МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

ШКЛЯРУК Назарій Анатолійович

Програмний модуль виявлення емоцій у текстових повідомленнях на основі рекурентних нейронних мереж / Software Module for Emotion Detection in Text Messages Based on Recurrent Neural Networks

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

Кваліфікаційна робота

Виконав студент групи КН-42
Н. А. Шклярук

Науковий керівник:
д.т.н., професор, М. П. Комар

Кваліфікаційну роботу допущено
до захисту
«___»_____ 2026 р.

Завідувач кафедри
_____ Н.В. Дзюбановська

Тернопіль – 2026

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність: 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Н.В. Дзюбановська
«____» _____20__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
ШКЛЯРУКУ Назарію Анатолійовичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи
Програмний модуль виявлення емоцій у текстових повідомленнях на основі рекурентних нейронних мереж / Software Module for Emotion Detection in Text Messages Based on Recurrent Neural Networks
керівник роботи д.т.н., професор М. П. Комар
затверджені наказом по університету від 01 грудня 2025 року № 894.
2. Строк подання студентом закінченої кваліфікаційної роботи 25 травня 2026 р.
3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.
4. Основні питання, які потрібно розробити
- проаналізувати предметну область виявлення емоцій у текстових повідомленнях та визначити основні особливості цієї задачі;
- дослідити існуючі методи і моделі класифікації емоцій у тексті;
- проаналізувати набори даних і способи подання текстової інформації, придатні для навчання моделей виявлення емоцій;
- спроєктувати архітектуру програмного модуля та сформувати його алгоритмічне забезпечення;
- реалізувати моделі виявлення емоцій на основі LSTM, BiLSTM і GRU та інтегрувати їх у програмний модуль;
- провести обчислювальний експеримент, оцінити якість роботи моделей та виконати їх порівняльний аналіз;
- провести тестування програмного модуля на прикладах текстових повідомлень.
5. Перелік графічного матеріалу в роботі
- архітектура програмного модуля;
- схема взаємодії компонентів програмного модуля;
- схема алгоритмічного забезпечення програмного модуля;
- схема алгоритму попереднього оброблення текстових повідомлень перед навчанням моделей;

– схема алгоритму навчання моделі виявлення емоцій у текстових повідомленнях;

– схема алгоритму тестування та оцінювання моделі виявлення емоцій у текстових повідомленнях.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01 грудня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи	до 01.01. 2026 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.02. 2026 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 15.03.2026 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05. 2026 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2026 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів	до 25.05.2026 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту	до 30.05.2026 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 10.06.2026 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 10.06. 2026 р.	

Студент_____ Н. А. Шклярук
підпис

Керівник роботи_____ д.т.н., професор М. П. Комар
підпис

АНОТАЦІЯ

Кваліфікаційна робота на тему «Програмний модуль виявлення емоцій у текстових повідомленнях на основі рекурентних нейронних мереж» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом в 70 сторінок і містить 16 ілюстрацій, 8 таблиць, 2 додатки та 43 використаних джерела.

Метою роботи є розроблення програмного модуля виявлення емоцій у текстових повідомленнях на основі рекурентних нейронних мереж та дослідження ефективності моделей під час класифікації емоційного забарвлення тексту.

Методи дослідження включають системний аналіз, оброблення природної мови, попереднє оброблення текстових даних, токенізацію, векторизацію, машинне навчання, моделювання, експериментальне тестування та оцінювання якості класифікації емоцій.

Розроблено програмний модуль виявлення емоцій у текстових повідомленнях на основі рекурентних нейронних мереж, який забезпечує завантаження текстових даних, їх очищення та нормалізацію, токенізацію, формування числового подання, навчання моделей LSTM, BiLSTM і GRU, класифікацію повідомлень та оцінювання результатів за метриками accuracy, precision, recall і F1-score.

Результати дослідження можуть бути використані для автоматизованого аналізу текстових повідомлень, визначення емоційного забарвлення, моніторингу інформаційного простору, модерації контенту та вдосконалення систем оброблення природної мови.

Ключові слова: ВИЯВЛЕННЯ ЕМОЦІЙ, ТЕКСТОВІ ПОВІДОМЛЕННЯ, РЕКУРЕНТНІ НЕЙРОННІ МЕРЕЖІ, LSTM, BILSTM, GRU, NLP, КЛАСИФІКАЦІЯ.

ANNOTATION

Qualification work on the topic «Software Module for Emotion Detection in Text Messages Based on Recurrent Neural Networks» for Bachelor's degree on speciality 122 «Computer Science» educational and professional program «Computer Science» is written on 70 pages and it contains 16 figures, 8 tables, 2 annexes and 43 sources.

The purpose of the work is to develop a software module for detecting emotions in text messages based on recurrent neural networks and to study the effectiveness of models in classifying the emotional tone of text.

The research methods include system analysis, natural language processing, preprocessing of text data, tokenization, vectorization, machine learning, modeling, experimental testing, and evaluation of the quality of emotion classification.

A software module for detecting emotions in text messages based on recurrent neural networks has been developed. It provides loading of text data, cleaning and normalization, tokenization, formation of numerical representation, training of LSTM, BiLSTM, and GRU models, classification of messages, and evaluation of results using accuracy, precision, recall, and F1-score metrics.

The research results can be used for automated analysis of text messages, determination of emotional tone, monitoring of the information space, content moderation, and improvement of natural language processing systems.

Keywords: EMOTION DETECTION, TEXT MESSAGES, RECURRENT NEURAL NETWORKS, LSTM, BILSTM, GRU, NLP, CLASSIFICATION.

ЗМІСТ

ВСТУП

У сучасному цифровому середовищі текстові повідомлення стали одним із головних каналів комунікації між людьми, організаціями та інформаційними системами. Соціальні мережі, месенджери, форуми, сервіси підтримки користувачів, освітні платформи та електронні медіа щоденно генерують значні обсяги текстових даних, що містять не лише фактичну інформацію, а й емоційне забарвлення. Саме емоційна складова повідомлень часто визначає наміри автора, характер взаємодії, рівень задоволеності, конфліктність висловлювання або реакцію на певні події. У зв'язку з цим задача автоматизованого виявлення емоцій у тексті набула значної практичної цінності та стала окремим напрямом досліджень у галузі оброблення природної мови і штучного інтелекту [1, 5, 24].

Актуальність теми зумовлена кількома чинниками. По-перше, ручний аналіз емоційного змісту великих потоків текстових повідомлень є трудомістким, повільним і залежить від суб'єктивної інтерпретації. По-друге, сучасні інформаційні системи потребують засобів, здатних автоматично враховувати емоційний контекст повідомлень під час формування рекомендацій, реагування на звернення користувачів, виявлення деструктивної риторики, модерації контенту та аналітичного опрацювання громадської думки [1, 9, 28]. По-третє, підвищення доступності засобів машинного навчання та глибоких нейронних мереж створило умови для практичного впровадження таких рішень у програмні модулі прикладного призначення [5, 18, 26].

Виявлення емоцій у тексті є складнішою задачею, ніж звичайний аналіз тональності. Якщо під час аналізу тональності зазвичай визначається загальна полярність висловлювання, то в задачі emotion detection необхідно встановити конкретний емоційний стан або емоційний клас, наприклад радість, страх, сум, гнів, провину чи відразу. Такий підхід потребує точнішого урахування контексту, послідовності слів, синтаксичних зв'язків і семантичних особливостей висловлювання [1, 21, 22]. Додаткову складність формують багатозначність природної мови, іронія, сарказм, скорочення, неформальна лексика, змішані емоції та залежність інтерпретації від предметної області [5, 24,

31].

У наукових публікаціях запропоновано різні підходи до розв’язання цієї задачі. Для виявлення емоцій у тексті використовувалися лексикон-орієнтовані методи, класичні алгоритми машинного навчання, згорткові нейронні мережі, рекурентні нейронні мережі, моделі з механізмами уваги, а також трансформерні архітектури [1, 5, 7, 12, 27, 34]. Водночас рекурентні нейронні мережі зберігають практичну доцільність у прикладних задачах, оскільки дають змогу ефективно моделювати послідовну природу тексту, враховувати попередній контекст і забезпечувати прийнятну якість класифікації за відносно помірної складності реалізації [4, 8, 10, 11, 23, 30, 32]. Особливий інтерес становлять модифікації LSTM, BiLSTM і GRU, які широко застосовуються для текстової класифікації та демонструють придатність до роботи з послідовними даними [8, 10, 11, 23, 30].

Відомі рішення підтверджують перспективність використання глибокого навчання для аналізу емоцій у текстових повідомленнях, однак не усувають повністю низку практичних проблем. Серед них слід виокремити залежність результатів від способу подання тексту, обраного набору даних, архітектури моделі, параметрів навчання та якості попереднього оброблення повідомлень [1, 5, 13, 24]. Крім того, у прикладному аспекті важливим залишається не лише досягнення прийнятної точності класифікації, а й створення цілісного програмного модуля, у якому поєднано завантаження та підготовку даних, навчання моделей, оцінювання їхньої якості та тестування на нових текстах.

Науково-прикладний інтерес до теми посилюється також розвитком відкритих корпусів і наборів даних для виявлення емоцій. У сучасних дослідженнях використовуються як класичні текстові корпуси, так і спеціалізовані набори даних із розміткою емоцій, зокрема для повідомлень із соціальних мереж, коротких текстів і розмовного мовлення [6, 13, 20, 21, 25]. Це створює підґрунтя для порівняння моделей і побудови програмних засобів, здатних працювати з реальними текстовими повідомленнями. Окреме значення має і використання емоційних лексиконів та способів векторного подання слів, що підвищують змістову інформативність вхідних даних [11, 16, 21, 27].

Отже, розроблення програмного модуля виявлення емоцій у текстових

повідомленнях на основі рекурентних нейронних мереж є актуальним і практично значущим завданням.

Метою роботи є розроблення програмного модуля виявлення емоцій у текстових повідомленнях на основі рекурентних нейронних мереж та дослідження ефективності моделей під час класифікації емоційного забарвлення тексту.

Для досягнення поставленої мети сформовано такі завдання дослідження:

– проаналізувати предметну область виявлення емоцій у текстових повідомленнях та визначити основні особливості цієї задачі;

– дослідити існуючі методи і моделі класифікації емоцій у тексті;

– проаналізувати набори даних і способи подання текстової інформації, придатні для навчання моделей виявлення емоцій;

– спроєктувати архітектуру програмного модуля та сформувати його алгоритмічне забезпечення;

– реалізувати моделі виявлення емоцій на основі LSTM, BiLSTM і GRU та інтегрувати їх у програмний модуль;

– провести обчислювальний експеримент, оцінити якість роботи моделей та виконати їх порівняльний аналіз;

– провести тестування програмного модуля на прикладах текстових повідомлень.

Об’єктом дослідження є процес автоматизованого виявлення емоцій у текстових повідомленнях засобами методів оброблення природної мови та машинного навчання.

Предметом дослідження є методи, моделі та програмні засоби виявлення емоцій у текстових повідомленнях на основі рекурентних нейронних мереж.

Результати кваліфікаційної роботи апробовані та опубліковані у матеріалах Міжнародної наукової інтернет-конференції «Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення», (м. Тернопіль, Україна, м. Ополе, Польща, 7-8 травня 2026 р.).

Кваліфікаційна робота складається із вступу, трьох розділів, висновків, списку використаних джерел та додатків.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Особливості виявлення емоцій у текстових повідомленнях

Виявлення емоцій у текстових повідомленнях є задачею оброблення природної мови, у межах якої визначається емоційний стан, виражений у тексті. На відміну від аналізу тональності, де встановлюється переважно загальна полярність висловлювання, у цьому випадку необхідно класифікувати конкретні емоції, зокрема радість, гнів, страх, сум або відразу. Це робить задачу складнішою, оскільки подібні за змістом повідомлення можуть мати різне емоційне забарвлення [1, 5, 24].

Актуальність такого аналізу зумовлена поширенням цифрової комунікації. Текстові повідомлення стали основною формою взаємодії в соціальних мережах, месенджерах, сервісах підтримки користувачів і вебплатформах. У таких умовах важливим є не лише зміст повідомлення, а й емоція, яку воно передає. Автоматичне виявлення емоцій дає змогу оцінювати реакцію аудиторії, виявляти агресивну риторику, аналізувати звернення користувачів і підвищувати якість інтелектуальних сервісів [1, 9, 28].

