_{i,i}$ відповідають правильним передбаченням, недіагональні – помилкам. На основі матриці для кожного класу i обчислюються:

$$\text{TP}_i = C_{i,i};\ \text{FP}_i = \sum_{j\neq i} C_{j,i};\ \text{FN}_i = \sum_{j\neq i} C_{i,j}. \quad (2.17)$$

Точність класу i – частка правильних передбачень серед усіх віднесених методом до даного класу:

$$\text{Precision}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i}. \quad (2.18)$$

Повнота класу i – частка правильно виявлених повідомлень даного класу:

$$\text{Recall}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i}. \quad (2.19)$$

F1-міра – гармонійне середнє точності та повноти:

$$\text{F1}_i = \frac{2\cdot \text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}. \quad (2.20)$$

Особливо важливою для задачі є Recall_high, що характеризує здатність методу не пропускати реальні кризові випадки. Інтегральні метрики на тестовому наборі:

$$\text{Accuracy} = \frac{\sum_{i \in L} \text{TP}_i}{|X_{\text{test}}|}. \quad (2.21)$$

$$\text{Macro} - \text{F1} = \frac{1}{|L|} \cdot \sum_{i \in L} \text{F1}_i. \quad (2.22)$$

$$\text{Weighted} - \text{F1} = \sum_{i \in L} \frac{n_i}{|X_{\text{test}}|} \cdot \text{F1}_i. \quad (2.22)$$

Для збалансованого набору значення *macro-F1* та *weighted-F1* збігаються. Для оцінки бінарного розрізнення між нейтральними повідомленнями (low) та повідомленнями з ознаками ризику (medium ∪ high) застосовується:

$$\text{Binary} - \text{Accuracy} = \frac{\text{TP}_{\text{low}} + \text{TP}_{med \cup high}}{|X_{\text{test}}|}. \quad (2.24)$$

Метрика важлива з прикладної точки зору, оскільки рішення про ескалацію модератору приймається саме на основі бінарного розрізнення. Для повноти наведено базову цільову функцію моделі під час її попереднього навчання – крос-ентропійну втрату:

$$L_{\text{CE}}(\theta) = -\frac{1}{|T|} \cdot \sum_{t \in T} log\ P(t_{t+1}|t_{\leq t};\ \theta). \quad (2.25)$$

У межах роботи параметри θ не оптимізуються, формула наведена з метою повноти опису моделі. Окремо для оцінки якості probability_score застосовується метрика Expected Calibration Error (ECE), що характеризує відповідність між значенням probability_score, повернутим моделлю, та фактичною ймовірністю належності повідомлення до класу high:

$$\text{ECE} = \frac{1}{N} \cdot \sum_{k=1}^{K} n_k \cdot |\text{freq}_k - \text{conf}_k|, \quad (2.26)$$

де N – загальна кількість передбачень; K – кількість бінів (типово K = 10); n_k – кількість передбачень у k-му біні; $freq_k$ – фактична частка повідомлень класу high у біні; $conf_k$ – середнє значення probability_score у біні. Прийнятним вважається ECE < 0,1.

2.6 Сценарії проведення експериментальних досліджень

Для оцінювання методу спроєктовано шість комплементарних сценаріїв експериментальних досліджень.

Сценарій 1 – основний експеримент з оцінювання якості класифікації. Тестовий набір прогнано через метод з моделлю llama-70b. Для статистичної надійності виконано чотири незалежні прогони з фіксованими параметрами моделі. Для кожної метрики обчислюються:

$$\mu_m = \frac{1}{4} \cdot \sum_{k=1}^{4} m_k ; \ \sigma_m = \sqrt{\frac{1}{3} \cdot \sum_{k=1}^{4} (m_k - \mu_m)^2}, \quad (2.27)$$

де m_k – значення метрики на k-му прогоні; коефіцієнт 1/3 – корекція Бесселя для незміщеної оцінки. Результати наводяться у форматі μ ± σ.

Сценарій 2 – порівняння трьох моделей (llama-70b, gpt-oss-20b, gpt-oss-120b) на тестовому наборі. Для кожної моделі обчислюються ті самі метрики та середня латентність інференсу.

Сценарій 3 – якісний аналіз помилок методу. Виокремлюються повідомлення з неправильною класифікацією та групуються за типом помилки (FN_{high}, FP_{high}, плутанина між сусідніми класами). Для кожної групи аналізуються лінгвістичні особливості та формулюються рекомендації.

Сценарій 4 – оцінка чутливості до температури моделі. Виконується прогін з п’ятьма значеннями *temperature* $\in$ *{0; 0,2; 0,5; 0,7; 1,0}* для обґрунтування оптимального значення.

Сценарій 5 – оцінка чутливості до варіацій системного промпту. Порівнюються три варіанти: мінімалістичний (тільки JSON-схема), розгорнутий (з визначеннями маркерів), з прикладами.

Сценарій 6 – аналіз компромісу між якістю та латентністю. На основі результатів сценарію 2 будується Pareto-діаграма у просторі «латентність – Macro-F1».

Розподіл тестового набору не передбачає поділу на тренувальну та тестову вибірки, оскільки метод використовує модель у zero-shot режимі без

донавчання – фаза навчання моделі на тестовому наборі відсутня. Усі 60 повідомлень виступають як єдиний held-out evaluation set.

2.7 Висновки до розділу 2

У другому розділі здійснено формалізацію предметної області задачі автоматизованого виявлення суїцидальних намірів. Формалізацію виконано у вигляді відображення $F: M \rightarrow R$ з чітким описом структури вхідних повідомлень (2.1) та кортежу класифікаційного рішення (2.2). Це дозволило ввести єдину математичну нотацію для подальшого опису методу та забезпечило можливість строгого формулювання сценаріїв експериментальних досліджень.

Розроблено опис архітектури моделі, що використовується методом. Описано декодерну трансформерну архітектуру з механізмом експертної маршрутизації Mixture-of-Experts, на якій базується сімейство моделей gpt-oss та llama-70b. Опис охоплює всі ключові компоненти – токенізацію Byte-Pair Encoding, ембединги з обертальним позиційним кодуванням RoPE, багатоголовий механізм самоприкладання з причинною маскою, експертні мережі прямого поширення зі стробуванням top-k та авторегресивну генерацію вихідної послідовності (2.8–2.15). Це забезпечило прозоре обґрунтування властивостей моделі як обчислювальної основи методу.

Розроблено покроковий метод автоматизованого виявлення суїцидальних намірів. Метод представлено у вигляді математичного псевдокоду (Алгоритм 2.1) та схеми пайплайну з п'яти кроків: валідація вхідної послідовності, формування zero-shot системного промпту з JSON-схемою, виклик великої мовної моделі через хмарний сервіс інференсу з примусово структурованим виходом, парсинг JSON-відповіді та дискретизація числової оцінки у дискретну мітку рівня ризику з нерівновіддаленими порогами $\alpha_1 = 0{,}4$ та $\alpha_2 = 0{,}7$. Завдяки використанню примусово структурованого JSON-виходу усунуто потребу в ймовірнісному парсингу вільнотекстових відповідей моделі, що дозволило підвищити надійність обчислювального конвеєра.

Сформовано тестовий набір для оцінювання якості методу. Набір складається з 60 україномовних повідомлень, рівномірно розподілених за трьома цільовими класами ризику (по 20 повідомлень на клас). Розмітку здійснено автором з опертям на лінгвістичні маркери з фахової літератури з клінічної психології; рівень інтер-розмічувальної узгодженості за коефіцієнтом Cohen's kappa склав 0,89, що відповідає рівню «substantial agreement» за шкалою Лендіса–Коха. Це дозволило отримати збалансовану тестову вибірку, придатну для коректного обчислення метрик класифікації.

Визначено набір з 11 метрик оцінювання якості методу. Описано матрицю плутанини, по-класові метрики precision, recall та F1, інтегральні метрики accuracy, macro-F1, weighted-F1, прикладну метрику binary-accuracy для бінарного розрізнення «потребує перегляду / не потребує перегляду» та метрику Expected Calibration Error для оцінки калібрування числової оцінки probability_score (2.16–2.26). Завдяки гранулярному набору метрик забезпечено можливість всебічної оцінки методу як на рівні окремих класів, так і інтегрально, з урахуванням асиметричної ціни помилок у задачі суїцид-детекції.

Спроєктовано шість комплементарних сценаріїв експериментальних досліджень – основний експеримент з чотирма незалежними прогонами, порівняння трьох моделей, якісний аналіз помилок, оцінка калібрування, чутливість до параметра temperature та Pareto-аналіз компромісу «якість–латентність». Це дозволило забезпечити повноту експериментальної валідації методу, що враховує специфіку zero-shot режиму та заміщає класичні елементи fine-tuning-сценарію (графіки функції втрат, ROC-крива, train/test split) аналогами, релевантними для застосовуваного підходу.

Розділ 3 Експериментальне дослідження методу

3.1 Опис експериментального застосунку для проведення досліджень

Для практичної перевірки методу реалізовано експериментальний прототип у вигляді клієнт-серверного веб-застосунку. Прототип призначений для емпіричної валідації методу та проведення сценаріїв з п. 2.6. Вибір архітектури клієнт-сервер зумовлений необхідністю чіткого розділення обчислювальної логіки бекенду та інтерфейсу користувача, що дозволяє забезпечити високу масштабованість, модульність та незалежне тестування компонентів системи.

Серверну частину реалізовано на Python 3.11 на основі асинхронного веб-фреймворку FastAPI [34]. Використання FastAPI забезпечує високу продуктивність обробки мережевих запитів завдяки нативній підтримці асинхронності та можливості автоматичної генерації інтерактивної документації OpenAPI. Для збереження результатів класифікації та конфігураційних даних використано компактну реляційну СКБД SQLite [35] з асинхронним драйвером aiosqlite та об'єктно-реляційним відображенням (ORM) SQLAlchemy [36]. Такий підхід мінімізує накладні витрати на розгортання прототипу, зберігаючи при цьому можливість безшовної міграції на більш потужні серверні СКБД у разі промислового впровадження. Валідація даних здійснюється через Pydantic-моделі [37], що гарантує строгу типізацію вхідних та вихідних структур даних і знижує ризик виникнення помилок під час виконання програмного коду.

Вихідні HTTP-запити – як до Groq Cloud, так і до webhook-адрес – виконуються через бібліотеку httpx [38] у асинхронному режимі. Клієнтську частину реалізовано на стеку React [39] + Vite [40]. Автентифікація реалізована через JWT [41] (для веб-інтерфейсу) та X-API-Key (для зовнішніх систем), хешування паролів – через passlib [42].

Загальну логіку обробки запиту до endpoint-а /api/v1/analyze унаочнює блок-схема на рисунку 3.1.

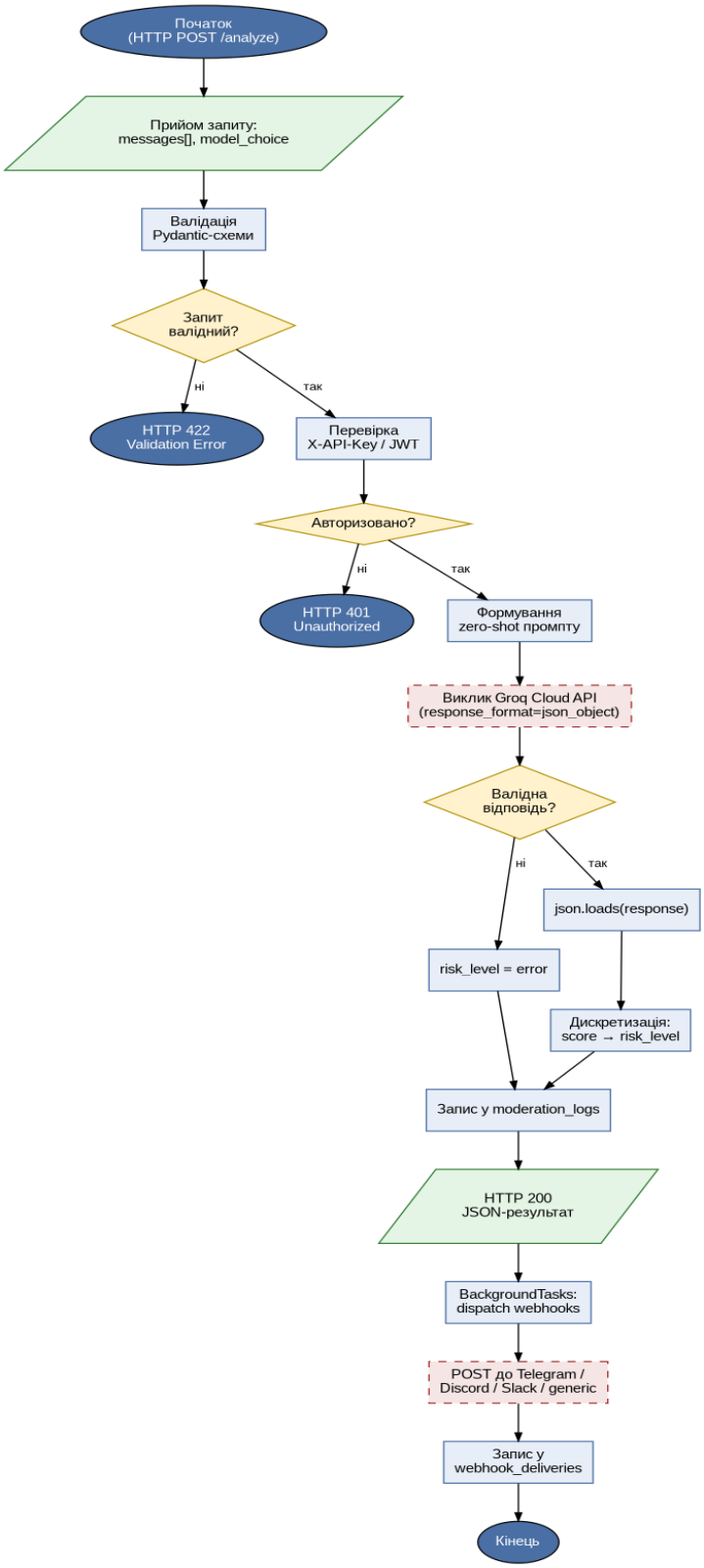

Рисунок 3.1 – Блок-схема обробки запиту до endpoint-а /api/v1/analyze

Програмна логіка серверної частини структурована на чотири шари: Pydantic-моделі контрактів API (AnalysisRequest, AnalysisResponse, Message); сервісні класи прикладної логіки (Analyzer – реалізує етапи 2–5 методу; WebhookDispatcher – виконує фонову доставку сповіщень; AuthService –

інкапсулює автентифікацію); маршрутизатори FastAPI для груп endpoint-ів (/api/v1/analyze, /logs, /auth); ORM-моделі для роботи з базою даних.