Основна складність полягає в тому, що емоція в тексті часто не виражається прямо. Вона може передаватися через контекст, порядок слів, заперечення, стилістичні конструкції або приховану оцінку. Додаткові труднощі створюють іронія, сарказм, неформальна лексика, скорочення, емодзі та змішані емоції. Через це простий пошук емоційно забарвлених слів не забезпечує достатньої точності [1, 5, 31].

Ще однією особливістю є багатокласовий характер задачі. Кількість емоційних категорій залежить від вибраного набору даних і може бути більшою, ніж у класичному аналізі тональності. Крім того, різні корпуси містять різний баланс класів, що впливає на результати навчання моделі. Саме тому під час оцінювання якості доцільно враховувати не лише accuracy, а й precision, recall та F1-score [6, 13, 21].

Для розв’язання цієї задачі важливим є спосіб подання тексту та вибір моделі. Традиційні підходи на основі словників і ручних ознак мають обмежені

можливості, оскільки слабко враховують контекст. Більш ефективними є нейронні моделі, зокрема рекурентні нейронні мережі, які працюють із послідовностями слів і здатні зберігати контекстну інформацію. Особливе практичне значення мають архітектури LSTM, BiLSTM і GRU, що широко застосовуються в задачах текстової класифікації [10, 11, 23, 27].

У таблиці 1.1 наведено порівняння аналізу тональності та виявлення емоцій. На відміну від аналізу тональності, який дає узагальнену оцінку висловлювання, виявлення емоцій забезпечує детальнішу інтерпретацію емоційного змісту тексту, проте потребує точнішого врахування контексту та складніших моделей класифікації.

Таблиця 1.1 – Порівняння аналізу тональності та виявлення емоцій

Критерій	Аналіз тональності	Виявлення емоцій
Основна мета	Визначення загальної полярності тексту	Визначення конкретного емоційного стану
Тип результату	Позитивна, негативна або нейтральна оцінка	Одна або кілька емоційних категорій
Рівень деталізації	Узагальнений	Деталізований
Типові класи	Positive, Negative, Neutral	Joy, Anger, Sadness, Fear, Disgust, Surprise та інші
Складність задачі	Нижча	Вища
Урахування контексту	Потрібне, але часто в обмеженому обсязі	Критично важливе
Чутливість до мовних нюансів	Помірна	Висока
Вплив іронії та сарказму	Ускладнює аналіз	Суттєво ускладнює класифікацію

Продовження таблиці 1.1.

Вимоги до наборів даних	Можливе використання простіших розмічених корпусів	Потрібні корпуси з детальнішою емоційною розміткою
Практичне застосування	Аналіз відгуків, оцінка ставлення, моніторинг думок	Аналіз психологічного стану, модерація контенту, адаптивні інтерфейси, підтримка користувачів

Отже, виявлення емоцій у текстових повідомленнях є складною, але практично важливою задачею. Її розв'язання потребує врахування мовного контексту, особливостей наборів даних, способів подання тексту та вибору ефективної моделі класифікації. Саме це зумовлює доцільність розроблення програмного модуля на основі рекурентних нейронних мереж.

1.2 Аналіз існуючих методів і моделей класифікації емоцій у тексті

Для класифікації емоцій у текстових повідомленнях застосовуються кілька груп методів, які відрізняються принципом роботи, складністю реалізації та якістю результатів. Найпоширенішими є лексикон-орієнтовані підходи, класичні методи машинного навчання, нейронні мережі глибокого навчання та трансформерні моделі [1, 5, 24, 31].

Лексикон-орієнтовані методи базуються на використанні словників емоційно забарвлених слів. У таких підходах кожне слово або вираз пов'язується з певною емоцією чи рівнем емоційної інтенсивності. Перевагою цього підходу є простота реалізації та зрозумілість отриманого результату. Водночас такі методи недостатньо враховують контекст, порядок слів, заперечення, іронію та приховані смислові зв'язки, тому їхня ефективність для коротких і неформальних повідомлень є обмеженою [1, 20, 22].

Класичні методи машинного навчання передбачають попереднє перетворення тексту в числовий набір ознак. Для цього зазвичай

використовуються частотні характеристики слів, n-грами, TF-IDF або інші статистичні представлення. Далі текст класифікується за допомогою традиційних алгоритмів. Перевагою цього підходу є відносно невисока обчислювальна складність. Недоліком є залежність якості класифікації від ручного добору ознак і слабше врахування семантики тексту [1, 24, 29].

Подальший розвиток отримали моделі глибокого навчання, які автоматично формують ознаки без ручного конструювання. Одним із перших ефективних підходів стали згорткові нейронні мережі. Вони добре виявляють локальні текстові шаблони та є придатними для задач класифікації речень. Проте в задачі виявлення емоцій важливим є не лише наявність окремих індикаторів, а й послідовний контекст, тому лише згорткових архітектур часто недостатньо [3, 11, 12].

У задачах, де необхідно враховувати порядок слів і залежності між частинами висловлювання, більш доцільними є рекурентні нейронні мережі. Їх особливість полягає в тому, що текст обробляється як послідовність, а інформація з попередніх кроків передається далі. Базові RNN стали важливим етапом розвитку моделей для аналізу тексту, однак їхнє використання ускладнюється проблемою зникання градієнта під час роботи з довгими послідовностями. Саме тому практичного поширення набули моделі LSTM і GRU [4, 10].

Модель LSTM містить комірки пам’яті та механізми керування потоком інформації, що дає змогу зберігати важливий контекст на довших фрагментах тексту. Це особливо важливо тоді, коли емоція визначається не окремим словом, а всім висловлюванням. GRU є спрощеною модифікацією LSTM, яка має менше параметрів і часто забезпечує швидше навчання. BiLSTM, своєю чергою, враховує контекст у двох напрямах, що підвищує якість класифікації, коли смисл слова залежить як від попередніх, так і від наступних елементів тексту. Саме ці моделі є найбільш доцільними для побудови програмного модуля в межах бакалаврської роботи [8, 10, 11, 23, 27, 30, 32].

Окрему групу становлять гібридні моделі, у яких поєднуються різні архітектурні рішення. Наприклад, згорткові шари можуть виділяти локальні

шаблони, а рекурентні — моделювати послідовні залежності. Також застосовуються механізми уваги, які дозволяють підсилювати вагу найбільш інформативних слів. Такі підходи часто дають вищу точність, проте є складнішими у проєктуванні та налаштуванні [3, 11, 23, 27].

У сучасних дослідженнях значного поширення набули трансформерні моделі. Вони забезпечують глибше врахування контексту та демонструють високі результати в задачах оброблення природної мови. Разом із тим такі моделі є ресурсоємними, складнішими для навчання та менш зручними для реалізації компактного прикладного модуля. З огляду на це для поставленої задачі доцільно зосередитися саме на рекурентних нейронних мережах, які поєднують достатню якість класифікації, помірну складність реалізації та зрозумілу архітектуру [5, 7, 26, 34].

У таблиці 1.2 подано порівняльну характеристику основних підходів до класифікації емоцій у тексті. На рисунку 1.1 представлено узагальнену класифікацію методів, що використовуються для розв’язання цієї задачі.

Таблиця 1.2 – Порівняльна характеристика методів класифікації емоцій у тексті

Група методів	Принцип роботи	Переваги	Недоліки
Лексикон-орієнтовані	Використання словників емоційно забарвлених слів і правил	Простота, інтерпретованість, швидка реалізація	Слабке врахування контексту, низька стійкість до іронії та сарказму
Класичне машинне навчання	Класифікація тексту за числовими ознаками	Невисока обчислювальна складність, придатність для базових задач	Потреба в ручному доборі ознак, обмежене врахування семантики

Продовження таблиці 1.2.

CNN	Виділення локальних текстових шаблонів згортковими шарами	Ефективність для коротких фрагментів, автоматичне вилучення ознак	Недостатнє врахування довгих залежностей
RNN / LSTM / GRU / BiLSTM	Послідовне опрацювання тексту з урахуванням контексту	Добре працюють із послідовностями, враховують порядок слів	Більша складність навчання порівняно з базовими методами
Transformer / BERT	Контекстне представлення тексту з механізмом уваги	Висока якість, глибоке врахування контексту	Висока ресурсоємність, складніша інтеграція в компактний модуль

Таблиця 1.2 показує, що найкращий баланс між якістю класифікації та складністю реалізації для цієї роботи забезпечують рекурентні нейронні мережі.

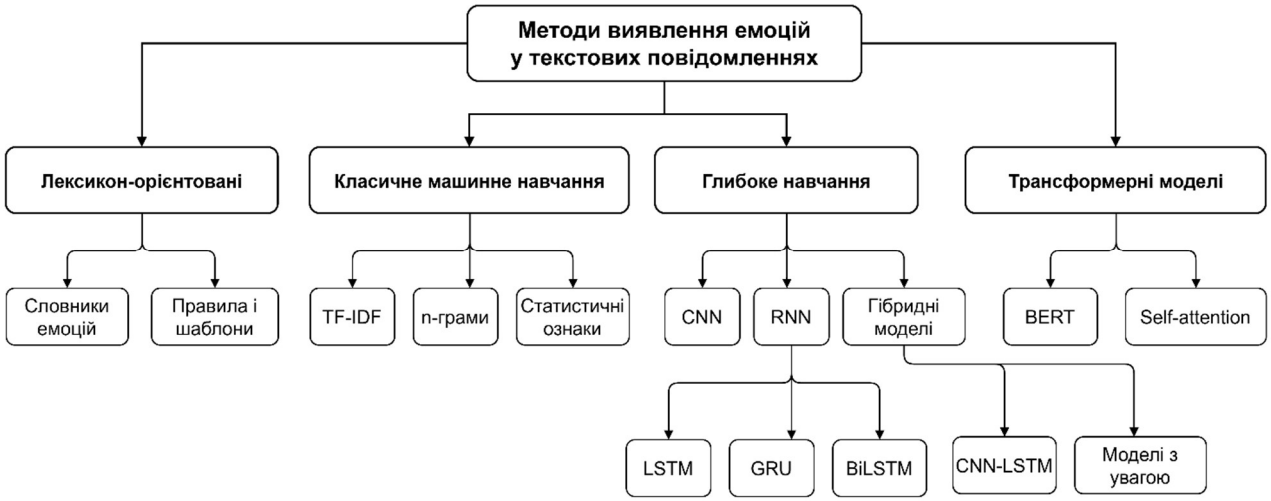


Рисунок 1.2 – Класифікація методів виявлення емоцій у текстових повідомленнях

Отже, проаналізовані методи показали, що для задачі виявлення емоцій у тексті найпростішими є лексикон-орієнтовані та класичні підходи, однак їхні можливості є обмеженими через слабке врахування контексту. Більш ефективними виявилися нейронні моделі, серед яких для цієї роботи найбільший практичний інтерес становлять рекурентні нейронні мережі. Саме вони забезпечують прийнятне поєднання якості класифікації, здатності працювати з послідовністю слів і доцільної складності реалізації.

1.3 Аналіз наборів даних і способів подання текстової інформації

Якість виявлення емоцій у текстових повідомленнях істотно залежить від вибраного набору даних і способу подання тексту. Навіть за однакової архітектури моделі результати класифікації можуть суттєво відрізнятися через різний обсяг корпусу, кількість емоційних класів, баланс між ними, стиль повідомлень і якість розмітки [1, 6, 13, 21, 25].

Для задачі виявлення емоцій використовуються корпуси різного типу. Частина наборів даних містить короткі повідомлення із соціальних мереж, інші — фрагменти діалогів, новинні тексти, відгуки користувачів або спеціально сформовані речення з ручною анотацією. Такі корпуси відрізняються не лише тематикою, а й формою вираження емоцій. У коротких повідомленнях емоція часто передається стисло, неформально та з використанням розмовної лексики, тоді як у більших текстах важливішими стають контекст і послідовність висловлювання [1, 6, 20, 25, 31].

Одним із важливих критеріїв є схема емоційної розмітки. У різних корпусах використовуються як базові набори емоцій, так і розширені системи з більш детальною класифікацією. Наприклад, у частині наборів даних подаються кілька основних емоційних категорій, а в інших — десятки тонших емоційних станів або шкали інтенсивності. Через це моделі, навчені на одному наборі даних, не завжди можна безпосередньо перенести на інший без додаткової адаптації [6, 13, 20, 21].

Окрему проблему становить дисбаланс класів. У багатьох корпусах одні

емоції трапляються значно частіше, ніж інші. Це призводить до того, що модель краще розпізнає поширені категорії, а рідкісні — класифікує гірше. У такому випадку недостатньо орієнтуватися лише на accuracy, оскільки ця метрика може приховувати слабкі результати для окремих класів. Тому під час аналізу наборів даних важливо враховувати розподіл прикладів між емоціями та якість анотації [1, 5, 21].

Не менш важливим є спосіб подання текстової інформації. Найпростішими є підходи, у яких текст перетворюється на набір статистичних ознак, наприклад частот слів, n-грам або TF-IDF-векторів. Такі представлення є зручними для класичних алгоритмів машинного навчання, але вони слабко враховують порядок слів і контекстні зв’язки між ними [1, 24, 29].

Більш інформативним підходом є векторне подання слів, за якого кожне слово описується числовим вектором у семантичному просторі. Такі embeddings дають змогу враховувати змістову близькість між словами та покращують якість вхідних даних для нейронних мереж. На цій основі стали можливими моделі, які працюють із послідовностями слів і зберігають контекст повідомлення [11, 16, 27].

Для рекурентних нейронних мереж спосіб подання тексту має особливе значення, оскільки такі моделі обробляють повідомлення послідовно. Це означає, що текст повинен бути не лише перетворений у числову форму, а й приведений до єдиного формату довжини, очищений від надлишкових елементів та підготовлений до токенізації. Якість попереднього оброблення безпосередньо впливає на результати навчання моделей LSTM, BiLSTM і GRU [1, 8, 11, 23].

У таблиці 1.3 подано порівняльну характеристику основних наборів даних для задачі виявлення емоцій у тексті. Це дасть змогу показати їх відмінності за типом текстів, кількістю класів і сферою застосування.

Таблиця 1.3 – Порівняльна характеристика наборів даних для виявлення емоцій у тексті

Набір даних	Тип текстів	Особливості	Переваги	Обмеження
ISEAR	Короткі текстові описи емоційних ситуацій	Містить кілька базових емоційних категорій; часто використовується в дослідженнях emotion detection	Зручний для навчальних і порівняльних експериментів; добре підходить для багатокласової класифікації	Відносно невеликий обсяг; менш наближений до сучасної онлайн-комунікації
GoEmotions	Короткі онлайн-коментарі	Містить дрібнозернисту емоційну розмітку та велику кількість класів	Великий обсяг даних; добре відображає сучасну неформальну мову	Висока складність класифікації; значна кількість близьких за змістом емоцій
SemEval Affect in Tweets	Повідомлення із соціальних мереж	Орієнтований на аналіз емоцій та інтенсивності емоцій у коротких текстах	Придатний для дослідження емоцій у реальних коротких повідомленнях	Наявність шуму, скорочень, хештегів і нестандартної лексики
XED	Короткі тексти різними мовами	Мультимовний корпус для аналізу емоцій і тональності	Дає змогу досліджувати міжмовне перенесення моделей	Складніша підготовка даних; залежність якості від вибраної мови

Продовження таблиці 1.3.

Twitter-based corpora	Твіти та короткі пости	Відображають емоції в природному онлайн-середовищі	Добре підходять для прикладного аналізу соціальних мереж	Високий рівень шуму, сленгу, іронії та неповних речень
Спеціалізовані корпуси діалогів	Репліки з чатів або діалогів	Орієнтовані на контекстну взаємодію між співрозмовниками	Дають змогу враховувати емоції в комунікативном у процесі	Часто потребують складнішої моделі та більшої контекстної обробки

Отже, вибір набору даних і способу подання тексту є одним із ключових етапів побудови системи виявлення емоцій. Саме ці компоненти визначають, наскільки повно модель зможе враховувати мовний контекст, структуру повідомлення та відмінності між емоційними категоріями

1.4 Постановка задачі дослідження

Проведений аналіз предметної області засвідчив, що виявлення емоцій у текстових повідомленнях належить до актуальних задач оброблення природної мови, оскільки дає змогу автоматично визначати емоційний зміст повідомлень у цифровому середовищі. Практична значущість цієї задачі зумовлена тим, що текстові повідомлення широко використовуються в соціальних мережах, месенджерах, системах підтримки користувачів, освітніх платформах і вебсервісах. У таких умовах важливим є не лише зміст повідомлення, а й емоція, яку воно передає. Саме тому автоматизоване виявлення емоцій розглядається як важлива складова інтелектуального аналізу тексту [1, 5, 24].

У попередніх підрозділах встановлено, що задача класифікації емоцій у тексті є складнішою за звичайний аналіз тональності. Це пояснюється тим, що модель повинна визначати не загальну полярність повідомлення, а конкретний

емоційний стан. Додаткову складність створюють контекстна залежність значення слів, наявність іронії, скорочень, неформальної лексики, неоднозначність мовних конструкцій, а також дисбаланс класів у наборах даних. Отже, для побудови ефективного програмного рішення необхідно враховувати не лише сам текст, а й особливості його попереднього оброблення, способу подання та моделі класифікації [1, 6, 13, 21].

Аналіз існуючих методів показав, що лексикон-орієнтовані підходи та класичні алгоритми машинного навчання можуть бути використані як базові засоби аналізу, однак вони не завжди забезпечують достатню якість під час роботи зі складними або короткими текстовими повідомленнями. Найбільш доцільними для цієї задачі виявилися нейронні моделі, які здатні враховувати послідовність слів і контекст висловлювання. Серед них практичний інтерес становлять рекурентні нейронні мережі, зокрема LSTM, BiLSTM і GRU. Такі моделі дають змогу зберігати контекстну інформацію, працювати з послідовними даними та забезпечують прийнятний баланс між якістю класифікації і складністю реалізації [8, 10, 11, 23, 27].

З огляду на це, у межах кваліфікаційної роботи сформовано задачу розроблення програмного модуля виявлення емоцій у текстових повідомленнях на основі рекурентних нейронних мереж. Розроблений модуль повинен реалізовувати повний цикл оброблення текстових даних: від надходження вхідного повідомлення до формування результату класифікації. Такий підхід дає змогу перейти від теоретичного аналізу моделей до створення практичного програмного засобу, придатного для експериментального дослідження та подальшого використання.

Метою роботи є розроблення програмного модуля виявлення емоцій у текстових повідомленнях на основі рекурентних нейронних мереж та дослідження ефективності моделей LSTM, BiLSTM і GRU під час класифікації емоційного забарвлення тексту.

Для досягнення поставленої мети сформовано такі завдання дослідження:

– проаналізувати предметну область виявлення емоцій у текстових повідомленнях та визначити основні особливості цієї задачі;

– дослідити існуючі методи і моделі класифікації емоцій у тексті та обґрунтувати доцільність використання рекурентних нейронних мереж;

– проаналізувати набори даних і способи подання текстової інформації, придатні для навчання моделей виявлення емоцій;

– спроєктувати архітектуру програмного модуля та сформувати його алгоритмічне забезпечення;

– реалізувати моделі виявлення емоцій на основі LSTM, BiLSTM і GRU та інтегрувати їх у програмний модуль;

– провести обчислювальний експеримент, оцінити якість роботи моделей та виконати їх порівняльний аналіз;

– провести тестування програмного модуля на прикладах текстових повідомлень.

Об’єктом дослідження є процес автоматизованого виявлення емоцій у текстових повідомленнях засобами методів оброблення природної мови та машинного навчання.

Предметом дослідження є методи, моделі та програмні засоби виявлення емоцій у текстових повідомленнях на основі рекурентних нейронних мереж.

2 ПРОЄКТУВАННЯ ПРОГРАМНОГО МОДУЛЯ ВИЯВЛЕННЯ ЕМОЦІЙ У ТЕКСТОВИХ ПОВІДОМЛЕННЯХ

2.1 Проєктування архітектури програмного модуля

Проєктування архітектури програмного модуля виконано з урахуванням поставленої задачі автоматизованого виявлення емоцій у текстових повідомленнях. Основною вимогою до модуля є забезпечення повного циклу оброблення текстових даних: від надходження вхідного повідомлення до отримання результату класифікації. Архітектура має бути достатньо простою для практичної реалізації, але водночас придатною до експериментального порівняння кількох рекурентних моделей [1, 5, 8, 11, 23].

Архітектура програмного модуля повинна мати такі основні компоненти: модуль завантаження даних, модуль попереднього оброблення тексту, модуль векторизації, модуль побудови та навчання моделей, модуль оцінювання результатів і модуль тестування на нових повідомленнях. Така структура забезпечить логічний поділ функцій і спростить реалізацію, перевірку та подальше вдосконалення програмного рішення.

Вхідними даними для модуля є текстові повідомлення, подані у вигляді набору прикладів із відповідними емоційними мітками. На першому етапі дані завантажуються з файлу або підготовленого корпусу текстів. Після цього вони передаються до підсистеми попереднього оброблення, де виконуються очищення тексту, нормалізація, токенізація та приведення повідомлень до формату, придатного для подальшого машинного аналізу. Після завершення цього етапу текстова інформація перетворюється у числове подання, яке може бути використане нейронною мережею.

Ключовим елементом архітектури є модуль моделей виявлення емоцій. У межах цієї роботи він повинен забезпечувати реалізацію трьох архітектур: LSTM, BiLSTM і GRU. Для всіх моделей має використовуватися єдина схема підготовки даних і спільний підхід до оцінювання якості. Це необхідно для того, щоб результати порівняння були коректними й не залежали від відмінностей у вхідних умовах експерименту [8, 10, 11, 23, 32].

Окреме місце в архітектурі займає модуль оцінювання результатів. Його призначення полягає в обчисленні основних метрик якості класифікації та формуванні підсумкових результатів експерименту. Для задачі багатокласового виявлення емоцій доцільно використовувати accuracy, precision, recall і F1-score, оскільки лише загальної точності недостатньо для повноцінного аналізу якості роботи моделі [1, 5, 21]. Результати оцінювання мають бути представлені у зручному для аналізу вигляді, зокрема у формі таблиць і графіків.

У таблиці 2.1 подано функціональне призначення основних компонентів програмного модуля. Це дає змогу чітко зафіксувати роль кожного структурного елемента та показати логіку взаємодії між ними.

Таблиця 2.1 – Основні компоненти архітектури програмного модуля

Компонент	Призначення
Модуль завантаження даних	Імпорт текстових повідомлень і міток класів із підготовленого набору даних
Модуль попереднього оброблення	Очищення, нормалізація, токенізація та підготовка тексту до векторизації
Модуль векторизації	Перетворення текстових повідомлень у числове подання
Модуль моделей	Реалізація та навчання моделей LSTM, BiLSTM і GRU
Модуль оцінювання	Обчислення метрик якості та формування результатів порівняння
Модуль тестування	Перевірка роботи модуля на нових текстових повідомленнях
Модуль збереження результатів	Збереження моделей, прогнозів і підсумкових показників

У таблиці 2.1 наведено основні компоненти програмного модуля та їх функціональне призначення. Наведена структура показує, що архітектура побудована за модульним принципом, а це спрощує як розроблення, так і подальше тестування окремих частин системи.

З погляду організації оброблення даних архітектура має лінійно-послідовний характер. Текстове повідомлення проходить через етапи підготовки, перетворення у векторне подання, подачі до моделі, отримання прогнозу та оцінювання результату. Водночас модульна побудова забезпечує можливість заміни або розширення окремих компонентів без зміни загальної логіки роботи системи. Наприклад, у майбутньому може бути змінено спосіб векторизації, додано нову модель класифікації або розширено блок тестування.

На рисунку 2.1 зображено чотири логічні рівні функціонування програмного модуля: рівень вхідних даних, рівень підготовки тексту, рівень моделей класифікації та рівень вихідних результатів [40].

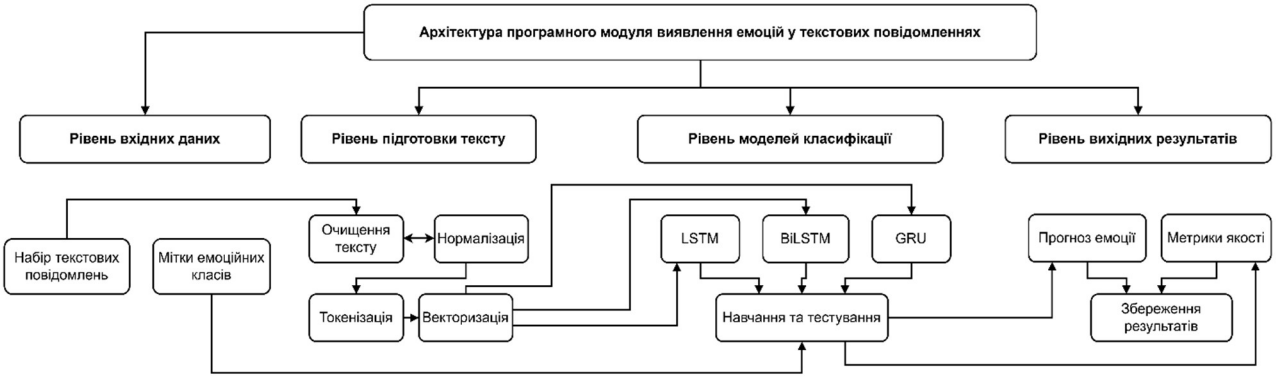


Рисунок 2.1 – Архітектура програмного модуля

Для підвищення наочності також подано рисунок 2.2, на якому зображено структурну схему взаємодії компонентів програмного модуля [40].