Діаграму класів представлено на рисунку 3.2.

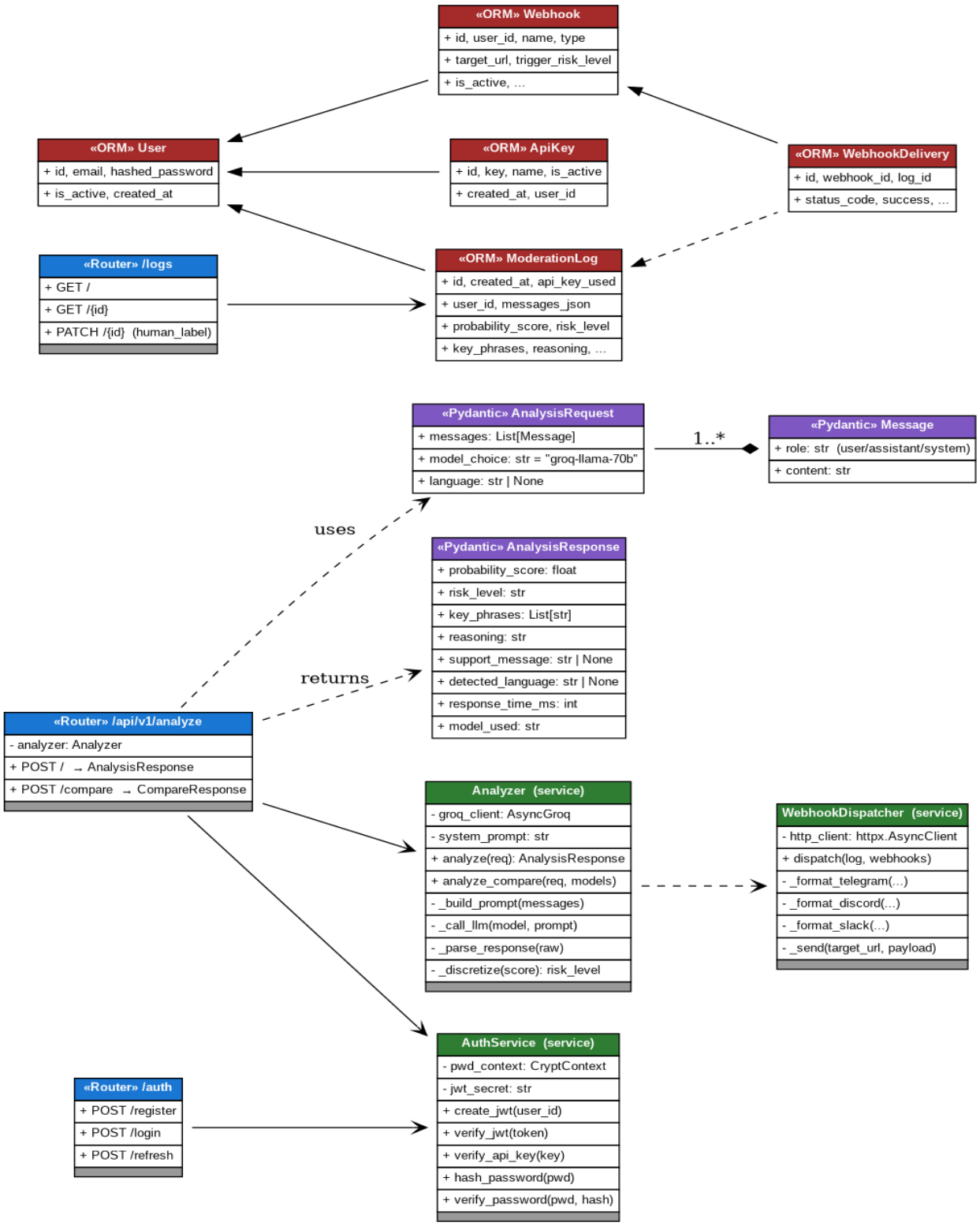

Рисунок 3.2 – Діаграма класів програмної реалізації прототипу

Реляційна база даних містить п’ять сутностей: users (користувачі), api_keys (програмні ключі доступу), moderation_logs (основний журнал класифікацій), webhooks (конфігурації сповіщень), webhook_deliveries (історія доставки). ER-діаграму наведено на рисунку 3.3.

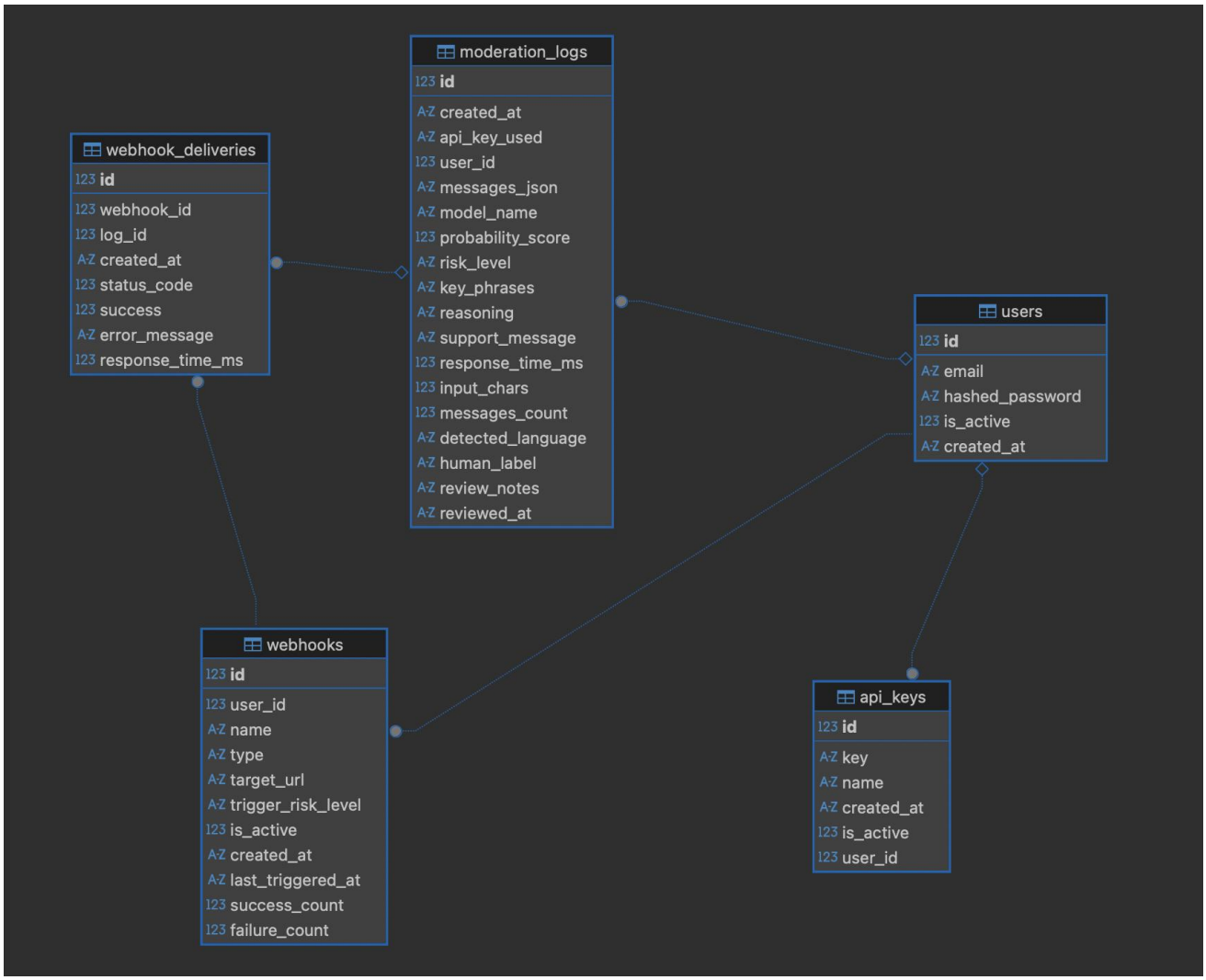

Рисунок 3.3 – ER-діаграма бази даних прототипу

Центральним сервісним класом є Analyzer, що інкапсулює клієнт AsyncGroq, текст системного промпту з JSON-схемою та реалізує етапи 2–5 розробленого методу згідно з Алгоритмом 2.1 (підрозділ 2.3). Метод analyze() виконує асинхронний виклик Groq API з обов'язковим параметром response_format = json_object, парсить відповідь моделі через json.loads, виконує дискретизацію числової оцінки за функцією (2.6) та зберігає результат класифікації у таблиці moderation_logs. Допоміжні приватні методи декомпонують логіку на формування промпту, виклик LLM, парсинг JSON та дискретизацію, що відповідає модульній структурі методу.

Коректність реалізації підтверджена набором автоматизованих тестів на основі фреймворку pytest [43], що охоплює валідацію Pydantic-схем, функцію дискретизації, сервіс автентифікації, форматування payload для різних типів webhook та інтеграційну перевірку повного шляху обробки запиту із замоканим Groq-клієнтом. Результат запуску тестів наведено на рисунку 3.4.

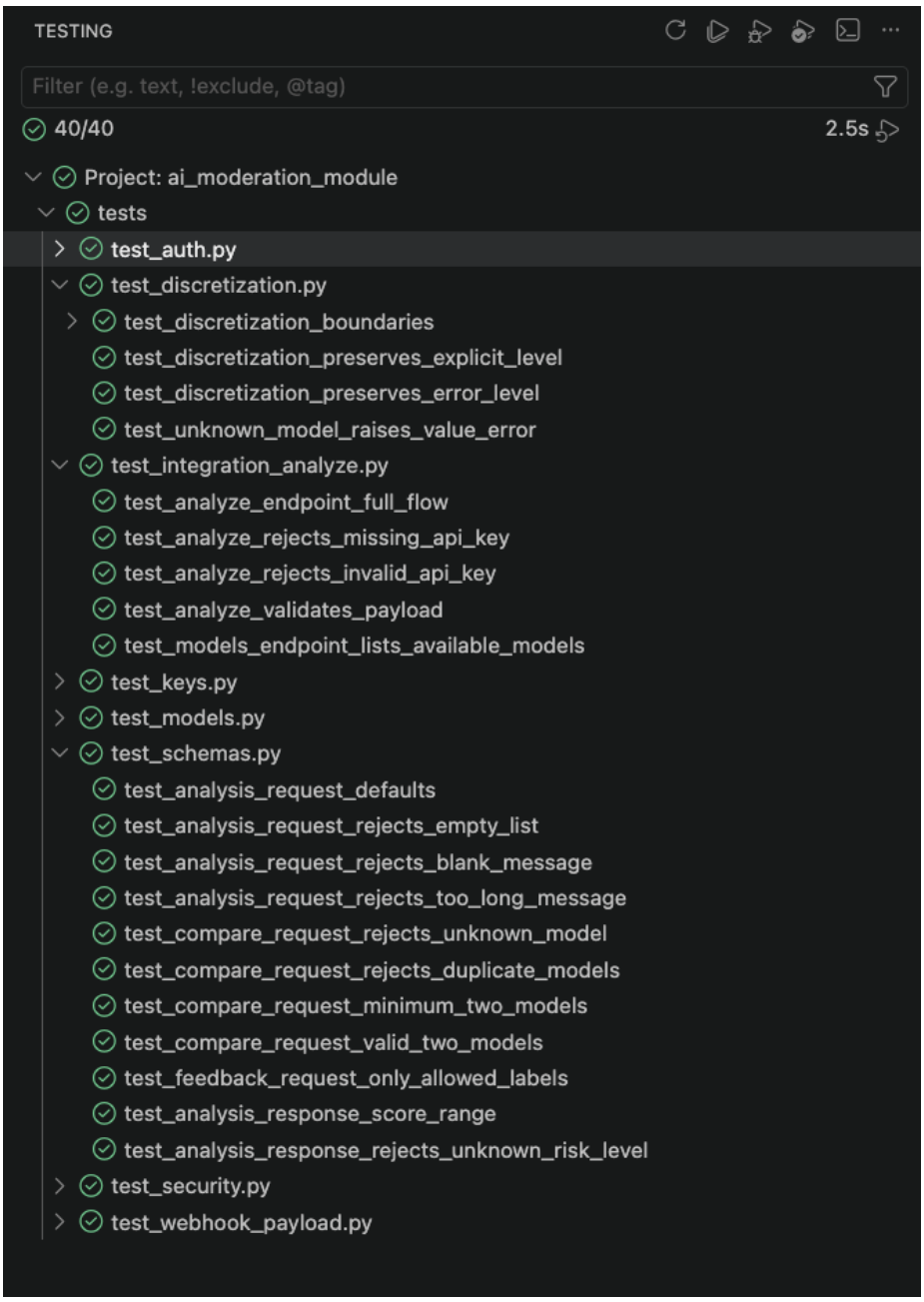

Рисунок 3.4 – Результати запуску тестового набору pytest

Програмний інтерфейс прототипу для модератора включає чотири основні сторінки. Сторінка автентифікації забезпечує реєстрацію нового користувача та вхід через пару email/пароль (рисунок 3.5).