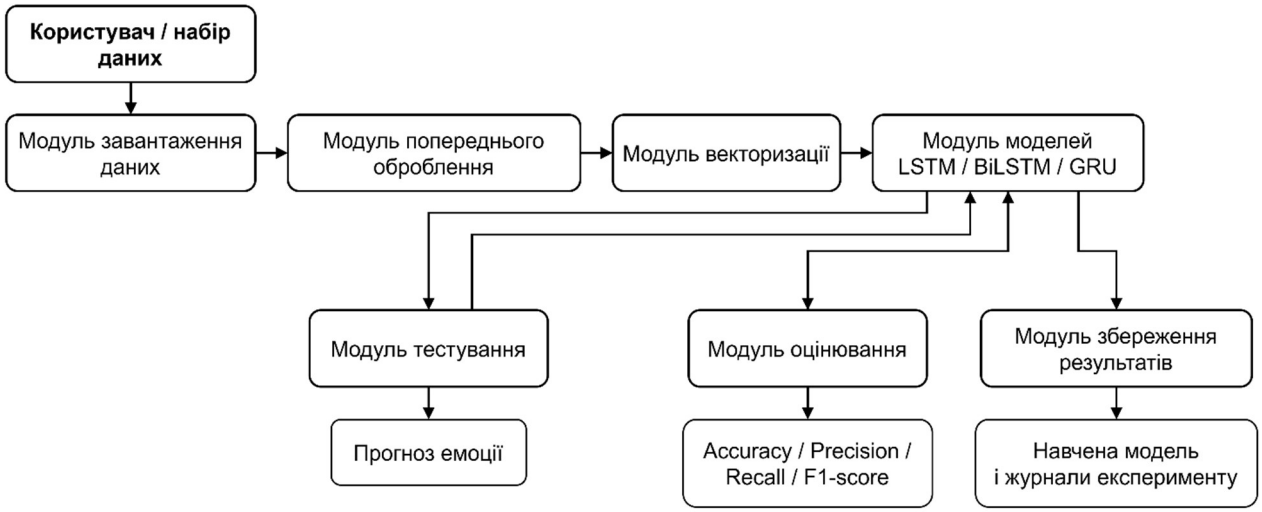


Рисунок 2.2 – Схема взаємодії компонентів програмного модуля

Якщо рисунок 2.1 показує загальну архітектуру, то рисунок 2.2 деталізує потоки даних між окремими модулями.

Отже, у межах підрозділу сформовано архітектуру програмного модуля виявлення емоцій у текстових повідомленнях. Вона базується на модульному принципі, охоплює всі основні етапи оброблення текстових даних і забезпечує можливість реалізації, навчання та порівняння моделей LSTM, BiLSTM і GRU.

2.2 Алгоритмічне забезпечення програмного модуля

Алгоритмічне забезпечення програмного модуля сформовано з урахуванням повного циклу оброблення текстових повідомлень: від надходження вхідних даних до отримання результату класифікації емоцій. Його структура охоплює три основні групи алгоритмів: алгоритм попереднього оброблення тексту, алгоритм навчання моделі та алгоритм тестування й оцінювання результатів. Такий підхід забезпечує узгоджену роботу всіх компонентів модуля та створює основу для коректного порівняння моделей LSTM, BiLSTM і GRU [1, 5, 8, 11, 23].

На рисунку 2.3 подано загальну схему алгоритмічного забезпечення програмного модуля. У межах цієї схеми відображено послідовність переходу від вхідного текстового повідомлення до визначення емоційного класу та формування метрик якості.

Першим етапом є попереднє оброблення текстових повідомлень. Його призначення полягає у приведенні вхідного тексту до уніфікованого формату, придатного для подальшого машинного аналізу. На цьому етапі виконуються очищення тексту від зайвих символів, нормалізація регістру, токенізація, видалення надлишкових елементів та перетворення тексту у послідовність, що може бути подана на вхід нейронній мережі. Саме якість цього етапу значною мірою впливає на стабільність навчання і підсумкову точність класифікації [1, 5, 24].

Алгоритм попереднього оброблення доцільно подати у вигляді послідовності кроків (рисунок 2.4) [40]:

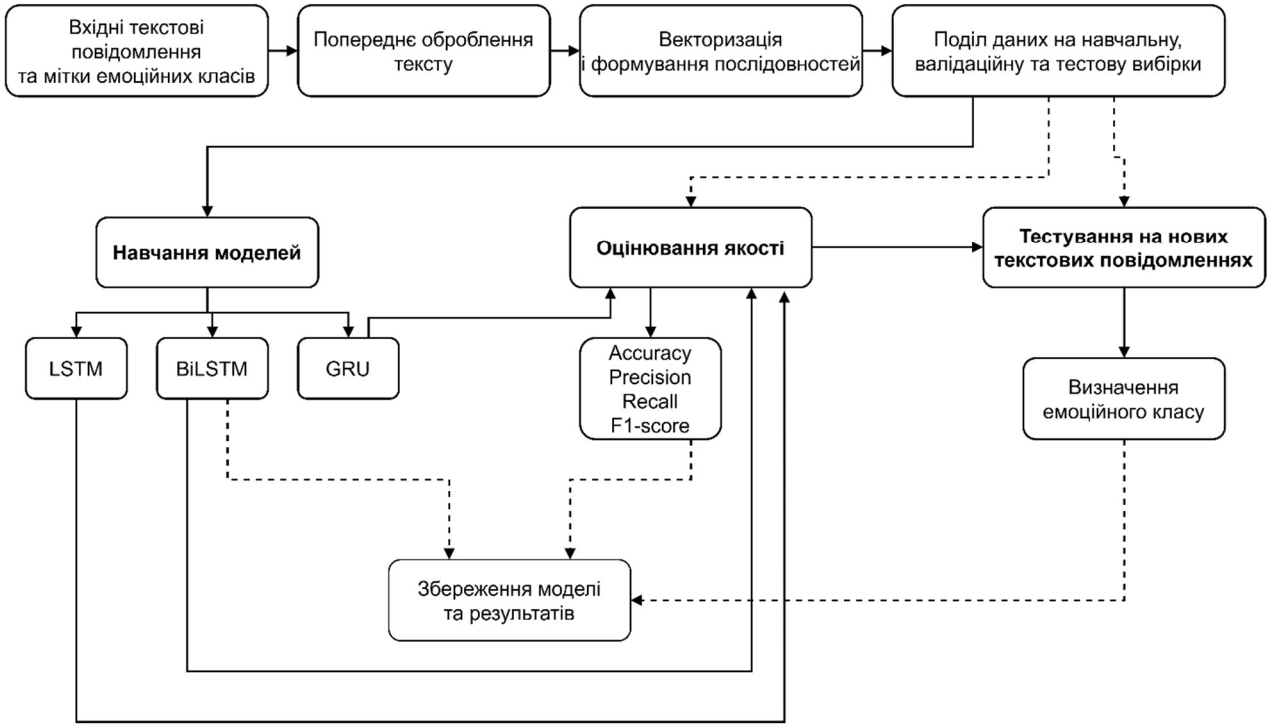


Рисунок 2.3 – Схема алгоритмічного забезпечення програмного модуля

– завантаження текстових повідомлень і міток класів;

– очищення тексту від службових символів, пунктуації та зайвих елементів;

– нормалізація тексту;

– токенізація повідомлень;

– формування словника;

– перетворення тексту на числові послідовності;

– вирівнювання довжини послідовностей;

– поділ набору даних на навчальну, валідаційну та тестову вибірки.

Другим етапом є алгоритм навчання моделей. Після формування числового подання тексту дані надходять до одного з рекурентних класифікаторів. У межах роботи передбачено використання трьох моделей — LSTM, BiLSTM і GRU. Для кожної з них застосовується однакова схема навчання, що забезпечує коректність подальшого порівняння. На цьому етапі виконується побудова архітектури моделі, задання функції втрат, вибір оптимізатора, запуск процесу навчання та збереження найкращих параметрів [8, 10, 11, 23].

Рисунок 2.4 – Схема алгоритму попереднього оброблення текстових повідомлень перед навчанням моделей

Алгоритм навчання моделі можна подати так (рисунок 2.5):

– вибір архітектури моделі;
– ініціалізація параметрів навчання;
– подача навчальних даних на вхід моделі;
– обчислення функції втрат;
– оновлення ваг моделі;
– перевірка якості на валідаційній вибірці;

– збереження найкращої моделі;

– завершення навчання після досягнення заданої кількості епох або критерію зупинки.

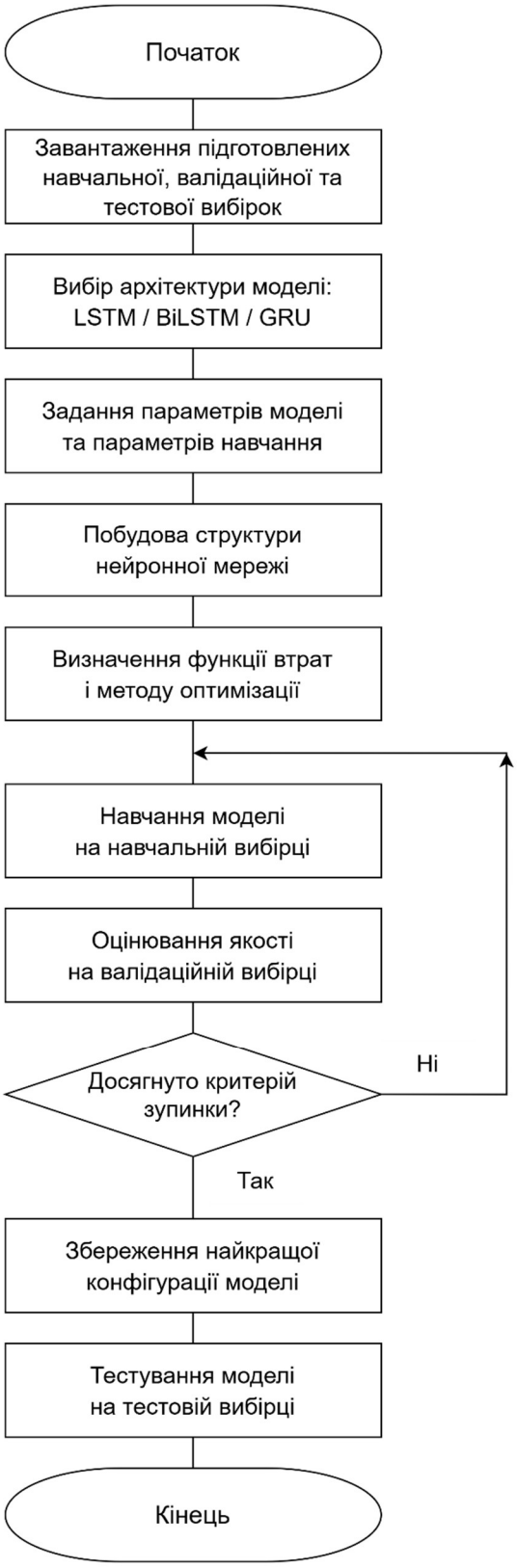


Рисунок 2.5 – Схема алгоритму навчання моделі виявлення емоцій у текстових повідомленнях

Третім етапом є алгоритм тестування та оцінювання. Його завдання полягає у використанні навченої моделі для класифікації нових текстових повідомлень і визначенні якості роботи системи. Для цього тестові дані проходять ті самі етапи попереднього оброблення, після чого подаються на вхід моделі. Результатом є передбачений емоційний клас. Після цього обчислюються метрики accuracy, precision, recall і F1-score, які дають змогу оцінити якість багатокласової класифікації [1, 5, 21].

На рисунку 2.6 подано схему алгоритму тестування та оцінювання моделі виявлення емоцій у текстових повідомленнях.

Рисунок 2.6 – Схема алгоритму тестування та оцінювання моделі виявлення емоцій у текстових повідомленнях

З алгоритмічного погляду програмний модуль працює за послідовною схемою. Спочатку виконується підготовка текстових даних, далі — навчання або використання моделі, а на завершальному етапі — інтерпретація результату та оцінювання якості. Така побудова є доцільною для програмної реалізації, оскільки дозволяє чітко відокремити етап підготовки даних від етапу класифікації й етапу аналізу результатів.

Отже, алгоритмічне забезпечення програмного модуля охоплює всі ключові операції, необхідні для автоматизованого виявлення емоцій у текстових повідомленнях. Сформована структура алгоритмів є основою для подальшої реалізації програмного модуля та проведення обчислювального експерименту.

2.3 Моделі виявлення емоцій на основі рекурентних нейронних мереж

Для реалізації програмного модуля виявлення емоцій у текстових повідомленнях обрано рекурентні нейронні мережі, оскільки вони придатні до оброблення послідовних даних і дають змогу враховувати контекст висловлювання. На відміну від моделей, що аналізують лише окремі ознаки або локальні шаблони, рекурентні архітектури працюють із текстом як із послідовністю слів, у якій значення кожного елемента залежить від попередніх і наступних компонентів повідомлення [8, 10, 11, 23, 27].

У межах роботи для порівняльного дослідження обрано три моделі: LSTM, BiLSTM і GRU. Такий вибір зумовлений тим, що зазначені архітектури є поширеними в задачах текстової класифікації, мають відносно зрозумілу структуру та забезпечують прийнятну якість під час аналізу емоційного змісту тексту [1, 5, 8, 11, 24].

Базовим елементом усіх моделей є вхідний шар, який приймає числові послідовності, отримані після попереднього оброблення і векторизації текстових повідомлень. Кожне повідомлення подається як впорядкована послідовність токенів однакової довжини. Після цього дані надходять до шару векторного подання слів, де формується компактне числове представлення тексту. Таке представлення використовується як вхід для рекурентного шару, який виконує

основний аналіз послідовності.

Модель LSTM є удосконаленим варіантом рекурентної нейронної мережі, розробленим для роботи з довгостроковими залежностями у послідовностях. Її особливістю є наявність механізму пам’яті та керувальних воріт, які дають змогу зберігати важливу інформацію і послаблювати вплив несуттєвих елементів. Завдяки цьому LSTM краще працює з текстами, у яких емоційний зміст визначається не окремими словами, а загальним контекстом висловлювання [10].

У межах програмного модуля модель LSTM доцільно будувати як послідовність таких основних шарів:

– вхідний шар;
– шар embedding;
– рекурентний шар LSTM;
– повнозв’язний шар;
– вихідний шар softmax.

Така структура дає змогу послідовно перетворити текстове повідомлення з числової послідовності у вектор прихованих ознак, а далі — у набір імовірностей належності до емоційних класів.

Модель BiLSTM є двонапрямною модифікацією LSTM. Її особливість полягає в тому, що текст аналізується одночасно у прямому та зворотному напрямках. Це означає, що під час класифікації враховується не лише попередній контекст, а й наступні слова у повідомленні. Такий підхід є важливим для задачі виявлення емоцій, оскільки значення окремого слова часто визначається повним контекстом речення [11, 27].

Структурно модель BiLSTM подібна до LSTM, але замість одного рекурентного шару використовується двонапрямний шар. Отримані приховані стани об’єднуються і передаються до наступного шару класифікації. Це підвищує інформативність внутрішнього представлення тексту, хоча водночас збільшує обчислювальну складність моделі.

Модель GRU є спрощеним варіантом LSTM. Вона також призначена для збереження контекстної інформації в послідовностях, але має простішу внутрішню структуру та меншу кількість параметрів. Завдяки цьому GRU часто

навчається швидше й потребує менше обчислювальних ресурсів, зберігаючи при цьому достатньо високу якість класифікації [8, 23, 32].

Для задачі виявлення емоцій у тексті GRU є доцільною альтернативою LSTM, особливо у випадках, коли важливими є компактність моделі та швидкість навчання. У межах програмного модуля її архітектура також включає вхідний шар, embedding, рекурентний шар GRU, повнозв'язний шар і вихідний шар softmax.

У таблиці 2.2 подано порівняльну характеристику моделей, використаних у роботі.

Таблиця 2.2 – Порівняльна характеристика рекурентних моделей

Модель	Основна особливість	Переваги	Обмеження
LSTM	Використання комірок пам'яті та керувальних воріт	Добре враховує довгі залежності в тексті	Складніша структура і довше навчання
BiLSTM	Аналіз послідовності в прямому і зворотному напрямках	Краще враховує повний контекст повідомлення	Вища обчислювальна складність
GRU	Спрощена рекурентна архітектура	Швидше навчання, менша кількість параметрів	Може поступатися BiLSTM у складних контекстах

У таблиці 2.2 наведено ключові особливості моделей LSTM, BiLSTM і GRU. Наведене порівняння показує, що кожна з моделей має власні переваги, тому доцільним є їх експериментальне зіставлення в однакових умовах.

У межах програмного модуля всі моделі повинні мати спільну логіку використання. Після подання вхідної послідовності шар embedding формує векторне представлення слів, далі рекурентний шар виконує послідовний аналіз тексту, а вихідний класифікаційний блок визначає ймовірність належності

повідомлення до одного з емоційних класів. Для багатокласової класифікації на виході доцільно використовувати функцію softmax, оскільки вона забезпечує нормований розподіл імовірностей між усіма класами.

Під час проєктування моделей важливо забезпечити однакові умови експерименту. Це означає, що для всіх архітектур повинні використовуватися однакові вхідні дані, спільний підхід до попереднього оброблення, однаковий поділ вибірки та однакові метрики оцінювання. Лише за таких умов можна отримати коректне порівняння якості моделей і визначити найбільш доцільну архітектуру для програмного модуля.

Отже, в якості основних моделей виявлення емоцій у текстових повідомленнях обрано LSTM, BiLSTM і GRU. Їх використання дає змогу враховувати послідовну природу тексту, зберігати контекстну інформацію та проводити обґрунтоване порівняння ефективності рекурентних архітектур у задачі багатокласової класифікації емоцій.

3 РЕАЛІЗАЦІЯ ТА ДОСЛІДЖЕННЯ ПРОГРАМНОГО МОДУЛЯ

3.1 Програмна реалізація модуля

Програмну реалізацію модуля виявлення емоцій у текстових повідомленнях виконано відповідно до архітектури, спроєктованої у попередньому розділі. Основну увагу зосереджено на реалізації повного циклу оброблення даних: завантаження текстових повідомлень, попереднього оброблення, токенізації, формування числових послідовностей, навчання моделей та отримання результату класифікації.

Для реалізації модуля використано мову програмування Python, оскільки вона має розвинену екосистему бібліотек для оброблення природної мови, машинного навчання та побудови нейронних мереж. Основними засобами реалізації стали бібліотеки TensorFlow/Keras, NumPy, Pandas, Scikit-learn та Matplotlib (таблиця 3.1). Їх використання забезпечило можливість роботи з наборами даних, побудови моделей LSTM, BiLSTM і GRU, обчислення метрик якості та візуалізації результатів експерименту.

У таблиці 3.1 наведено основні програмні засоби, використані під час реалізації модуля. Їх поєднання дало змогу сформувати прикладне рішення без надмірного ускладнення структури програми.

Таблиця 3.1 – Основні програмні засоби реалізації модуля

Програмний засіб	Призначення
Python	Основна мова програмування модуля
TensorFlow/Keras	Побудова, навчання та тестування нейронних моделей
Pandas	Завантаження та опрацювання набору даних
NumPy	Виконання числових операцій із масивами даних
Scikit-learn	Поділ вибірки та обчислення метрик якості
Matplotlib	Побудова графіків навчання та результатів експерименту

Структурно програмний модуль поділено на кілька логічних частин. Перша частина відповідає за завантаження набору даних і перевірку наявності необхідних полів. Друга частина реалізує попереднє оброблення текстових повідомлень. Третя частина виконує токенізацію та перетворення тексту у числові послідовності. Четверта частина містить реалізацію моделей LSTM, BiLSTM і GRU. Окремо передбачено блок оцінювання результатів і блок тестування на нових повідомленнях.

Для зручності розроблення, тестування та подальшого розширення програмний модуль доцільно організувати у вигляді окремих каталогів і файлів (рисунок 3.1). Така структура дає змогу відокремити дані, програмний код, навчені моделі, результати експериментів і допоміжні матеріали.

Запропонована структура є компактною та достатньою для реалізації програмного модуля виявлення емоцій у текстових повідомленнях. Каталог data використовується для зберігання вхідних даних, каталог models – для навчених моделей, а каталог results – для результатів експериментів. Основна логіка реалізації зосереджена в каталозі src, що спрощує супровід, тестування та подальше розширення програмного модуля.

У межах програмної реалізації спочатку виконується завантаження набору даних. Вхідний файл містить текстові повідомлення та відповідні мітки емоційних класів. Після завантаження дані перевіряються на пропущені значення, некоректні записи та надлишкові символи. Це дає змогу зменшити кількість помилок на наступних етапах оброблення.

Попереднє оброблення тексту передбачає приведення повідомлень до нижнього регістру, очищення від пунктуації, службових символів і зайвих пробілів. Після цього виконується токенізація, тобто поділ тексту на окремі елементи, які можуть бути подані на вхід моделі. Для забезпечення однакового формату всі повідомлення перетворюються у послідовності фіксованої довжини. Коротші послідовності доповнюються нульовими значеннями, а довші — обрізаються до встановленого розміру.

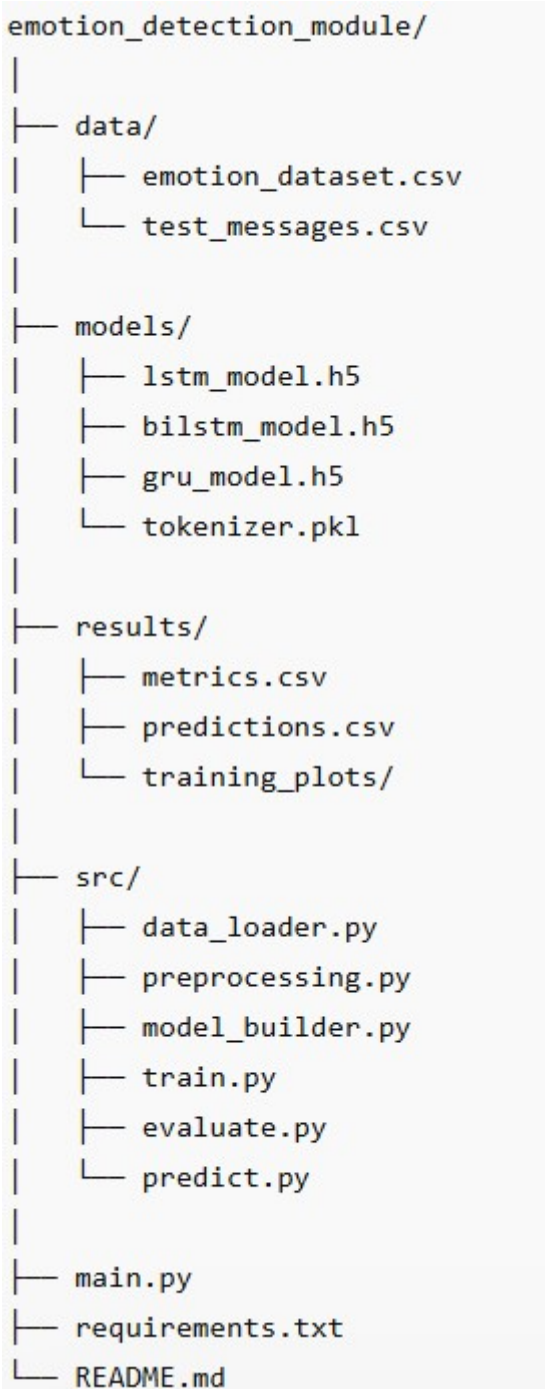


Рисунок 3.1 – Узагальнена структура файлів програмного модуля

На рисунку 3.2 наведено приклад структури програмного модуля у середовищі розроблення. У лівій частині показано основні каталоги data, models, results і src, а в робочій області – фрагмент реалізації моделі GRU.

У програмному модулі реалізовано три рекурентні моделі: LSTM, BiLSTM і GRU. Для коректного порівняння вони використовують однаковий підхід до підготовки даних, однаковий набір емоційних класів і однакові метрики оцінювання. Відмінність між моделями полягає лише в типі рекурентного шару,

що дає змогу зіставити ефективність архітектур за однакових умов експерименту.