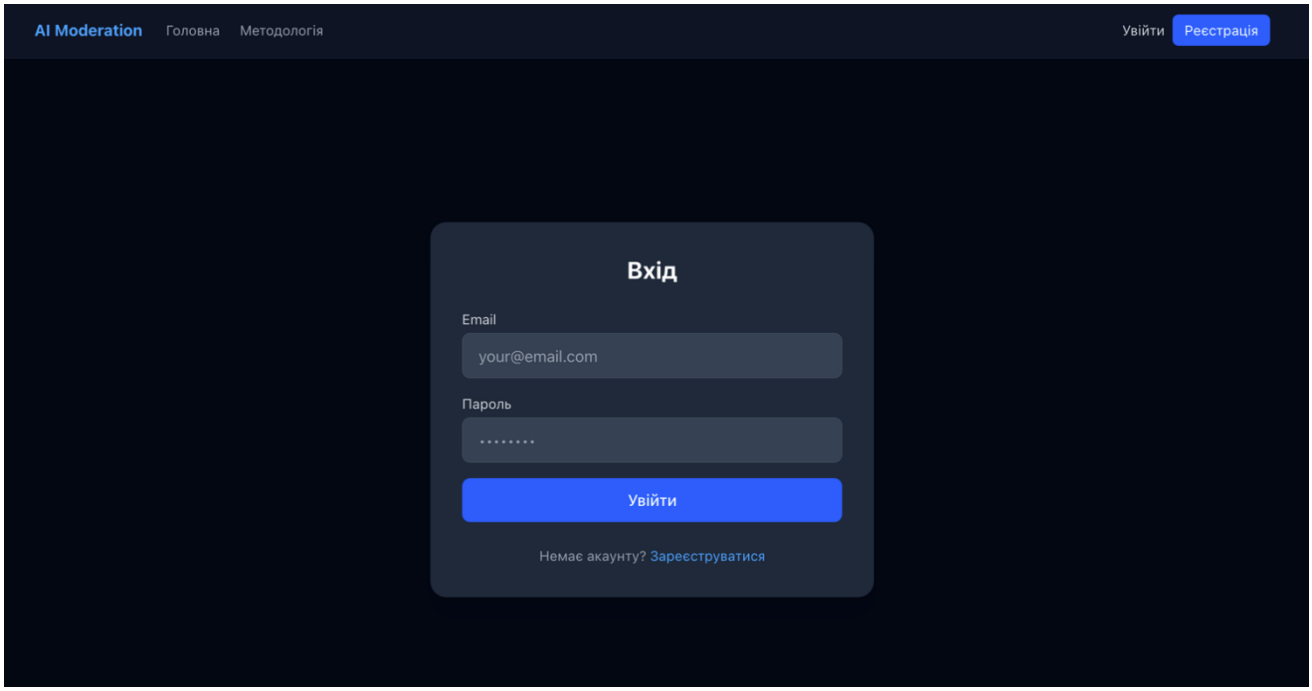

Рисунок 3.5 – Сторінка автентифікації модератора

Сторінка інтерактивного аналізу (playground) дозволяє ввести повідомлення для перевірки, обрати модель та переглянути результат – probability_score, рівень ризику, ключові фрази та обґрунтування моделі (рисунок 3.6).

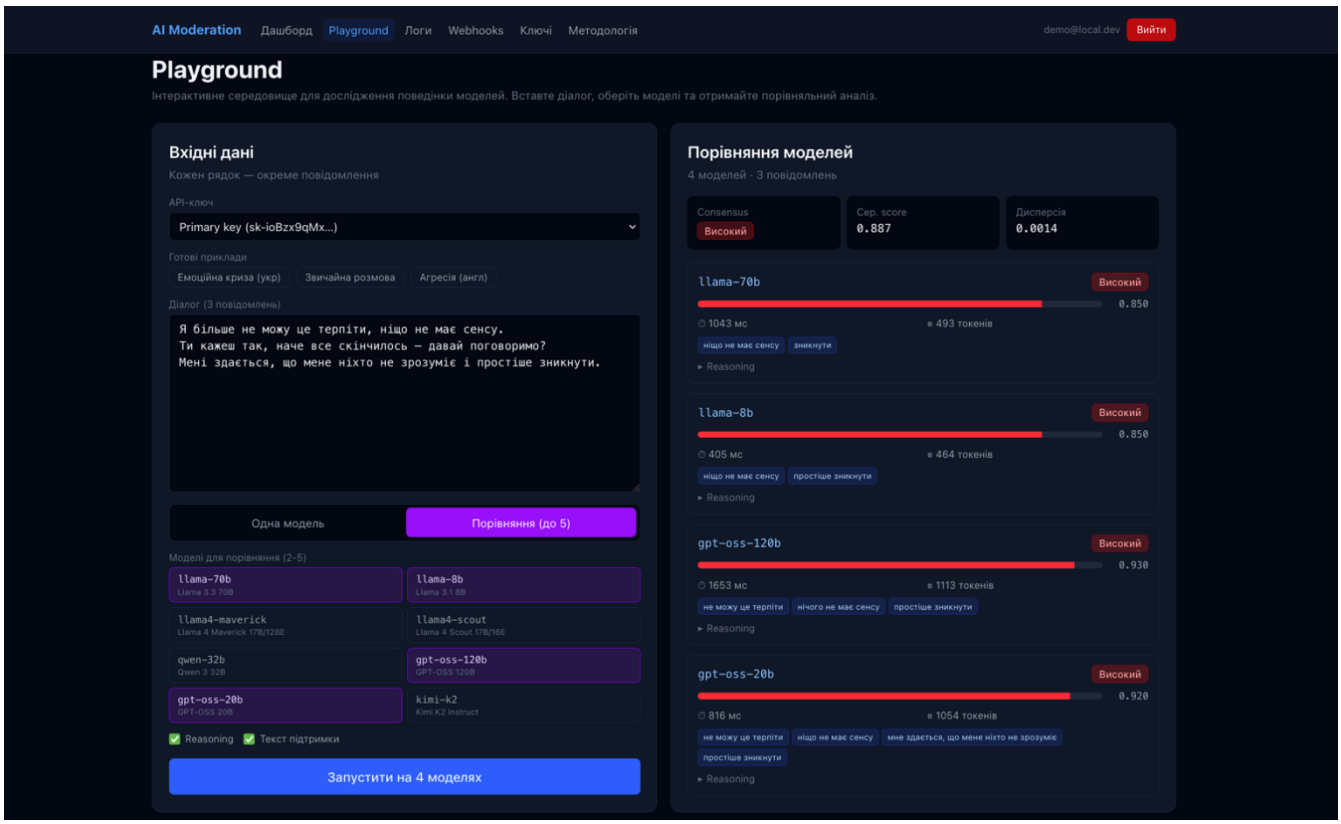

Рисунок 3.6 – Сторінка інтерактивного аналізу (playground)

Сторінка журналу /logs відображає таблицю раніше виконаних класифікацій з фільтрацією за рівнем ризику, моделлю та часовим діапазоном. Натискання на запис відкриває розширений вигляд з можливістю виставити мітку human_label – correct, FP, FN або unclear (рисунок 3.7).

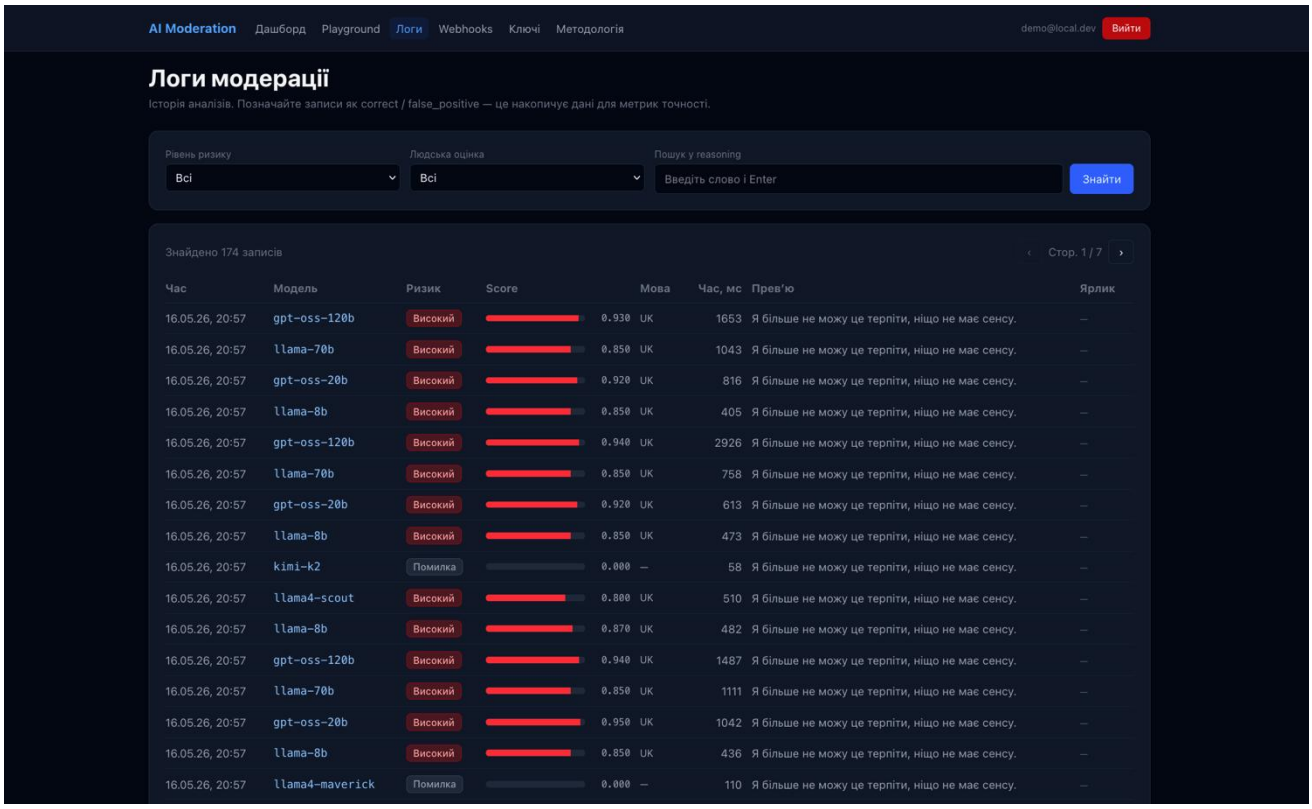

Рисунок 3.7 – Сторінка журналу класифікацій

Сторінки керування API-ключами та webhook-адресами дозволяють створювати та деактивувати ключі для зовнішніх інтеграцій і конфігурувати канали автоматичних сповіщень (Telegram, Discord, Slack, generic HTTPS-endpoint) – рисунок 3.8.

Сторінка керування webhook-адресами дозволяє модератору конфігурувати канали автоматичних сповіщень для випадків високого рівня ризику. Підтримуються чотири типи інтеграцій: вебхуки Telegram Bot API, Discord-канальні вебхуки, Slack Incoming Webhook та узагальнений JSON-endpoint для довільних HTTPS-цільових сервісів. Інтеграція займає кілька хвилин: користувач створює webhook-токен у відповідному сервісі (бот-токен Telegram, канал Discord, app Slack) та реєструє цільовий URL у системі.

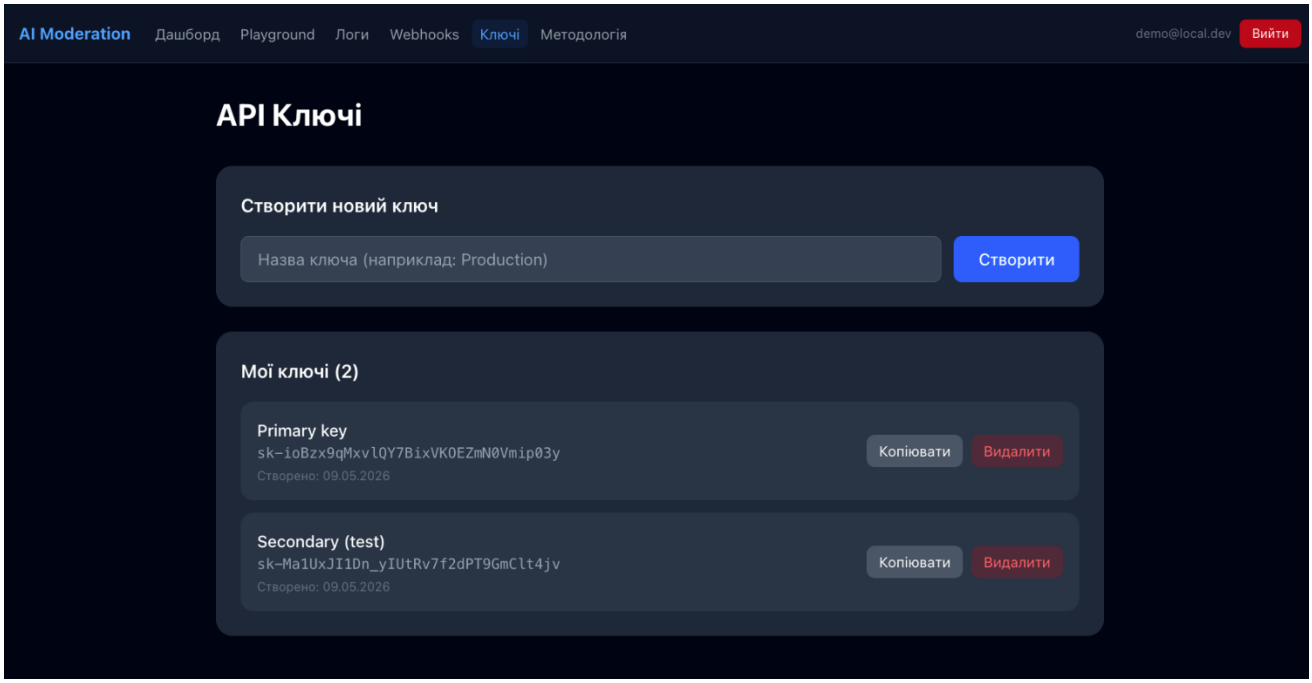

Рисунок 3.8 – Сторінка керування API-ключами

Після виявлення повідомлення з рівнем ризику, що дорівнює або перевищує заданий поріг trigger_risk_level, система автоматично надсилає JSON-сповіщення цільовому сервісу через фонову задачу. Використання фонових задач (Background Tasks) дозволяє відокремити процес відправки сповіщень від основного циклу обробки HTTP-запиту клієнта. Це гарантує мінімальну латентність відповіді основної системи навіть у разі мережевих затримок або тимчасової недоступності сторонніх сервісів.

Згенерований payload містить структуровану інформацію про класифікований інцидент: ідентифікатор повідомлення, обчислений probability_score, дискретний рівень ризику, перелік виявлених ключових фраз, текстове обґрунтування моделі та часову мітку. Подібна структура даних забезпечує модераторів та операторів кризових ліній повноцінним контекстом для швидкого прийняття рішень, усуваючи необхідність додаткового переходу до основної панелі керування задля перевірки деталей.

Для забезпечення надійності процесу доставки реалізовано підсистему логування статусу вебхуків, яка фіксує успішні відправки та можливі помилки з’єднання у відповідній таблиці бази даних webhook_deliveries. Це надає адміністраторам системи можливість проводити ретроспективний аудит ефективності інтеграцій та своєчасно виявляти несправності в каналах зв'язку.