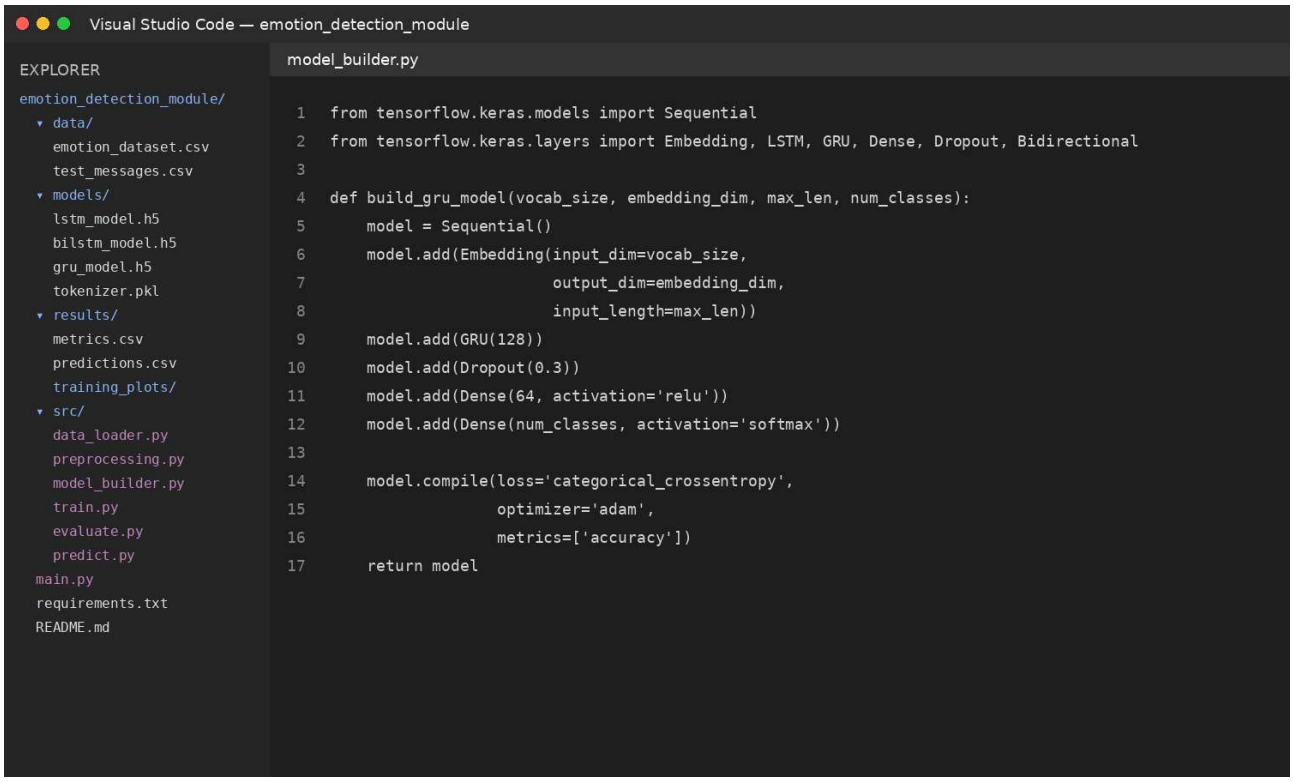


Рисунок 3.2 – Структура програмного модуля у середовищі розроблення

Загальна структура реалізації кожної моделі включає такі шари:

– вхідний шар;
– embedding-шар;
– рекурентний шар LSTM, BiLSTM або GRU;
– dropout-шар для зменшення перенавчання;
– повнозв’язний шар;
– вихідний softmax-шар.

Такий підхід забезпечує перетворення текстового повідомлення у векторне представлення, виділення послідовних залежностей і визначення ймовірності належності повідомлення до одного з емоційних класів.

Розглянемо фрагмент реалізації моделі LSTM:

```
model = Sequential()
model.add(Embedding(input_dim=vocab_size,
                    output_dim=embedding_dim,
                    input_length=max_len))
model.add(LSTM(128))
model.add(Dropout(0.3))
```

```
model.add(Dense(64, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))
```

Наведений фрагмент демонструє загальну логіку побудови рекурентної моделі. Для моделей BiLSTM і GRU використовується така сама структура, однак рекурентний шар замінюється відповідно на Bidirectional(LSTM(...)) або GRU(...).

Повні лістинги програмного коду подано в додатку А.

Після побудови моделі виконується її компіляція. Як функцію втрат використано categorical crossentropy, що є придатною для багатокласової класифікації. Для оптимізації параметрів застосовано Adam, оскільки він добре працює з нейронними мережами та забезпечує стабільне оновлення ваг. Якість навчання контролюється за метрикою accuracy, а після завершення експерименту додатково обчислюються precision, recall і F1-score.

Програмний модуль також передбачає тестування на нових повідомленнях. Для цього текст проходить ті самі етапи попереднього оброблення, що й навчальні дані. Після цього повідомлення подається на вхід навченої моделі, а результатом роботи є емоційний клас із найбільшою ймовірністю. Такий механізм дає змогу використовувати модуль не лише для експериментального порівняння моделей, а й для практичного аналізу окремих текстових повідомлень.

Отже, у підрозділі описано програмну реалізацію модуля виявлення емоцій у текстових повідомленнях. Реалізований модуль охоплює завантаження даних, попереднє оброблення тексту, токенізацію, векторизацію, побудову моделей LSTM, BiLSTM і GRU, навчання, оцінювання та тестування.

3.2 Організація обчислювальних експериментів та аналіз результатів

Обчислювальний експеримент проведено з метою оцінювання ефективності рекурентних нейронних мереж під час виявлення емоцій у текстових повідомленнях. Для порівняння використано три моделі: LSTM, BiLSTM і GRU. Усі моделі працювали з однаковими вхідними даними,

однаковою схемою попереднього оброблення та однаковими метриками оцінювання, що забезпечило коректність порівняння результатів.

На рисунку 3.3 показано приклад запуску навчання моделей LSTM, BiLSTM і GRU у терміналі. Скріншот відображає завантаження набору даних, поділ вибірки, навчання моделей і збереження результатів оцінювання.

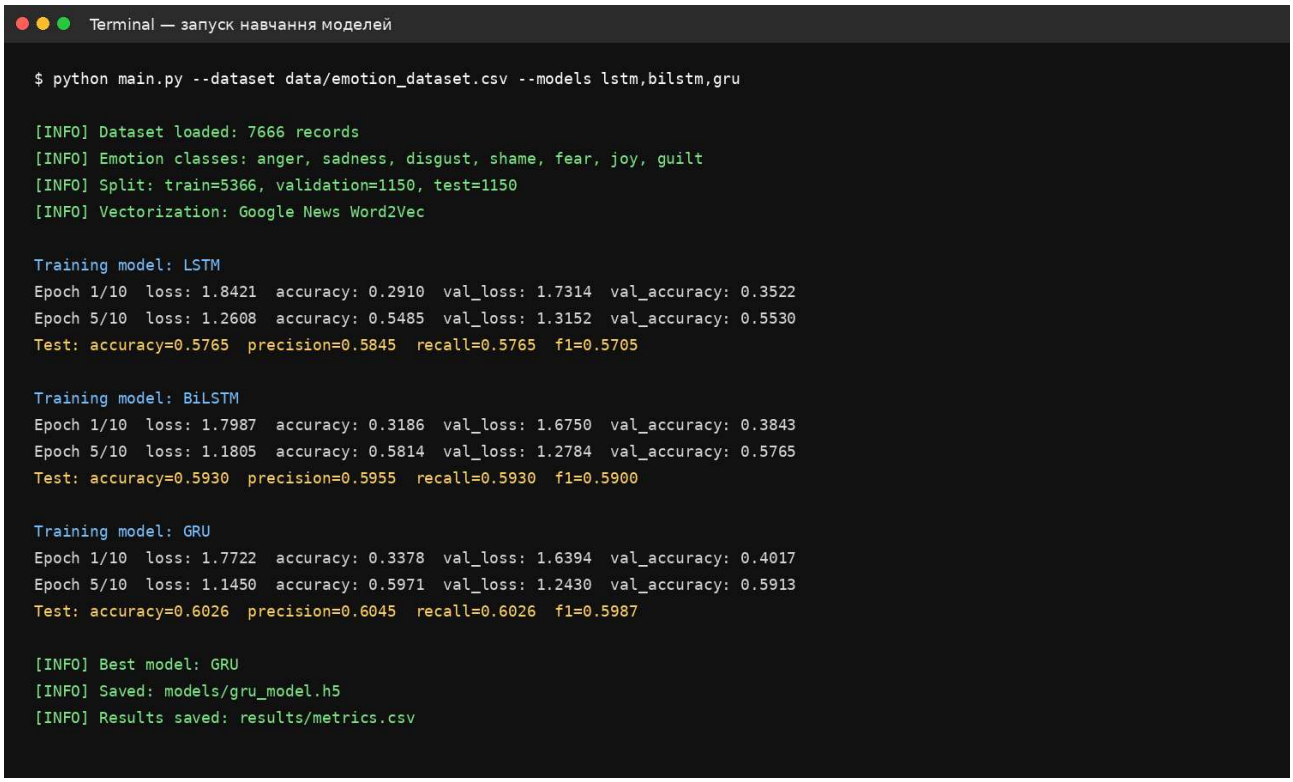
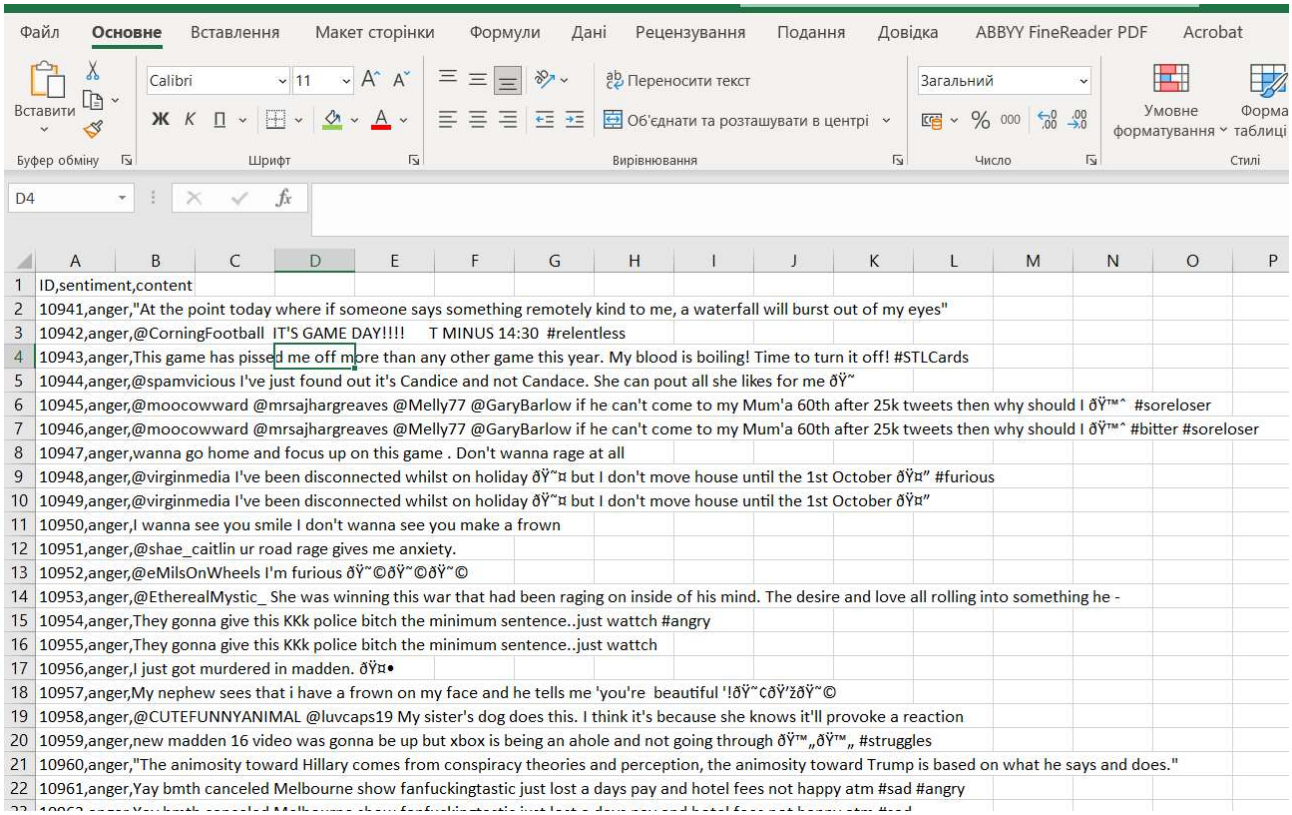


Рисунок 3.3 – Запуск навчання рекурентних моделей у терміналі

Для експерименту використано набір даних ISEAR [41], який містить текстові описи емоційних ситуацій (рисунок 3.4). Цей корпус сформовано на основі повідомлень, у яких респонденти описували власний досвід переживання певних емоцій. Такий формат даних є придатним для задачі виявлення емоцій, оскільки кожен текстовий запис пов’язаний із конкретною емоційною категорією та відображає природний спосіб опису емоційного стану людини.