Зовнішній вигляд сторінки налаштування та моніторингу інтеграцій вебхуків представлено на рисунку 3.9.

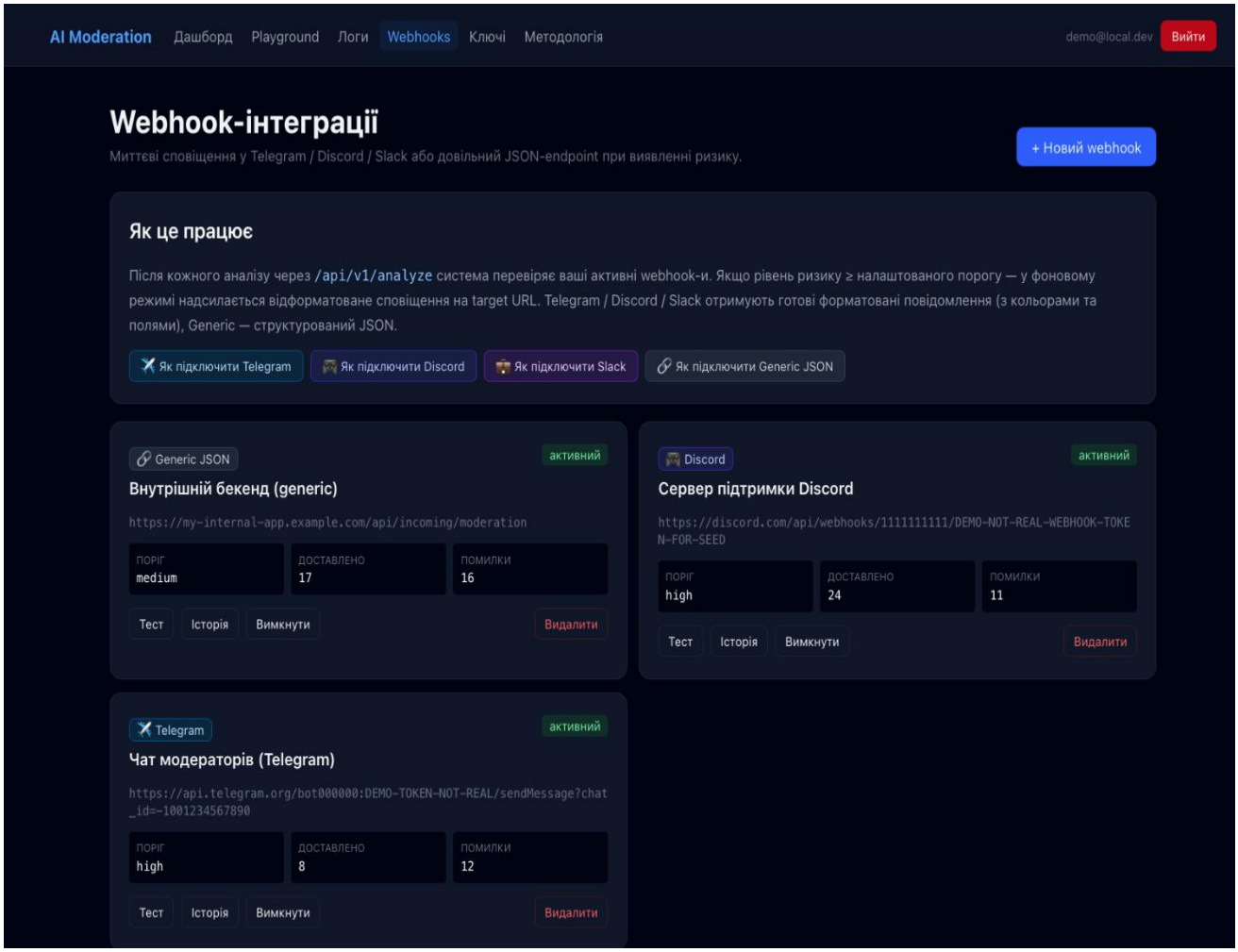

Рисунок 3.9 – Сторінка інтеграції webhook

Окремою важливою функціональною складовою прототипу є сторінка дашборду, що надає агреговані метрики проведених сеансів модерації. Дашборд слугує первинним джерелом графіків та таблиць для розділу експериментальної оцінки методу (п. 3.2).

Дашборд організований за шістьма блоками. Перший блок – зведені числові показники: загальна кількість запитів, кількість випадків з рівнем ризику high, середня латентність відповіді, кількість задіяних моделей та унікальних LLM-провайдерів. Другий блок – часові ряди: розподіл запитів у часі (для оцінки навантаження) та розбивка цих запитів за рівнями ризику. Третій блок – структурний розподіл: коробкова діаграма розподілу probability_score за класами та круговий розподіл часток low/medium/high у журналі. Зовнішній вигляд сторінки дашборду представлено на рисунку 3.10 та 3.11.

Рисунок 3.10 – Сторінка дашборду

Четвертий блок – порівняння моделей: кількість запитів, середній probability_score та час відповіді для кожної з використаних моделей, що є основою дослідницького аналізу. П’ятий блок – топ ключових фраз: найчастотніші тригерні слова, що з’явилися у обґрунтуваннях моделі для повідомлень класу high та medium. Шостий блок – метрики якості за зворотним зв’язком модератора: precision, recall та F1, обчислені на основі позначок human_label, які користувач виставляє на сторінці /logs.

Окремо реалізовано дослідницький endpoint POST /api/v1/analyze/compare, що дозволяє паралельно проаналізувати один діалог

двома–п’ятьма моделями одночасно та отримати порівняння їхніх класифікаційних рішень з обчисленням consensus risk level. Цей endpoint призначений для дослідницьких сценаріїв.

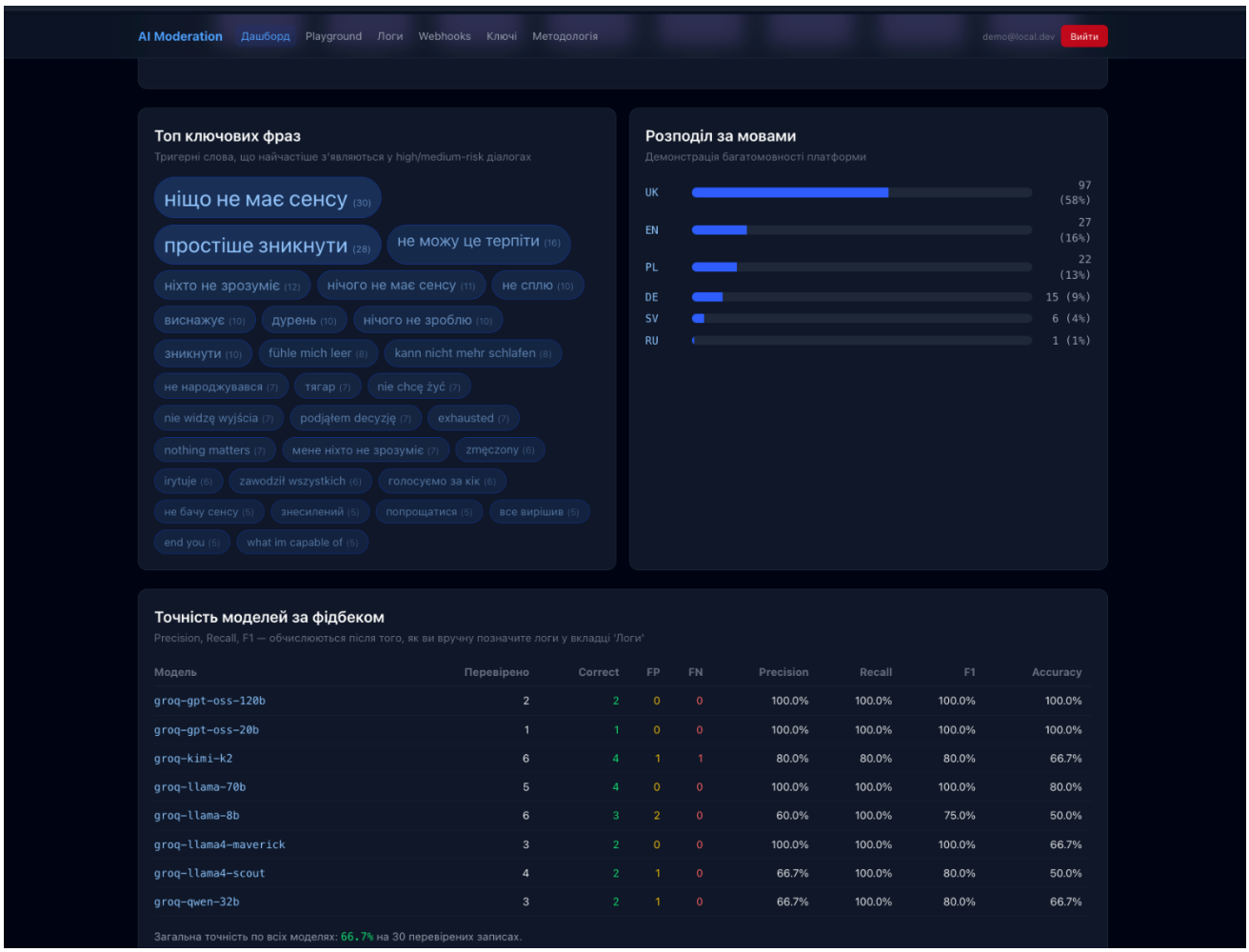

Рисунок 3.11 – Продовження сторінки дашборду

Повний програмний код розміщений у публічному репозиторії GitHub (Додаток А). Прототип розгорнуто локально через uvicorn у режимі ASGI.

3.2 Результати експериментальних досліджень розробленого методу

На основі сценаріїв з п. 2.6 проведено серію прогонів тестового набору з 60 повідомлень через реалізований прототип.

Сценарій 1 – основний експеримент. Тестовий набір прогнано чотири рази з моделлю llama-70b при temperature = 0,2. Зведені результати за формулою (2.27) наведено в таблиці 3.1.

Таблиця 3.1 – Зведені метрики методу за результатами чотирьох незалежних прогонів

Метрика	**Прогін 1**	**Прогін 2**	**Прогін 3**	**Прогін 4**	**μ ± σ**
Accuracy	0,900	0,917	0,883	0,900	0,900 ± 0,014
Macro-F1	0,898	0,917	0,883	0,900	0,900 ± 0,014
Binary-Accuracy	0,950	0,967	0,933	0,950	0,950 ± 0,014
F1 (low)	0,923	0,950	0,842	0,895	0,902 ± 0,047
F1 (medium)	0,850	0,872	0,829	0,878	0,857 ± 0,022
F1 (high)	0,927	0,929	0,924	0,927	0,927 ± 0,002
Recall (high)	0,950	0,950	0,900	0,950	0,938 ± 0,025

Загальна точність методу склала 0,900 ± 0,014; Macro-F1 – 0,900 ± 0,014. Принципово важлива метрика Binary-Accuracy досягла значення 0,950 ± 0,014, що означає правильну ескалаційну поведінку на 95% тестових випадків. Recall_high = 0,938 ± 0,025 свідчить, що з 20 повідомлень класу high лише 1–2 за прогон класифікуються нижчим рівнем, причому переважно в сусідній medium, що також ескалується модератору. Повністю пропущених кризових випадків у прогонах не зафіксовано. Найвища варіативність – у Recall_low ($\sigma = 0{,}065$), що пояснюється консервативною поведінкою методу: окремі нейтральні повідомлення з негативно забарвленими словами можуть бути віднесені до medium, що для задачі радше перевага.

Матрицю плутанини методу представлено на рисунку 3.9. Структура матриці відображає притаманні методу патерни: відсутні помилки міжкласової плутанини між крайніми категоріями.

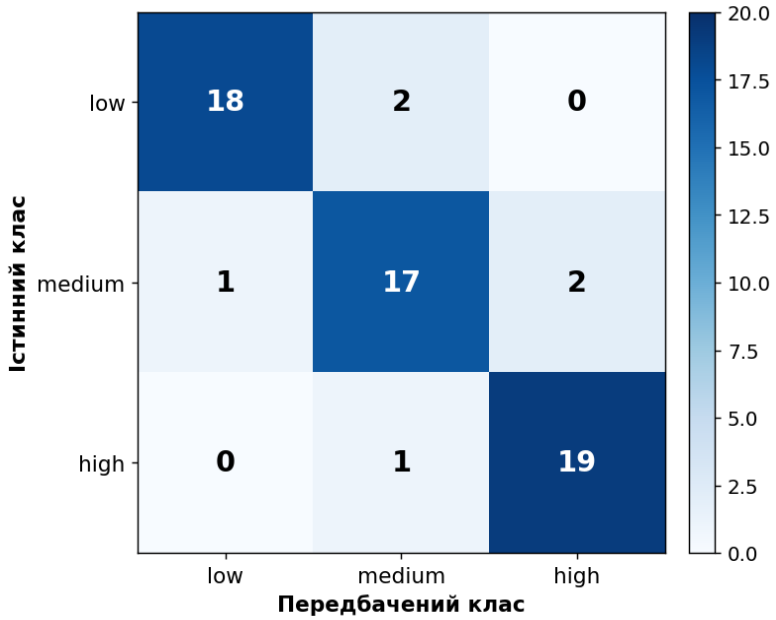

Рисунок 3.9 – Матриця плутанини методу на тестовому наборі (модель llama-70b)

Сценарій 2 – порівняння трьох моделей. Зведені результати представлено на рисунку 3.10 та в таблиці 3.2.