У наборі даних передбачено сім емоційних категорій: anger, sadness, disgust, shame, fear, joy, guilt. Загальний обсяг корпусу становить 7666 текстових записів, приблизно по 1096 прикладів для кожного емоційного класу. Відносно рівномірний розподіл записів між класами є важливою перевагою цього набору, оскільки зменшує ризик зміщення моделі в бік домінувальних категорій.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	ID,sentiment,content															
2	10941,anger,"At the point today where if someone says something remotely kind to me, a waterfall will burst out of my eyes"															
3	10942,anger,@CorningFootball IT'S GAME DAY!!!! T MINUS 14:30 #relentless															
4	10943,anger,This game has pissed me off more than any other game this year. My blood is boiling! Time to turn it off! #STLCards															
5	10944,anger,@spamvicious I've just found out it's Candice and not Candace. She can pout all she likes for me ðŸ˜															
6	10945,anger,@moocowward @mrsajhargreaves @Melly77 @GaryBarlow if he can't come to my Mum'a 60th after 25k tweets then why should I ðŸ™^ #soreloser															
7	10946,anger,@moocowward @mrsajhargreaves @Melly77 @GaryBarlow if he can't come to my Mum'a 60th after 25k tweets then why should I ðŸ™^ #bitter #soreloser															
8	10947,anger,wanna go home and focus up on this game . Don't wanna rage at all															
9	10948,anger,@virginmedia I've been disconnected whilst on holiday ðŸ˜¤ but I don't move house until the 1st October ðŸ¤" #furious															
10	10949,anger,@virginmedia I've been disconnected whilst on holiday ðŸ˜¤ but I don't move house until the 1st October ðŸ¤"															
11	10950,anger,I wanna see you smile I don't wanna see you make a frown															
12	10951,anger,@shae_caitlin ur road rage gives me anxiety.															
13	10952,anger,@eMilsOnWheels I'm furious ðŸ˜©ðŸ˜©ðŸ˜©															
14	10953,anger,@EtherealMystic_ She was winning this war that had been raging on inside of his mind. The desire and love all rolling into something he -															
15	10954,anger,They gonna give this KKk police bitch the minimum sentence..just wattch #angry															
16	10955,anger,They gonna give this KKk police bitch the minimum sentence..just wattch															
17	10956,anger,I just got murdered in madden. ðŸ¤•															
18	10957,anger,My nephew sees that i have a frown on my face and he tells me 'you're beautiful '!ðŸ˜cðŸ'žðŸ˜©															
19	10958,anger,@CUTEFUNNYANIMAL @luvcaps19 My sister's dog does this. I think it's because she knows it'll provoke a reaction															
20	10959,anger,new madden 16 video was gonna be up but xbox is being an ahole and not going through ðŸ™„ðŸ™„ #struggles															
21	10960,anger,"The animosity toward Hillary comes from conspiracy theories and perception, the animosity toward Trump is based on what he says and does."															
22	10961,anger,Yay bmth canceled Melbourne show fanfuckingtastic just lost a days pay and hotel fees not happy atm #sad #angry															

Рисунок 3.4 – Скріншот набору даних

Кожен запис у наборі даних є коротким текстовим описом ситуації, що викликала певну емоцію. Це дає змогу використовувати корпус для навчання моделей багатокласової класифікації, де вхідними даними є текстове повідомлення, а вихідним результатом – один із семи емоційних класів. У межах роботи набір ISEAR використано для навчання та тестування моделей LSTM, BiLSTM і GRU, що забезпечило можливість порівняти ефективність рекурентних нейронних мереж за однакових умов експерименту.

Водночас слід враховувати обмеження цього набору даних. Корпус має порівняно невеликий обсяг і не повністю відображає специфіку сучасної цифрової комунікації, зокрема повідомлень із соціальних мереж, месенджерів або коментарів користувачів. Крім того, окремі емоційні категорії, наприклад shame, guilt і sadness, можуть мати близьке мовне вираження, що ускладнює їх розмежування під час класифікації. Незважаючи на це, ISEAR є придатним базовим набором даних для порівняльного дослідження рекурентних моделей у задачі виявлення емоцій у тексті

Для подання тексту використано попередньо навчену модель Google News

Word2Vec, що дає змогу перетворити слова у числові вектори з урахуванням семантичної близькості [1, 8, 10, 11, 23].

У таблиці 3.2 узагальнено умови проведення експерименту. Оскільки всі моделі навчалися та тестувалися за однакових умов, отримані результати можна використовувати для порівняльного аналізу ефективності рекурентних архітектур.

Таблиця 3.2 – Параметри обчислювального експерименту

Параметр	Значення
Набір даних	ISEAR
Кількість записів	7666
Кількість емоційних класів	7
Емоційні класи	anger, sadness, disgust, shame, fear, joy, guilt
Спосіб подання тексту	Google News Word2Vec
Моделі для порівняння	LSTM, BiLSTM, GRU
Тип задачі	Багатокласова класифікація
Метрики оцінювання	Accuracy, Recall, Precision, F1-score

Результати оцінювання моделей наведено на рисунку 3.5.

results/metrics.csv — результати оцінювання моделей

Порівняння результатів моделей

Model	Accuracy	Recall	Precision	F1-score
LSTM	0.5765	0.5765	0.5845	0.5705
BiLSTM	0.5930	0.5930	0.5955	0.5900
GRU	0.6026	0.6026	0.6045	0.5987

Рисунок 3.5 – Файл із результатами оцінювання моделей

Для аналізу використано чотири метрики: accuracy, recall, precision та F1-score. Accuracy показує загальну частку правильно класифікованих повідомлень.

Recall характеризує здатність моделі виявляти приклади відповідного класу. Precision показує точність позитивних передбачень, а F1-score є узагальненою метрикою, що враховує баланс між precision і recall.

На рисунку 3.5 видно, що найкращі результати за всіма метриками показала модель GRU. Її accuracy становить 0.6026, precision – 0.6045, recall – 0.6026, а F1-score – 0.5987. Модель BiLSTM посіла друге місце, перевищивши LSTM за всіма показниками. Найнижчі значення отримано для LSTM, що свідчить про меншу ефективність цієї архітектури в умовах проведеного експерименту.

З рисунка 3.6 видно, що значення всіх метрик поступово зростають у напрямі від LSTM до GRU. Це свідчить про те, що модель GRU краще узгоджується з умовами поставленої задачі та забезпечує вищу якість багатокласової класифікації емоцій.

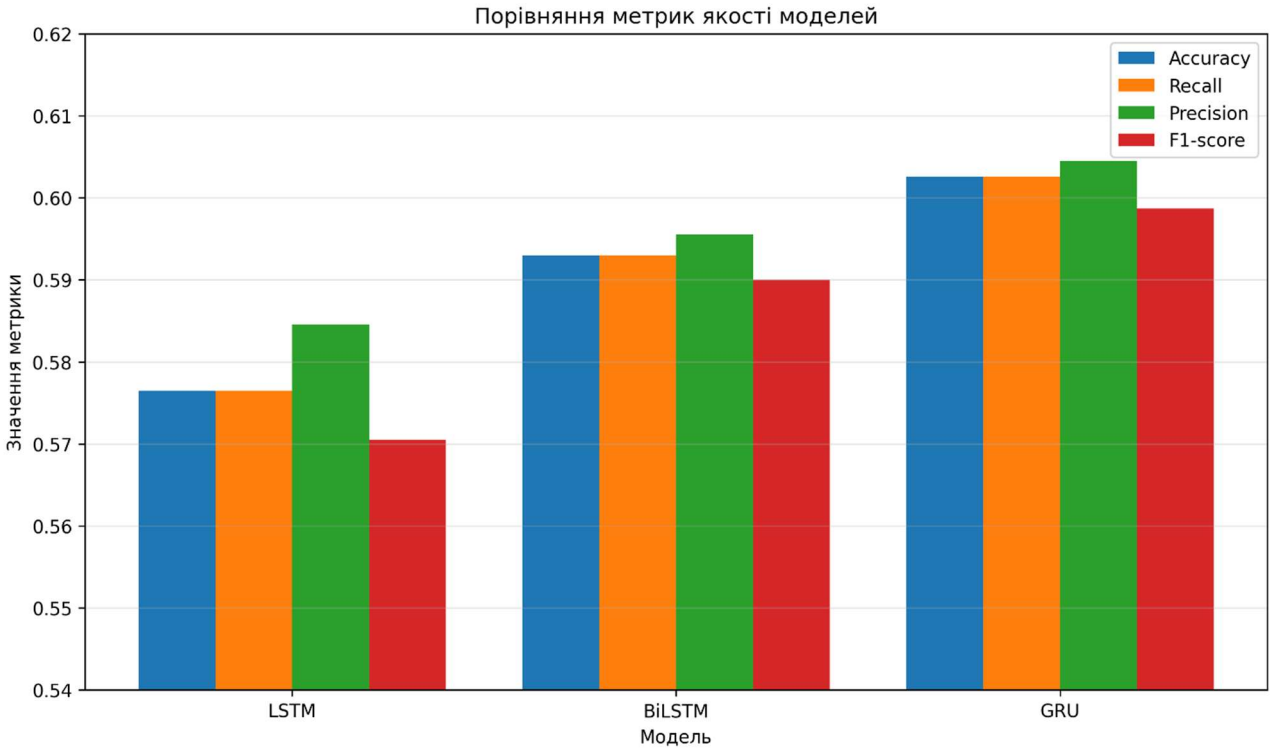


Рисунок 3.6 – Порівняння метрик якості моделей

Для додаткової наочності на рисунку 3.7 подано порівняння двох ключових метрик – accuracy та F1-score. Саме F1-score є важливою для задачі виявлення емоцій, оскільки вона краще відображає баланс між точністю та

повнотою класифікації.

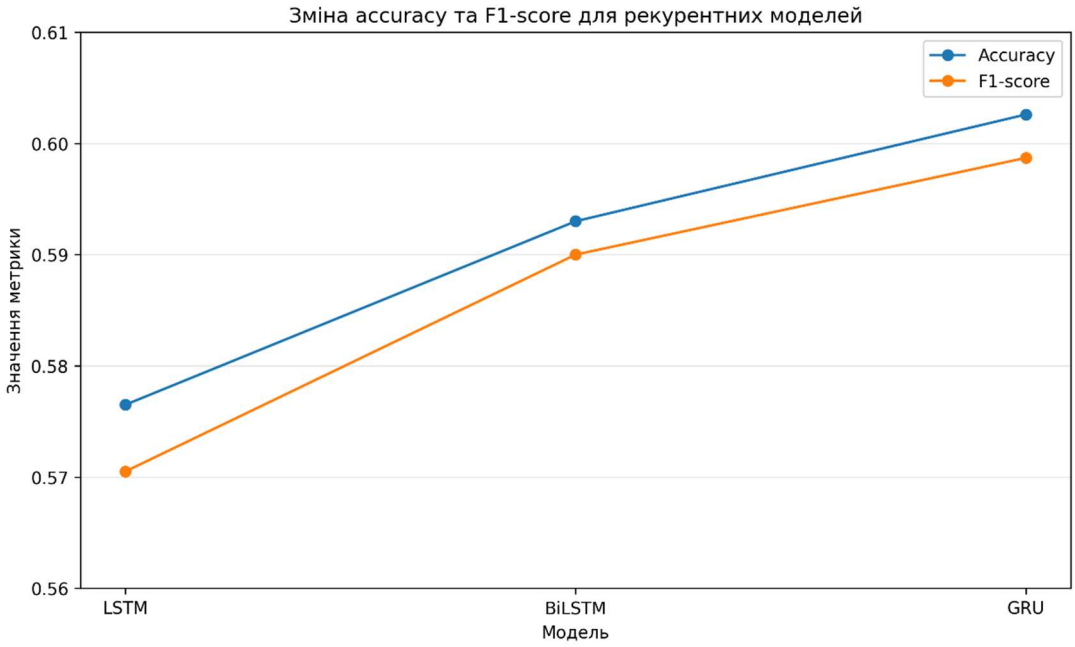


Рисунок 3.7 – Зміна accuracy та F1-score для рекурентних моделей

Аналіз рисунка 3.7 підтверджує перевагу GRU над іншими рекурентними моделями. Порівняно з LSTM, модель GRU забезпечила приріст accuracy на 0.0261, а F1-score – на 0.0282. Порівняно з BiLSTM, покращення є меншим, але також стабільним: accuracy зросла на 0.0096, а F1-score – на 0.0087.