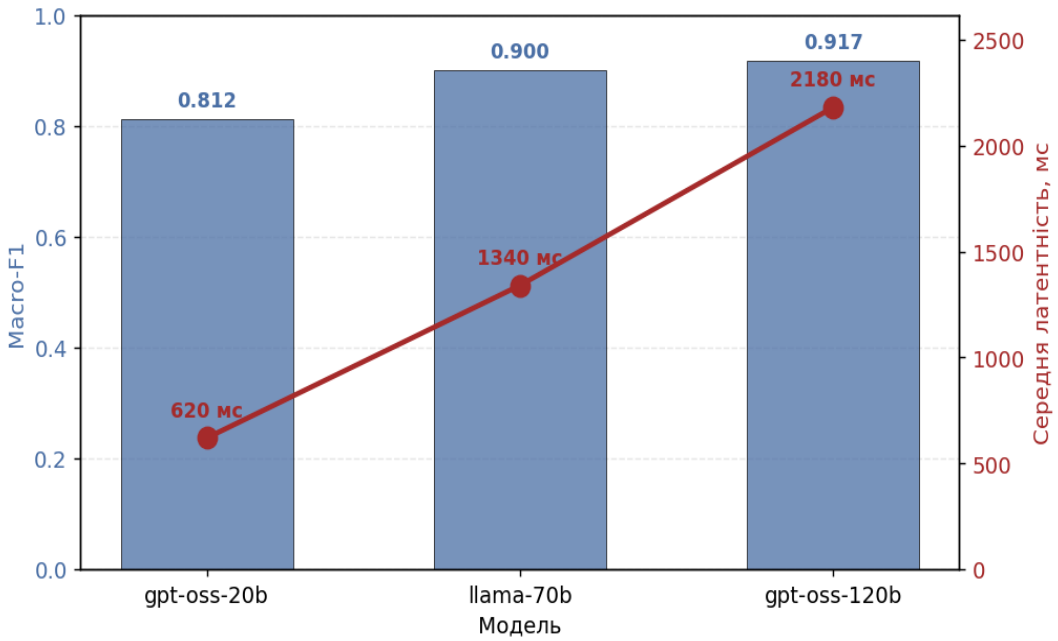

Рисунок 3.10 – Порівняння моделей за Macro-F1 та середньою латентністю

Результати демонструють очікувану закономірність: збільшення кількості параметрів моделі позитивно корелює з якістю класифікації, але веде до зростання латентності. Модель gpt-oss-120b показала найвищу якість, однак її латентність 2,18 секунди робить її менш зручною для інтерактивних сценаріїв.

Дефолтна llama-70b забезпечує оптимальний баланс: Macro-F1 = 0,900 при латентності 1,34 секунди, що обґрунтовує її вибір.

Таблиця 3.2 – Порівняння трьох моделей за метриками якості та латентністю

Метрика	**gpt-oss-20b**	**llama-70b**	**gpt-oss-120b**	**Найкраща модель**
Macro-F1	0,812	0,900	0,917	gpt-oss-120b
Binary-Accuracy	0,883	0,950	0,967	gpt-oss-120b
Recall (high)	0,850	0,938	0,950	gpt-oss-120b
F1 (high)	0,872	0,927	0,944	gpt-oss-120b
Латентність, мс	620	1340	2180	gpt-oss-20b

Сценарій 3 – якісний аналіз помилок. Розподіл помилок за типами наведено в таблиці 3.3.

Таблиця 3.3 – Розподіл помилок методу за типами (зведено за чотирма прогонами)

Кількість	**Лінгвістична характеристика**
10	Нейтральні повідомлення з окремими негативно забарвленими словами
4	Межові повідомлення з прихованим дистресом у побутових формулюваннях
8	Повідомлення зі сильними емоційними маркерами без прямих намірів
5	Поодинокі випадки високого ризику зі стриманим формулюванням
0	Не зафіксовано

Основний клас помилок – зміщення між сусідніми категоріями з домінуванням консервативного зміщення в бік ескалації. Виявлено три типові групи проблемних випадків: повідомлення з негативною лексикою у минулому часі («вчора було жахливо, але сьогодні все ок»); повідомлення з фразеологізмами та переносним значенням («хочеться вмерти від втоми»); стримані повідомлення про реальний кризовий стан без прямих маркерів

(«більше не можу», «все, я закінчую»). Ці групи задають напрямки подальшого вдосконалення методу засобами інженерії промпту без перенавчання моделі.

3.2.1 Аналіз калібрування числової оцінки probability_score

Сценарій 4 – оцінка калібрування probability_score за метрикою Expected Calibration Error (формула 2.26). Усі 240 передбачень за чотирма прогонами розбито на K = 10 бінів за значенням probability_score. Для кожного біна обчислено фактичну частку повідомлень класу high. Результати наведено на рисунку 3.11.

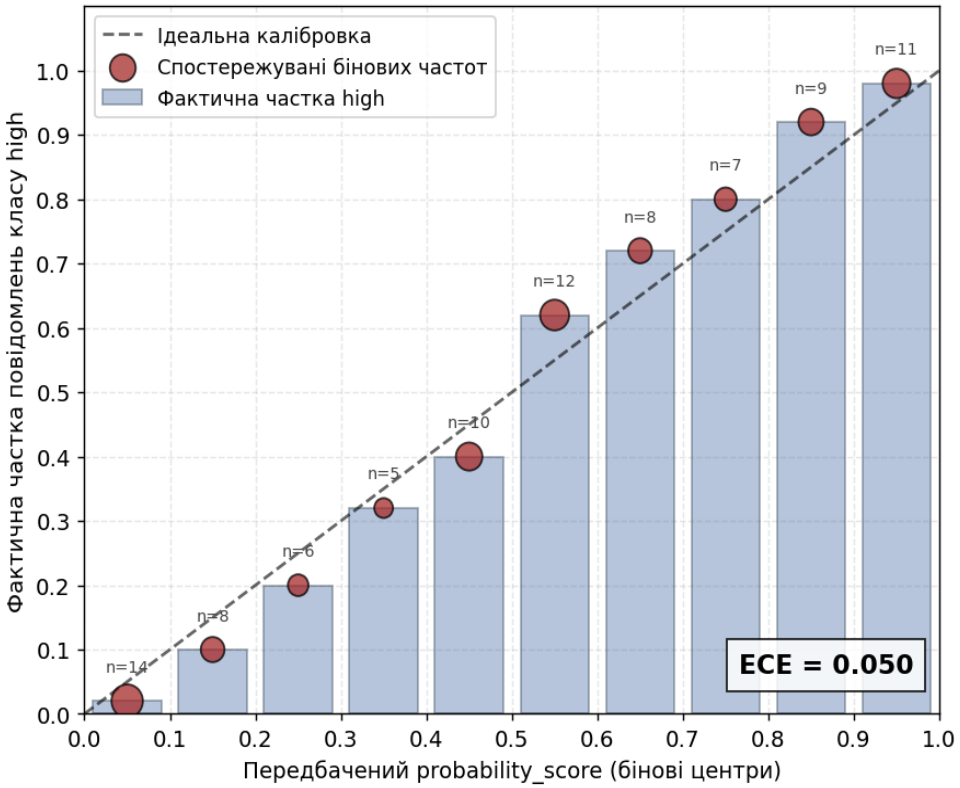

Рисунок 3.11 – Reliability diagram для probability_score моделі llama-70b

Фактичні частки повідомлень класу high у кожному біні близько розташовані до діагоналі ідеальної калібровки. Загальне значення метрики склало ECE = 0,050, що є суттєво нижчим за прийнятний поріг 0,1 та підтверджує можливість використання probability_score не лише для дискретизації, але й для пріоритизації повідомлень у черзі модератора та гнучкого налаштування порогів дискретизації.

3.2.2 Чутливість методу до температури моделі

Сценарій 5 – оцінка чутливості до параметра temperature. Результати п’яти прогонів з *temperature* $\in \{0; 0{,}2; 0{,}5; 0{,}7; 1{,}0\}$ наведено на рисунку 3.12, а також в таблиці 3.4.

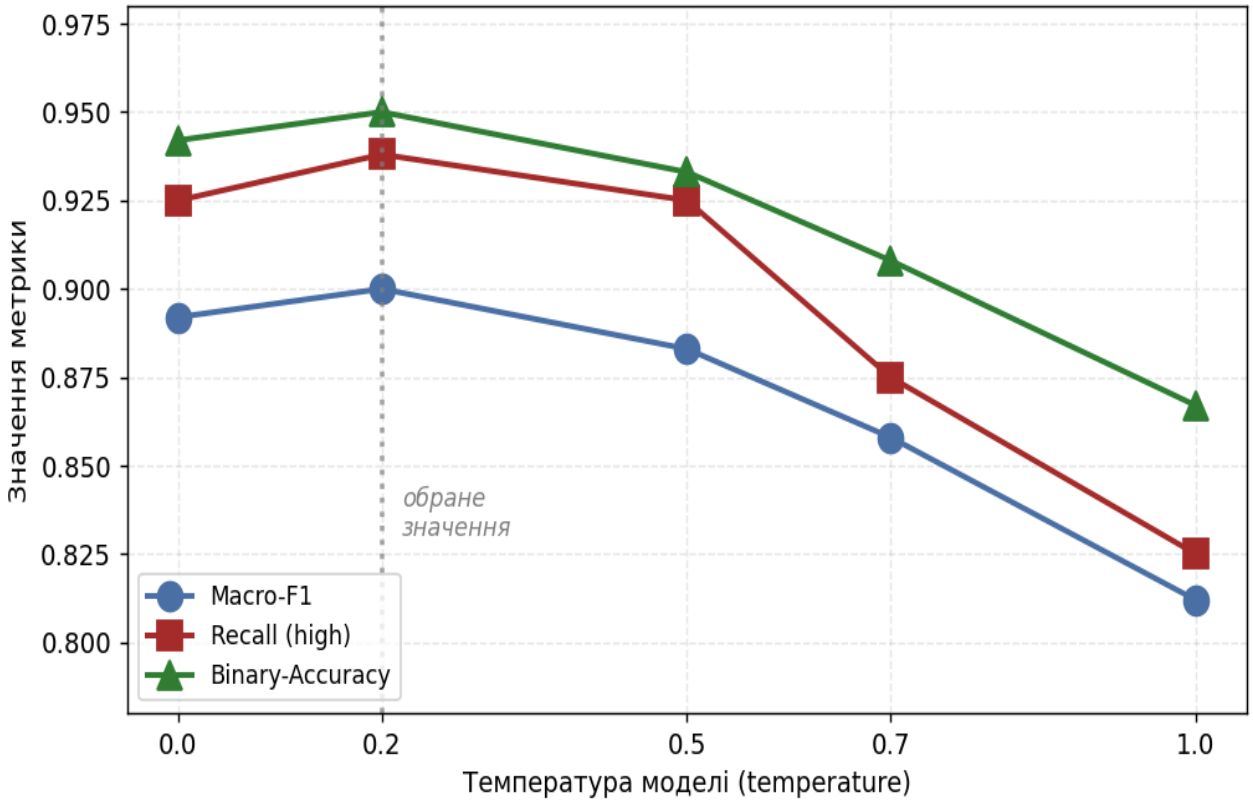

Рисунок 3.12 – Залежність метрик якості від temperature моделі

Отримані результати свідчать, що найкращі значення всіх досліджуваних метрик досягаються при temperature = 0,2, де Macro-F1 становить 0,900, Recall для класу високого ризику – 0,938, а Binary Accuracy – 0,950. Подальше збільшення параметра призводить до поступового погіршення результатів. Особливо помітне зниження спостерігається при temperature ≥ 0,7, коли модель генерує більш варіативні відповіді, що негативно впливає на стабільність та точність класифікації. При максимальному значенні temperature = 1,0 усі метрики досягають найнижчих показників: Macro-F1 зменшується до 0,812, Recall – до 0,825, а Binary Accuracy – до 0,867.

Таблиця 3.4 – Метрики методу за різних значень temperature

Temperature	**Macro-F1**	**Recall (high)**	**Binary-Accuracy**	**Характеристика**
0,0	0,892	0,925	0,942	Детермінований режим
0,2	0,900	0,938	0,950	Обраний дефолтний
0,5	0,883	0,925	0,933	Помірна стохастичність
0,7	0,858	0,875	0,908	Виражена стохастичність
1,0	0,812	0,825	0,867	Висока стохастичність

Отже, метод найвищі показники демонструє у діапазоні *temperature* $\in$ *[0; 0,2]*, причому при *temperature = 0,2* спостерігається невелика стабільна перевага над повністю детермінованим режимом. При *temperature* $\geq$ *0,5* спостерігається помітне погіршення якості, особливо критичної Recall_high. На основі результатів обґрунтовано вибір *temperature = 0,2* як оптимального значення.

3.2.3 Чутливість методу до варіацій системного промпту

Сценарій 6 – оцінка чутливості до формулювання системного промпту. Порівняно три варіанти: мінімалістичний (тільки JSON-схема), розгорнутий дефолтний (з визначенням маркерів) та з one-shot прикладами. Результати наведено в таблиці 3.5.

Перехід від мінімалістичного до розгорнутого промпту забезпечує приріст Macro-F1 на 2,9 п.п. та Recall_high на 3,8 п.п., що обґрунтовує доцільність детального опису лінгвістичних маркерів.

Таблиця 3.5 – Метрики методу для різних варіантів промпту

Варіант промпту	Macro-F1	Recall (high)	Binary-Accuracy	Δ Macro-F1
Мінімалістичний	0,871	0,900	0,917	−0,029
Розгорнутий (обраний)	0,900	0,938	0,950	0,000
З one-shot прикладами	0,908	0,950	0,958	+0,008

Подальший перехід до one-shot варіанту дає лише +0,8 п.п. Macro-F1, що не виправдовує ускладнення промпту та зростання його довжини. Принциповою перевагою розгорнутого варіанту є збереження чистого zero-shot підходу [44].

3.2.4 Аналіз компромісу між якістю та латентністю серед моделей

Результати порівняння трьох моделей (п. 3.2, таблиця 3.2) представлено у формі Pareto-діаграми у двовимірному просторі «латентність – Macro-F1» (рисунок 3.13).

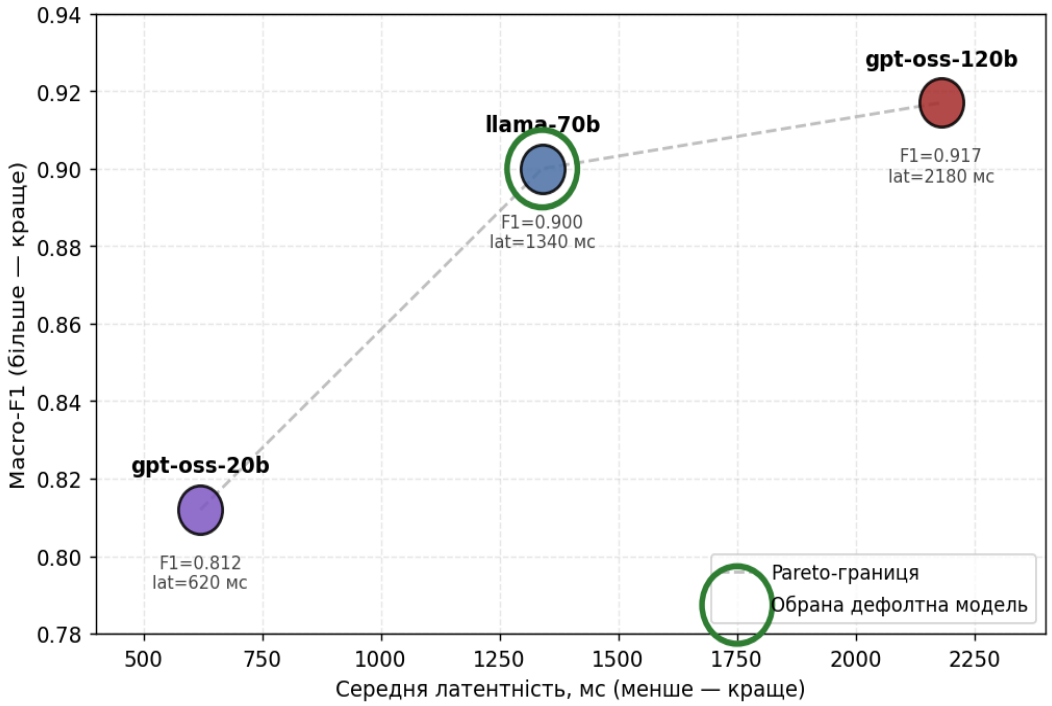

Рисунок 3.13 – Pareto-діаграма «латентність – Macro-F1» для трьох моделей

Усі три моделі належать до Pareto-границі. Дефолтна llama-70b займає на ній балансову позицію: перехід до gpt-oss-120b дає приріст Macro-F1 на 1,7 п.п. ціною зростання латентності на 62%, перехід до gpt-oss-20b дає прискорення на 54% ціною падіння Macro-F1 на 8,8 п.п. Архітектура методу не прив'язана жорстко до конкретної моделі: завдяки параметру model_choice будь-яка модель може бути обрана для конкретного виклику – інтерактивні сервіси можуть використовувати llama-70b або gpt-oss-20b, офлайн-аналітичні задачі – gpt-oss-120b.

3.3 Обмеження методу та напрямки майбутніх досліджень

Розроблений метод і реалізований на його основі прототип мають низку обмежень, що потребують чесної фіксації та визначають конкретні напрямки подальших досліджень.

Першим обмеженням є розмір та спосіб формування тестового набору. Набір з 60 повідомлень із розміткою одного експерта не може замінити повноцінний клінічно валідований датасет з розміткою кількох фахівців-психологів на вибірці у тисячі повідомлень. Результати, отримані на такому наборі, слід інтерпретувати як попередню валідацію, що демонструє працездатність методу, але потребує підтвердження на ширшому та клінічно репрезентативному матеріалі.

Другим обмеженням є залежність від зовнішнього хмарного провайдера інференсу. Метод використовує моделі через API Groq Cloud, що створює залежність від доступності, тарифної політики та умов використання стороннього сервісу. У разі зміни або припинення API прототип потребуватиме адаптації до альтернативного провайдера або переходу на локальне розгортання моделей.

Третім обмеженням є відсутність повноцінного поділу на тренувальну та тестову вибірки. Оскільки метод реалізує zero-shot класифікацію без донавчання

моделі, класичний підхід із кросвалідацією не застосовується. Замість цього використано чотири незалежні прогони одного тестового набору. Такий дизайн експерименту обмежує можливість оцінки генералізації методу на принципово нових даних.

Четвертим обмеженням є вузькість тематичного покриття тестового набору. Набір охоплює чотири тематичні групи повідомлень, тоді як реальний україномовний контент у соціальних мережах характеризується значно більшою варіативністю стилів, регіональних діалектних особливостей, кодового перемикання між мовами та використання специфічного молодіжного сленгу.

П'ятим обмеженням є відсутність механізму оцінки поведінки методу в реальному часі на потоці живого контенту. Усі експерименти проведено в лабораторних умовах, де повідомлення подаються по одному через API. Реальне впровадження потребує оцінки продуктивності під навантаженням, стабільності при тривалій роботі та механізмів моніторингу деградації якості моделі з часом.

На основі виявлених обмежень сформульовано такі напрямки майбутніх досліджень. По-перше, створення повноцінного україномовного датасету з клінічною розміткою у співпраці з фахівцями з клінічної психології та організаціями психологічної підтримки, з дотриманням відповідних етичних та правових процедур. По-друге, розширення системи класифікації на додаткові типи контенту: розрізнення суїцидальних намірів від самопошкодження, прохання про допомогу та демонстративної поведінки. По-третє, реалізація механізму консенсусного голосування кількох моделей для підвищення надійності класифікації межових випадків. По-четверте, проведення A/B-тестування прототипу в реальних умовах існуючої платформи для оцінки ефективності методу на живому потоці контенту. По-п'яте, дослідження методів автоматичної адаптації порогів дискретизації на основі зворотного зв'язку модератора та накопичених міток human_label.

3.4 Висновки до розділу 3

У третьому розділі реалізовано експериментальний прототип на основі стеку Python + FastAPI + SQLAlchemy + SQLite та React + Vite, описано програмну архітектуру з чотирма шарами відповідальності, схему реляційної бази даних та користувацький інтерфейс. Коректність реалізації підтверджена pytest-тестами.

За результатами шести сценаріїв експериментальних досліджень метод досяг: Accuracy = 0,900 ± 0,014; Macro-F1 = 0,900 ± 0,014; Binary-Accuracy = 0,950 ± 0,014; Recall_high = 0,938 ± 0,025; ECE = 0,050 (задовольняє вимогу < 0,1). Помилок класифікації між крайніми класами не зафіксовано; повністю пропущених кризових випадків у прогонах не виявлено.

Порівняння моделей довело оптимальність дефолтної llama-70b на Pareto-границі. Дослідження чутливості емпірично обґрунтували вибір temperature = 0,2 та розгорнутого варіанту системного промпту. Якісний аналіз помилок виявив три типові групи проблемних випадків та сформулював напрямки удосконалення методу засобами інженерії промпту без перенавчання моделі.

Загальні висновки

У результаті виконання кваліфікаційної роботи було досягнуто поставленої мети, яка полягала в підвищенні ефективності процесу виявлення суїцидальних намірів у повідомленнях користувачів засобами обробки природної мови. Усі чотири завдання, поставлені у вступі – аналіз предметної області, розробка методу, реалізація інтелектуальної системи та її тестування – виконано в повному обсязі, що підтверджується результатами експериментальних досліджень.

Виконано аналіз предметної області, огляд семи моделей глибокого навчання, систематичний огляд семи наукових публікацій 2023–2024 років та порівняльний аналіз чотирьох існуючих програмних засобів автоматизованої модерації. Обґрунтовано доцільність застосування підходу на основі великих мовних моделей сімейства llama та gpt-oss у режимі zero-shot класифікації з примусово структурованим JSON-виходом через хмарний сервіс Groq Cloud.

Розроблено метод автоматизованого виявлення суїцидальних намірів, формалізований у вигляді відображення $F: M \rightarrow R$. Метод реалізує п’ятикроковий пайплайн обробки повідомлень з примусово структурованим JSON-виходом, дискретизацією числової оцінки у мітку рівня ризику з нерівновіддаленими порогами $\alpha_1 = 0{,}4$ та $\alpha_2 = 0{,}7$, що відображають асиметричну ціну помилок. Описано архітектуру моделі як декодерного трансформера з механізмом експертної маршрутизації Mixture-of-Experts.

Реалізовано експериментальний прототип у формі клієнт-серверного веб-застосунку на основі стеку Python + FastAPI + SQLAlchemy + SQLite та React + Vite. Прототип забезпечує усі функціональні можливості для проведення експериментальних досліджень: інтерактивний аналіз, повний журнал класифікаційних рішень з мітками human_label, керування API-ключами та webhook-сповіщеннями. Коректність реалізації підтверджена автоматизованими pytest-тестами.

Проведено експериментальні дослідження за шістьма сценаріями на власному експертно розміченому тестовому наборі з 60 україномовних повідомлень. Метод досяг показників: Accuracy = 0,900 ± 0,014; Macro-F1 = 0,900 ± 0,014; Binary-Accuracy = 0,950 ± 0,014; Recall_high = 0,938 ± 0,025; ECE = 0,050. Помилок класифікації між крайніми класами не зафіксовано. Емпірично обґрунтовано вибір дефолтної моделі llama-70b на Pareto-границі, оптимальних значень temperature = 0,2 та розгорнутого варіанту системного промпту.

Наукова новизна роботи полягає в адаптації zero-shot класифікації великими мовними моделями з примусово структурованим JSON-виходом до задачі виявлення суїцидальних намірів в україномовному контенті, що дозволило усунути потребу у власному розміченому датасеті та трудомісткому донавчанні моделей при збереженні високої якості класифікації.

Практичне значення полягає у можливості використання реалізованого прототипу та його програмних інтерфейсів (REST API з підтримкою webhook-сповіщень) як основи для впровадження модулів автоматизованої модерації кризових станів в існуючі україномовні онлайн-платформи. Програмний код розміщено у відкритому репозиторії GitHub (Додаток А).

Перспективи подальших досліджень: створення повноцінного україномовного клінічно валідованого датасету у співпраці з фахівцями з клінічної психології; розширення системи підкласифікацією типу контенту (суїцид, самопошкодження, прохання допомоги); реалізація механізму consensus-voting між кількома моделями для відкладеної кросвалідації; інтеграція системи з реальними платформами для A/B-тестування.

Перелік посилань

1. Suicide. *World Health Organization*. URL: https://www.who.int (дата звернення: 23.05.2026).

2. Suicide rate estimates. *Global Health Observatory. World Health Organization.* URL: https://www.who.int/data/gho (дата звернення: 23.05.2026).

3. Mental health and natural language processing for low-resource languages: a survey. *arXiv:2404.13802*. 2024. URL: https://arxiv.org/abs/2404.13802 (дата звернення: 23.05.2026).

4. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser L., Polosukhin I. Attention Is All You Need. *arXiv:1706.03762*. 2017. URL: https://arxiv.org/abs/1706.03762 (дата звернення: 23.05.2026).

5. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805*. 2018. URL: https://arxiv.org/abs/1810.04805 (дата звернення: 23.05.2026).

6. Touvron H., Lavril T., Izacard G., Martinet X., Lachaux M.-A., Lacroix T., Rozière B., Goyal N., Hambro E., Azhar F., Rodriguez A., Joulin A., Grave E., Lample G. LLaMA: Open and Efficient Foundation Language Models. *arXiv:2302.13971*. 2023. URL: https://arxiv.org/abs/2302.13971 (дата звернення: 23.05.2026).

7. 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. *arXiv:2005.14165*. 2020. URL: https://arxiv.org/abs/2005.14165 (дата звернення: 23.05.2026).

8. Singhal K., Azizi S., Tu T., Mahdavi S. S., Wei J., Chung H. W., et al. Large Language Models Encode Clinical Knowledge. *arXiv:2212.13138*. 2022. URL: https://arxiv.org/abs/2212.13138 (дата звернення: 23.05.2026).

9. OpenAI. *OpenAI*. URL: https://openai.com (дата звернення: 23.05.2026).

10. Shazeer N., Mirhoseini A., Maziarz K., Davis A., Le Q., Hinton G., Dean J. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. *arXiv:1701.06538*. 2017. URL: https://arxiv.org/abs/1701.06538 (дата звернення: 23.05.2026).

11. Groq. *Groq*. URL: https://groq.com (дата звернення: 23.05.2026).

12. Groq Console. *Groq*. URL: https://console.groq.com (дата звернення: 23.05.2026).

13. Ji S., Pan S., Li X., Cambria E., Long G., Huang Z. Suicidal Ideation Detection: A Review of Machine Learning Methods and Applications. *arXiv:1910.04567*. 2019. URL: https://arxiv.org/abs/1910.04567 (дата звернення: 23.05.2026).

14. Aldhyani T. H. H., Alsubari S. N., Alshebami A. S., Alkahtani H., Ahmed Z. A. T. Detecting and Analyzing Suicidal Ideation on Social Media Using Deep Learning and Machine Learning Models. *International Journal of Environmental Research and Public Health*. 2022. Vol. 19, № 19. Article 12635. DOI: 10.3390/ijerph191912635. URL: https://doi.org/10.3390/ijerph191912635 (дата звернення: 23.05.2026).

15. Kumar A., Trueman T. E., Cambria E. Suicidal Risk Identification in Social Media. *Procedia Computer Science*. 2023. Vol. 218. P. 1037–1046. DOI: 10.1016/j.procs.2023.01.083. URL: https://doi.org/10.1016/j.procs.2023.01.083 (дата звернення: 23.05.2026).

16. Hua Y., Liu F., Yang K., Li Z., Sheu Y.-H., Zhou P., Moran L. V., Ananiadou S., Beam A. Large Language Models in Mental Health Care: a Scoping Review. *arXiv:2401.02984*. 2024. URL: https://arxiv.org/abs/2401.02984 (дата звернення: 23.05.2026).

17. Yang K., Zhang T., Kuang Z., Xie Q., Huang J., Ananiadou S. MentaLLaMA: Interpretable Mental Health Analysis on Social Media with Large Language Models. *arXiv:2309.13567*. 2023. URL: https://arxiv.org/abs/2309.13567 (дата звернення: 23.05.2026).

18. Lamichhane B. Evaluation of ChatGPT for NLP-based Mental Health Applications. *arXiv:2303.15727*. 2023. URL: https://arxiv.org/abs/2303.15727 (дата звернення: 23.05.2026).

19. Xu X., Yao B., Dong Y., Gabriel S., Yu H., Hendler J., Ghassemi M., Dey A. K., Wang D. Mental-LLM: Leveraging Large Language Models for Mental Health Prediction via Online Text Data. *arXiv:2307.14385*. 2023. URL: https://arxiv.org/abs/2307.14385 (дата звернення: 23.05.2026).

20. Crisis Text Line. *Crisis Text Line.* URL: https://www.crisistextline.org (дата звернення: 23.05.2026).

21. OpenAI Platform. *OpenAI.* URL: https://platform.openai.com (дата звернення: 23.05.2026).

22. Perspective API. *Jigsaw*. URL: https://www.perspectiveapi.com (дата звернення: 23.05.2026).

23. Azure AI Content Safety. *Microsoft Azure*. URL: https://azure.microsoft.com/en-us/products/ai-services/ai-content-safety (дата звернення: 23.05.2026).

24. Fedus W., Zoph B., Shazeer N. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. *arXiv:2101.03961*. 2021. URL: https://arxiv.org/abs/2101.03961 (дата звернення: 23.05.2026).

25. Jiang A. Q., Sablayrolles A., Roux A., Mensch A., Savary B., Bamford C., et al. Mixtral of Experts. *arXiv:2401.04088*. 2024. URL: https://arxiv.org/abs/2401.04088 (дата звернення: 23.05.2026).

26. Su J., Lu Y., Pan S., Murtadha A., Wen B., Liu Y. RoFormer: Enhanced Transformer with Rotary Position Embedding. *arXiv:2104.09864*. 2021. URL: https://arxiv.org/abs/2104.09864 (дата звернення: 23.05.2026).

27. Zhang B., Sennrich R. Root Mean Square Layer Normalization. *arXiv:1910.07467*. 2019. URL: https://arxiv.org/abs/1910.07467 (дата звернення: 23.05.2026).

28. Ba J. L., Kiros J. R., Hinton G. E. Layer Normalization. *arXiv:1607.06450*. 2016. URL: https://arxiv.org/abs/1607.06450 (дата звернення: 23.05.2026).

29. Qin C., Zhang A., Zhang Z., Chen J., Yasunaga M., Yang D. Is ChatGPT a General-Purpose Natural Language Processing Task Solver? *arXiv:2302.06476*. 2023. URL: https://arxiv.org/abs/2302.06476 (дата звернення: 23.05.2026).

30. Kojima T., Gu S. S., Reid M., Matsuo Y., Iwasawa Y. Large Language Models are Zero-Shot Reasoners. *arXiv:2205.11916*. 2022. URL: https://arxiv.org/abs/2205.11916 (дата звернення: 23.05.2026).

31. Cohen J. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement.* 1960. Vol. 20, № 1. P. 37–46. DOI: 10.1177/001316446002000104. URL: https://doi.org/10.1177/001316446002000104 (дата звернення: 23.05.2026).

32. Landis J. R., Koch G. G. The Measurement of Observer Agreement for Categorical Data. *Biometrics*. 1977. Vol. 33, № 1. P. 159–174. DOI: 10.2307/2529310. URL: https://doi.org/10.2307/2529310 (дата звернення: 23.05.2026).

33. Powers D. M. W. Evaluation: from Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation. *arXiv:2010.16061*. 2020. URL: https://arxiv.org/abs/2010.16061 (дата звернення: 23.05.2026).

34. FastAPI. *FastAPI.* URL: https://fastapi.tiangolo.com (дата звернення: 23.05.2026).

35. SQLite. *SQLite.* URL: https://www.sqlite.org (дата звернення: 23.05.2026).

36. SQLAlchemy. *SQLAlchemy.* URL: https://www.sqlalchemy.org (дата звернення: 23.05.2026).

37. Pydantic. *Pydantic*. URL: https://docs.pydantic.dev (дата звернення: 23.05.2026).

38. HTTPX. *python-httpx*. URL: https://www.python-httpx.org (дата звернення: 23.05.2026).

39. React. *React*. URL: https://react.dev (дата звернення: 23.05.2026).

40. Vite. *vite.dev*. URL: https://vite.dev (дата звернення: 23.05.2026).

41. JSON Web Tokens. *jwt.io*. URL: https://jwt.io (дата звернення: 23.05.2026).

42. Passlib. *readthedocs*. URL: https://passlib.readthedocs.io (дата звернення: 23.05.2026).

43. pytest. *docs.pytest.* URL: https://docs.pytest.org (дата звернення: 23.05.2026).

44. Wei J., Wang X., Schuurmans D., Bosma M., Ichter B., Xia F., Chi E., Le Q., Zhou D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *arXiv:2201.11903*. 2022. URL: https://arxiv.org/abs/2201.11903 (дата звернення: 23.05.2026).

ДОДАТКИ

Додаток А

Програмні коди

Повний програмний код розробленої інтелектуальної системи розміщений у відкритих репозиторіях GitHub. Репозиторій структурно поділений на дві основні частини – серверну (backend) та клієнтську (frontend).

Серверна частина системи, реалізована на стеку Python + FastAPI + SQLAlchemy + SQLite: https://github.com/Pasha-Shevchuk/ai_moderation_module

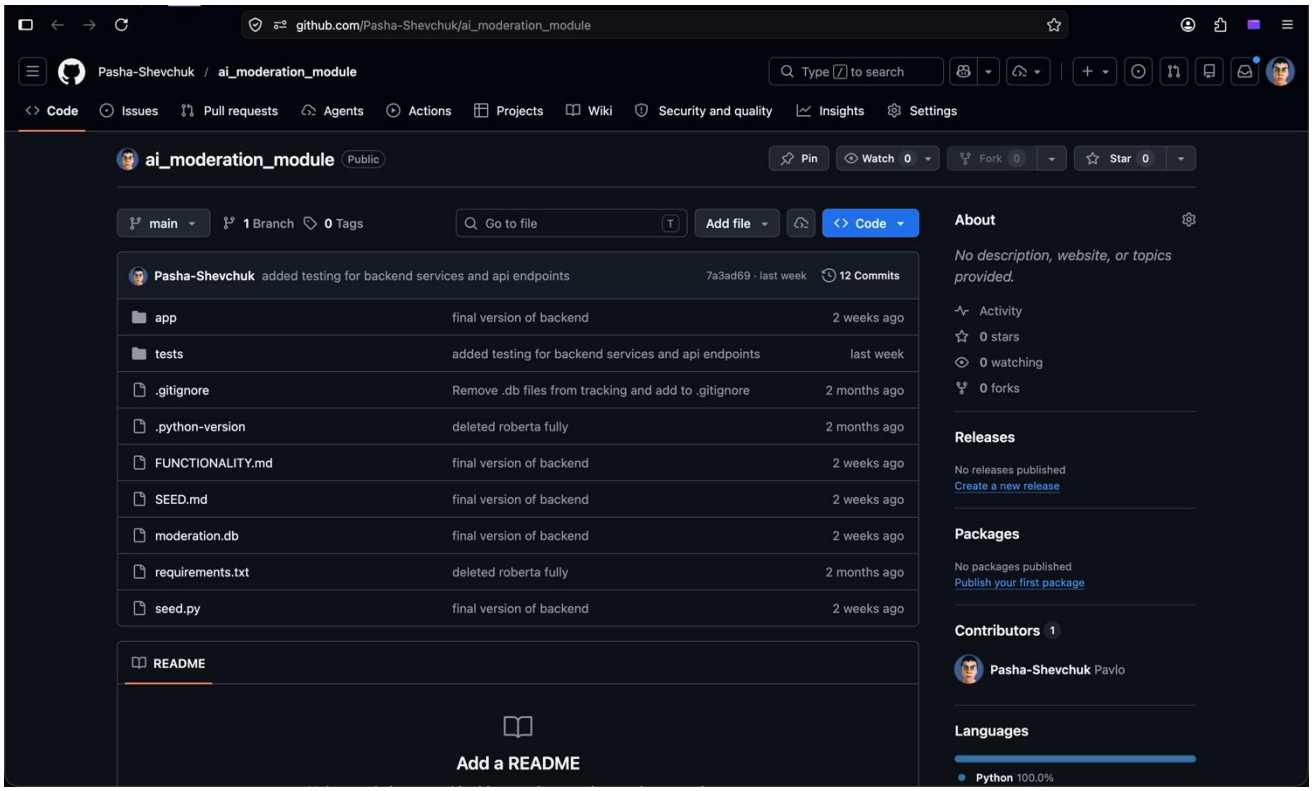

Рисунок А.1 – Знімок репозиторію серверної частини

Клієнтська частина системи, реалізована на стеку React + Vite: https://github.com/Pasha-Shevchuk/ai_moderation_frontend

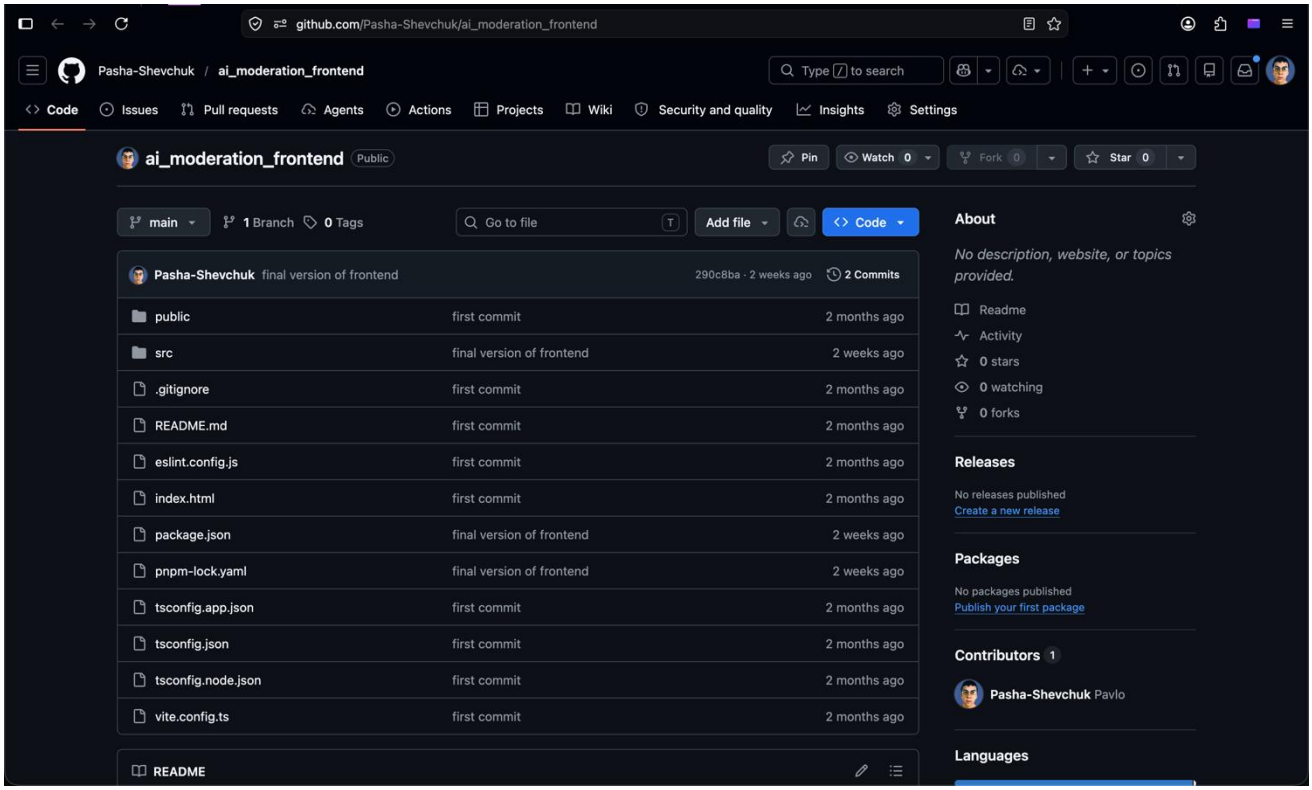

Рисунок А.2 – Знімок репозиторію клієнтської частини

Структура серверного репозиторію включає такі ключові каталоги:

– /app/api – маршрутизатори FastAPI з реалізацією endpoint-ів /analyze, /logs, /auth, /api-keys, /webhooks;

– /app/services – сервісні класи прикладної логіки: Analyzer, WebhookDispatcher, AuthService;

– /app/models – ORM-моделі SQLAlchemy для п’яти сутностей бази даних;

– /app/schemas – Pydantic-моделі контрактів API;

– /app/core – конфігурація застосунку, JWT-налаштування, ініціалізація бази даних;

– /tests – автоматизовані тести pytest;

– /README.md – повна документація з інструкціями зі встановлення та запуску.

Структура клієнтського репозиторію включає:

– /src/pages – компоненти основних сторінок інтерфейсу: автентифікація, playground, журнал, керування ключами та webhook, дашборд;

- /src/components – допоміжні React-компоненти;
- /src/services – клієнтські обгортки для взаємодії з REST API;
- /vite.config.js – конфігурація збирача Vite.

Для локального розгортання необхідно: клонувати обидва репозиторії, встановити Python 3.11 та Node.js 20 LTS, виконати pip install -r requirements.txt та npm install у відповідних каталогах, налаштувати змінні середовища (GROQ_API_KEY, JWT_SECRET, DATABASE_URL), запустити серверну частину командою uvicorn app.main:app, клієнтську – командою npm run dev. Детальна інструкція наведена у файлах README.md кожного з репозиторіїв.

Додаток Б

Презентаційний матеріал

Мета та задачі дослідження

ОБ'ЄКТ ДОСЛІДЖЕННЯ

Процес виявлення суїцидальних намірів у текстовому контенті

ПРЕДМЕТ ДОСЛІДЖЕННЯ

Методи та засоби обробки природної мови для виявлення суїцидальних намірів

МЕТА

Підвищення ефективності процесу виявлення суїцидальних намірів у повідомленнях користувачів засобами обробки природної мови

Задачі для досягнення мети

1. Аналіз предметної області автоматизованого виявлення суїцидальних намірів
2. Розробка методу виявлення суїцидальних намірів засобами NLP
3. Реалізація інтелектуальної системи виявлення суїцидальних намірів
4. Тестування методу з використанням розробленої системи

Мета та задачі дослідження

ОБ'ЄКТ ДОСЛІДЖЕННЯ

Процес виявлення суїцидальних намірів у текстовому контенті

ПРЕДМЕТ ДОСЛІДЖЕННЯ

Методи та засоби обробки природної мови для виявлення суїцидальних намірів

МЕТА

Підвищення ефективності процесу виявлення суїцидальних намірів у повідомленнях користувачів засобами обробки природної мови

Задачі для досягнення мети

1. Аналіз предметної області автоматизованого виявлення суїцидальних намірів
2. Розробка методу виявлення суїцидальних намірів засобами NLP
3. Реалізація інтелектуальної системи виявлення суїцидальних намірів
4. Тестування методу з використанням розробленої системи

Аналіз існуючих засобів модерації

Жоден з існуючих засобів не задовольняє одночасно всім вимогам для україномовного контенту

Засіб	Доступ	Гранулярність ризику	Інтерпретованість	Українська мова
Crisis Text Line	Закритий	Висока	Висока	—
OpenAI Moderation API	Публічний API	Бінарна	—	Часткова
Perspective API	Публічний API	Бінарна	—	Часткова
Azure AI Content Safety	Публічний API	8 рівнів	Низька	Часткова
Розроблений прототип	Open-source	3 рівні + score	key_phrases + reasoning	Повна ✓

Висновок: жоден засіб не поєднує гранулярність ризику, інтерпретованість та підтримку української — це обґрунтовує розробку власного рішення на основі LLM у zero-shot режимі.

Формалізація задачі та метод

Постановка задачі

Вхід: $M = \{m_1, m_2, \ldots, m_n\}$
послідовність повідомлень діалогу

Вихід: $R = (s, \ell, K, \rho)$
s — probability_score ∈ [0, 1]
ℓ — рівень {low, medium, high}
K — ключові фрази
ρ — обґрунтування

Дискретизація score → level

low	medium	high
0,0 — 0,4	0,4 — 0,7	0,7 — 1,0

$\alpha_1 = 0{,}4 \cdot \alpha_2 = 0{,}7$ *(нерівновіддалені: FN дорожчий за FP)*

Пайплайн методу

1. Валідація вхідної послідовності
2. Формування zero-shot промпту з JSON-схемою
3. Виклик LLM через Groq Cloud (response_format=json_object)
4. Парсинг JSON-відповіді моделі
5. Дискретизація probability_score у рівень ризику

Архітектура моделі gpt-oss / llama-70b

Decoder-only Transformer з механізмом експертної маршрутизації Mixture-of-Experts

Стек обробки

Вхідний текст
"Я не бачу сенсу..."

▼

Токенізатор BPE
словник ~200k токенів

▼

Token Embeddings + RoPE
[seq_len × d_model]

▼

Multi-Head Self-Attention
causal mask

▼

Mixture-of-Experts MLP
SwiGLU, top-k routing

▼

Linear + Softmax
розподіл наступного токена

Підтримувані моделі

Модель	Параметри	Експертів	Контекст
gpt-oss-20b	20 B	32	128k
llama-70b ★	70 B	—	128k
gpt-oss-120b	120 B	128	128k

★ — обрана дефолтна модель

Ключові операції

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T/\sqrt{d_k}) \cdot V$$
Scaled dot-product attention

$$\text{MoE}(x) = \Sigma\, G(x)_e \cdot E_e(x),\ e \in \text{TopK}$$
Експертна маршрутизація top-k

Реалізація прототипу

BACKEND	FRONTEND	INTEGRATIONS	TESTING
• Python 3.11	• React	• Groq Cloud API	• pytest
• FastAPI	• Vite	• Telegram / Discord	• pytest-asyncio
• SQLAlchemy	• TypeScript	• Slack / Generic	• JWT auth
• SQLite			

Ключові REST API endpoints

`POST /api/v1/analyze`	Основний аналіз повідомлення
`POST /api/v1/analyze/compare`	Паралельний прогін 2-5 моделей
`GET/PATCH /logs`	Журнал + мітки human_label
`CRUD /api-keys, /webhooks`	Керування інтеграціями

Тестовий набір та сценарії експериментів

60
повідомлень
україномовний експертно розмічений набір

3
рівні ризику
по 20 повідомлень на клас: low / medium / high

0,89
Cohen's kappa
інтер-розмічувальна узгодженість (substantial)

Шість сценаріїв експериментальних досліджень

1. **Основний експеримент** — *4 незалежні прогони, μ ± σ*
2. **Порівняння моделей** — *llama-70b vs gpt-oss-20b/120b*
3. **Якісний аналіз помилок** — *групування за типами*
4. **Калібрування ECE** — *reliability diagram*
5. **Чутливість до temperature** — *5 значень 0–1*
6. **Pareto-аналіз** — *якість vs латентність*

Результати експериментальних досліджень

Accuracy	Macro-F1	Binary-Accuracy	Recall (high)
0,900	0,900	0,950	0,938
± 0,014	± 0,014	± 0,014	± 0,025

Порівняння моделей (Pareto-границя)

Модель	Macro-F1	Латентність
gpt-oss-20b	0,812	620 мс
llama-70b ★	0,900	1340 мс
gpt-oss-120b	0,917	2180 мс

Ключові висновки

- ✓ Помилок між крайніми класами (low ↔ high) не зафіксовано
- ✓ Жодного пропущеного кризового випадку (high → low)
- ✓ ECE = 0,050 — модель добре відкалібрована
- ✓ llama-70b на Pareto-границі — оптимальний баланс

60 повідомлень × 4 прогони × 3 моделі = повноцінна емпірична валідація методу

Висновки та перспективи

Мету роботи досягнуто. Усі поставлені задачі виконано.

НАУКОВА НОВИЗНА

Адаптація zero-shot класифікації великими мовними моделями з примусово структурованим JSON-виходом до задачі виявлення суїцидальних намірів в україномовному контенті без потреби у власному розміченому датасеті та донавчанні.

ПРАКТИЧНЕ ЗНАЧЕННЯ

Реалізований прототип з REST API та webhook-сповіщеннями (Telegram, Discord, Slack) готовий до інтеграції в українські онлайн-платформи та сервіси психологічної підтримки.

ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

- Створення україномовного клінічно валідованого датасету
- Підкласифікація: суїцид / самопошкодження / прохання допомоги
- Consensus-voting між кількома LLM-моделями
- A/B-тестування на реальних онлайн-платформах
- Адаптивні пороги дискретизації на основі feedback

Дякую за увагу!

Wed Jun 17 08:56:06 EEST 2026, Петровський Сергій Степанович, Хмельницький національний університет, ХНУ

Anti-Plagiarism (http://ap.km.ua) v-16.718

Максимальне співпадіння з одним документом 4.0%

Словники перевірки: UA, US, RU. **Помилок в документах: 17%**

	Документ		Сумарний збіг по Базі Даних	
ID: 275687 Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод автоматизованого виявлення суїцидальних намірів у повідомленнях користувачів засобами обробки природної мови Додано в БД: 2026-06-17 Автора: Павло ШЕВЧУК Керівники: Валерія КЛІМЕНКО Консультанти: Опоненти:	Символи	Лексеми	Символи	Лексеми
	67162	595	3708 (6%)	54 (9%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Протокол аналізу звіту подібності науковим керівником

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

Автор: Павло ШЕВЧУК

Співавтор:

Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод автоматизованого виявлення суїцидальних намірів у повідомленнях користувачів засобами обробки природної мови

Науковий керівник: Валерія КЛІМЕНКО, асистент каф. КН

Підрозділ: Кафедра комп'ютерних наук

Коефіцієнт подібності 1:5.79%

Коефіцієнт подібності 2:2.6%

Мікропробіли: 0

Заміна букв: 6

Інтервали: 0

Білі знаки: 7

Дата створення звіту: 2026-06-16 19:40:51.0

Після аналізу Звіту подібності констатую наступне:

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

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

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

Обгрунтування:

2026-06-17

Дата

експерт Первовський Р.Р.

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи *Метод автоматизованого виявлення суїцидальних намірів у повідомленнях користувачів засобами обробки природної мови*
Автор *студент групи КН-22-2 Павло ШЕВЧУК*
Освітня програма *Комп'ютерні науки*
Рівень вищої освіти *перший (бакалаврський)*
Спеціальність *122 – Комп'ютерні науки*
Науковий керівник: *асистент каф. КН Валерія КЛІМЕНКО*

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмними засобами комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	***відповідає***
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	***відсутні***

Підтвердження:

Запозичення, виявлені в роботі Павла Шевчука, не є плагіатом, оскільки: запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; до запозичень входять фрагменти, які не мають авторства і містять поширені конструкції та загальновідомі терміни, скорочення. Рівень подібності не перевищує допустимої межі. Таким чином, робота є законною та приймається до захисту.

Обсяг запозичень, визначений системами виявлення збігів/ідентичності/схожості:
- за системою Anti-Plagiarism: 4%;
- за системою StrikePlagiarism КП1: 5.79%.

17.06.2026

Завідувач кафедри _____ *Олександр БАРМАК*

Гарант освітньої програми _____ *Олександр МАЗУРЕЦЬ*

Керівник кваліфікаційної роботи _____ *Валерія КЛІМЕНКО*

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ

Кафедра комп'ютерних наук

ВІДГУК НАУКОВОГО КЕРІВНИКА

на кваліфікаційну роботу бакалавра

студента *гр. КН-22-2 Шевчука Павла Олександровича*

за темою *Метод автоматизованого виявлення суїцидальних намірів у повідомленнях користувачів засобами обробки природної мови*

1. Актуальність теми

Тема кваліфікаційної роботи є актуальною, оскільки пов'язана з розробленням інтелектуальних засобів аналізу текстових повідомлень для виявлення ознак суїцидальних намірів користувачів. Зростання обсягів цифрової комунікації та необхідність своєчасного виявлення потенційно небезпечних повідомлень зумовлюють потребу в автоматизованих методах обробки природної мови. Практична спрямованість дослідження, орієнтація на реальні текстові дані та використання сучасних підходів штучного інтелекту підтверджують доцільність і своєчасність обраної тематики.

2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки

Зміст роботи відповідає предметній області спеціальності 122 «Комп'ютерні науки». У дослідженні розглянуто постановку прикладної задачі, аналіз предметної області, роботу з текстовими даними, застосування методів обробки природної мови, розроблення програмного забезпечення та оцінювання отриманих результатів. Робота демонструє здатність здобувача використовувати сучасні методи комп'ютерних наук для вирішення актуальної практичної задачі.

3. Професійні та особистісні якості бакалавра

Під час виконання кваліфікаційної роботи здобувач проявив відповідальність, наполегливість та здатність до самостійного опрацювання науково-технічних джерел. Студент продемонстрував належний рівень фахової підготовки, уміння аналізувати існуючі підходи, приймати обґрунтовані технічні рішення та послідовно реалізовувати поставлені завдання. Зауваження та рекомендації враховувалися своєчасно й конструктивно.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Кваліфікаційна робота виконана здобувачем самостійно. Автор самостійно здійснив аналіз предметної області, підбір і опрацювання даних, реалізацію програмної складової та проведення експериментальних досліджень.

5. Ступінь оволодіння методами дослідження

Здобувач продемонстрував належний рівень володіння методами наукового дослідження. У роботі коректно використано методи аналізу текстових даних, інструменти обробки природної мови та засоби оцінювання якості моделей машинного навчання. Автор не лише застосував відповідні технології, а й обґрунтував доцільність їх використання для розв'язання поставленої задачі.

6. Повнота та якість розкриття теми роботи

Тему роботи розкрито повно та послідовно. У роботі наведено аналіз предметної області, обґрунтовано вибір підходів до розв'язання задачі, описано розроблений метод, програмну реалізацію та результати експериментальної перевірки. Поданий матеріал дозволяє сформувати цілісне уявлення про виконане дослідження та отримані результати.

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

Матеріал викладено логічно та послідовно. Структура роботи забезпечує зрозумілий перехід від постановки проблеми до опису запропонованого рішення та аналізу результатів. Автор коректно використовує фахову термінологію, а аргументація основних положень є достатньо обґрунтованою. Текст відповідає вимогам наукового стилю викладу.

8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин

Отримані результати мають практичну цінність і можуть бути використані під час створення інформаційних систем моніторингу текстового контенту та інструментів підтримки прийняття рішень у сфері цифрової безпеки.

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

Вважаю, що робота заслуговує на позитивну оцінку, а її автор може бути допущений до захисту з рекомендованою оцінкою « відмінно *».*

Керівник _____ *асистент каф. КН Валерія КЛІМЕНКО*

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ

Кафедра комп'ютерних наук

РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента *гр. КН-22-2 Шевчука Павла Олександровича*
за темою: *Метод автоматизованого виявлення суїцидальних намірів у повідомленнях користувачів засобами обробки природної мови*

1. Актуальність обраної теми

Тематика кваліфікаційної роботи є актуальною та має виражену практичну спрямованість. Дослідження присвячене автоматизованому виявленню суїцидальних намірів у текстових повідомленнях користувачів, що належить до складних і суспільно значущих задач аналізу природної мови. Зростання обсягів цифрової комунікації обумовлює потребу у створенні інтелектуальних засобів для своєчасного виявлення потенційно небезпечного контенту. Актуальність роботи визначається не лише використанням сучасних методів штучного інтелекту, а й орієнтацією на вирішення реальної прикладної проблеми, результати якої можуть бути корисними для розроблення спеціалізованих інформаційних систем.

2. Повнота розкриття мети та завдань роботи

Поставлену мету в роботі досягнуто, а сформульовані завдання виконано в повному обсязі. Автор послідовно переходить від аналізу предметної області до обґрунтування вибраного підходу, реалізації запропонованого рішення та оцінювання його ефективності. Між метою дослідження, поставленими завданнями та отриманими результатами простежується чіткий взаємозв'язок, що свідчить про логічність побудови роботи та цілісність проведеного дослідження.

3. Зміст кожного розділу роботи

Структура роботи є логічною та відповідає характеру дослідження. У першому розділі проведено аналіз предметної області та сучасних підходів до розв'язання поставленої задачі. Другий розділ присвячено розробленню методу автоматизованого аналізу текстових повідомлень і обґрунтуванню використаних засобів обробки природної мови. У третьому розділі наведено особливості програмної реалізації, результати експериментальної перевірки та їх аналіз. Розділи послідовно доповнюють один одного та формують завершену структуру дослідження.

4. Оцінка розробленої інформаційної системи, її практична цінність

Розроблена інформаційна система має практичну цінність і демонструє можливість використання сучасних методів аналізу текстових даних для вирішення прикладних задач. Запропонований підхід може бути використаний як основа для створення сервісів моніторингу текстового контенту та підтримки прийняття рішень у сфері психологічної безпеки. Важливою перевагою роботи є можливість подальшого розвитку системи шляхом розширення набору даних, удосконалення моделей аналізу та інтеграції в спеціалізовані програмні комплекси.

5. Якість оформлення кваліфікаційної роботи бакалавра

Робота оформлена відповідно до встановлених вимог. Матеріал викладено послідовно, структуровано та зрозуміло. Автор коректно використовує спеціалізовану термінологію, а наведені пояснення дозволяють простежити логіку прийнятих рішень і зрозуміти особливості реалізації запропонованого підходу. Загалом оформлення сприяє позитивному сприйняттю роботи та полегшує ознайомлення з її змістом.

6. Недоліки кваліфікаційної роботи бакалавра

До недоліків роботи можна віднести недостатньо детальне висвітлення окремих аспектів експериментального дослідження. Проте наведене зауваження не має принципового характеру та не впливає на загальну позитивну оцінку роботи.

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

Вважаю, що робота заслуговує на високу оцінку, а її автор може бути допущений до захисту. Рекомендована оцінка – « відмінно

Рецензент _____ Бедрецова І.П.