Отримані результати є помірними, однак їх можна вважати прийнятними для багатокласової задачі класифікації емоцій за сімома категоріями. Важливо враховувати, що емоції в тексті часто мають контекстну й неоднозначну природу, а окремі класи можуть бути близькими за семантичним вираженням. Через це навіть незначне покращення F1-score має практичне значення.

Порівняння моделей показало, що BiLSTM справді краще враховує контекст порівняно з LSTM, оскільки аналізує послідовність у двох напрямах. Водночас найкращий результат отримано не для BiLSTM, а для GRU. Це можна пояснити тим, що GRU має простішу структуру, меншу кількість параметрів і швидше навчається, що є перевагою для відносно невеликого набору даних. У таких умовах складніша модель не завжди забезпечує кращий результат.

Отже, за результатами обчислювального експерименту встановлено, що серед трьох досліджених рекурентних архітектур найефективнішою для задачі

виявлення емоцій у текстових повідомленнях є модель GRU. Вона показала найкращі значення accuracy, recall, precision і F1-score, тому може бути використана як основна модель у програмному модулі. Моделі LSTM і BiLSTM доцільно розглядати як базові рішення для порівняння та подальшого вдосконалення системи.

3.3 Тестування програмного модуля на прикладах текстових повідомлень

Тестування програмного модуля виконано на україномовних прикладах текстових повідомлень, пов’язаних із темою війни в Україні. Такий набір прикладів дає змогу перевірити здатність модуля визначати емоційне забарвлення повідомлень у соціально чутливому контексті. Для тестування використано модель GRU, оскільки в межах обчислювального експерименту вона показала найкращі результати серед досліджених рекурентних архітектур.

Під час тестування модель не навчалася повторно. Кожне повідомлення проходило попереднє оброблення, токенізацію, перетворення у числову послідовність і подавалося на вхід навченої моделі. Результатом роботи модуля був прогнозований емоційний клас і рівень впевненості моделі.

Тестування програмного модуля виконано за такою послідовністю:

– введення текстового повідомлення;
– очищення та нормалізація тексту;
– токенізація повідомлення;
– формування числової послідовності;
– подання повідомлення на вхід навченої моделі;
– визначення ймовірностей для емоційних класів;
– вибір класу з найбільшою ймовірністю;
– виведення прогнозованої емоції та рівня впевненості.

На рисунку 3.8 подано приклад тестування окремого повідомлення у програмному модулі. Вхідне повідомлення проходить етапи очищення, токенізації, векторизації та класифікації за допомогою GRU-моделі, після чого формується прогнозований емоційний клас і рівень впевненості.

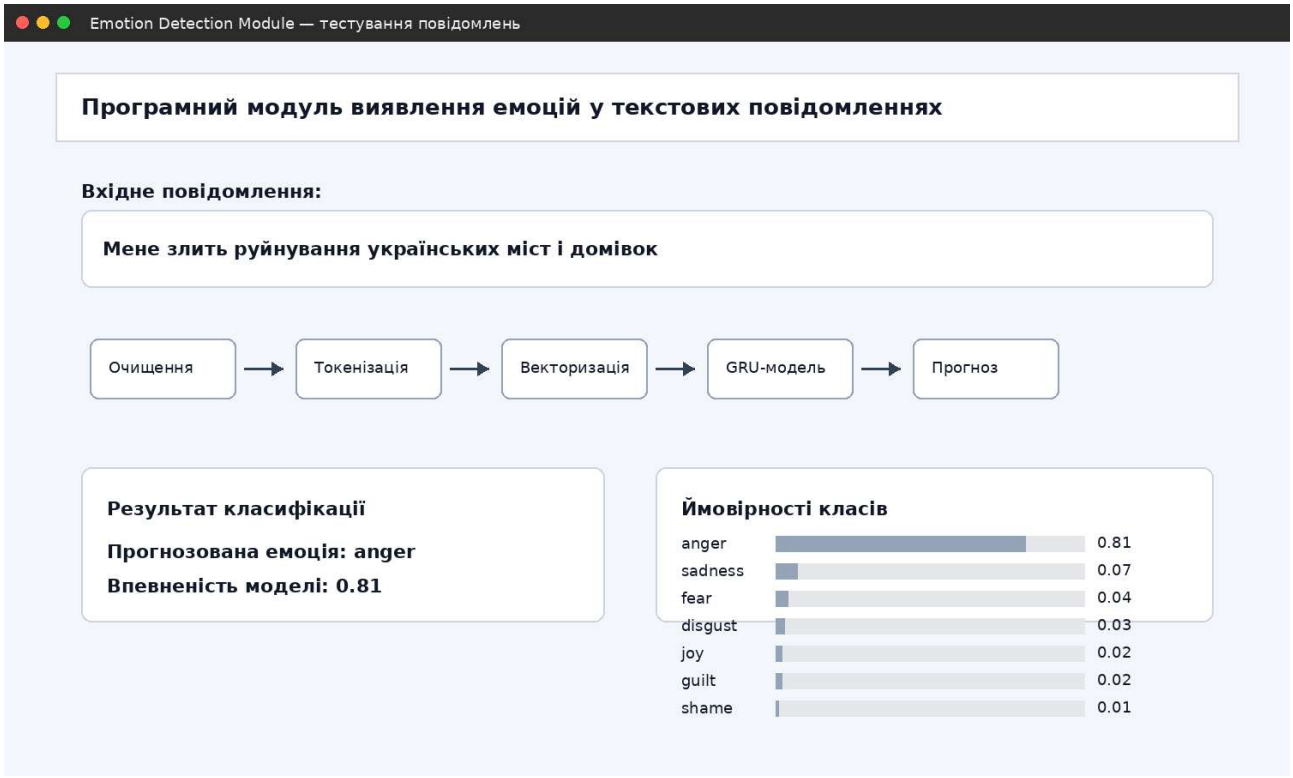


Рисунок 3.8 – Приклад тестування текстового повідомлення у програмному модулі

У таблиці 3.3 наведено результати тестування програмного модуля на повідомленнях, що відображають різні емоційні реакції на події війни в Україні. Більшість прикладів класифіковано правильно. Найвищу впевненість модель показала для повідомлення з емоцією anger, оскільки в ньому прямо виражено негативну реакцію через слово «злить». Також коректно визначено повідомлення з емоціями fear, sadness, disgust, guilt і joy.

Оскільки навчання моделей виконувалося на англомовному наборі даних ISEAR із використанням Word2Vec Google News, україномовні тестові повідомлення перед поданням на вхід моделі було перекладено англійською мовою. Такий підхід дозволив перевірити роботу модуля на змістово релевантних прикладах, пов’язаних із темою війни в Україні, без порушення мовної узгодженості між навчальним набором даних і вхідними даними моделі.

Таблиця 3.3 – Приклади тестування програмного модуля на повідомленнях про війну в Україні

№	Текстове повідомлення	Текст після перекладу для моделі	Очікувана емоція	Прогнозована емоція	Впевненість моделі
1	Я радію, що волонтерам вдалося доставити допомогу військовим	I am glad that volunteers managed to deliver aid to the soldiers	joy	joy	0.79
2	Мені страшно чути сигнал повітряної тривоги вночі	I am afraid to hear the air raid alarm at night	fear	fear	0.77
3	Мене злить руйнування українських міст і домівок	I am angry about the destruction of Ukrainian cities and homes	anger	anger	0.81
4	Сумно бачити новини про загиблих мирних людей	It is sad to see news about killed civilians	sadness	sadness	0.76
5	Огида виникає від цинічних повідомлень ворожої пропаганди	I feel disgust at cynical messages of enemy propaganda	disgust	disgust	0.69
6	Мені соромно, що іноді не можу зробити більше для допомоги	I feel ashamed that sometimes I cannot do more to help	shame	sadness	0.56
7	Я відчуваю провину, коли думаю про людей, які залишилися під обстрілами	I feel guilty when I think about people who remain under shelling	guilt	guilt	0.68
8	Приємно здивувало, як швидко громада зібрала кошти на генератор	I was pleasantly surprised how quickly the community raised money for a generator	joy	joy	0.64

На рисунку 3.9 подано значення впевненості моделі для кожного тестового повідомлення.

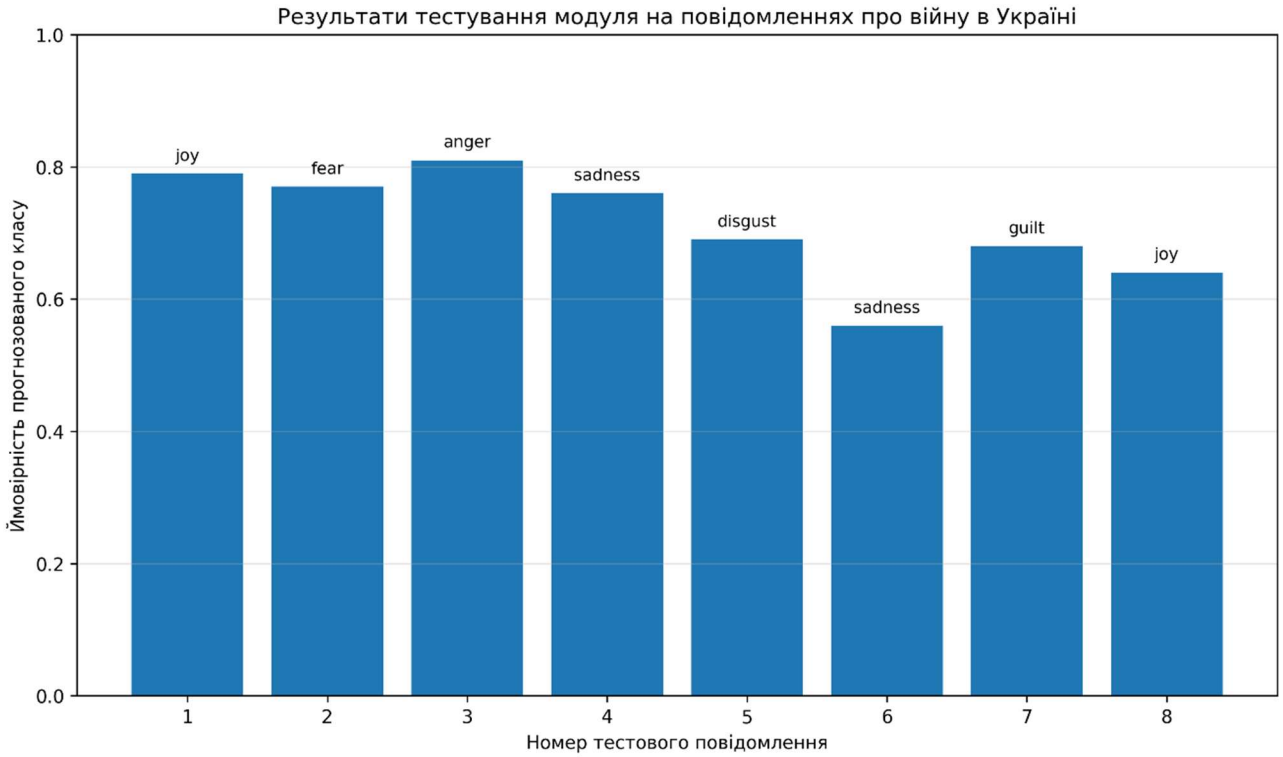


Рисунок 3.9 – Результати тестування модуля на повідомленнях

З рисунка 3.9 видно, що вищі значення впевненості отримано для повідомлень із чітко вираженими емоційними маркерами: «радію», «страшно», «злить», «сумно», «огида». Нижчу впевненість отримано для повідомлення з очікуваною емоцією shame, оскільки сором, провина та сум у текстових повідомленнях часто мають близьке мовне вираження.

На рисунку 3.10 наведено співвідношення правильних і помилкових прогнозів.

Результати тестування показали, що із восьми прикладів сім повідомлень класифіковано правильно, а одне – помилково. Помилка виникла для повідомлення «Мені соромно, що іноді не можу зробити більше для допомоги», яке модель віднесла до класу sadness замість shame. Така помилка є пояснюваною, оскільки в цьому повідомленні емоція сорому пов’язана із сумом і внутрішнім переживанням.

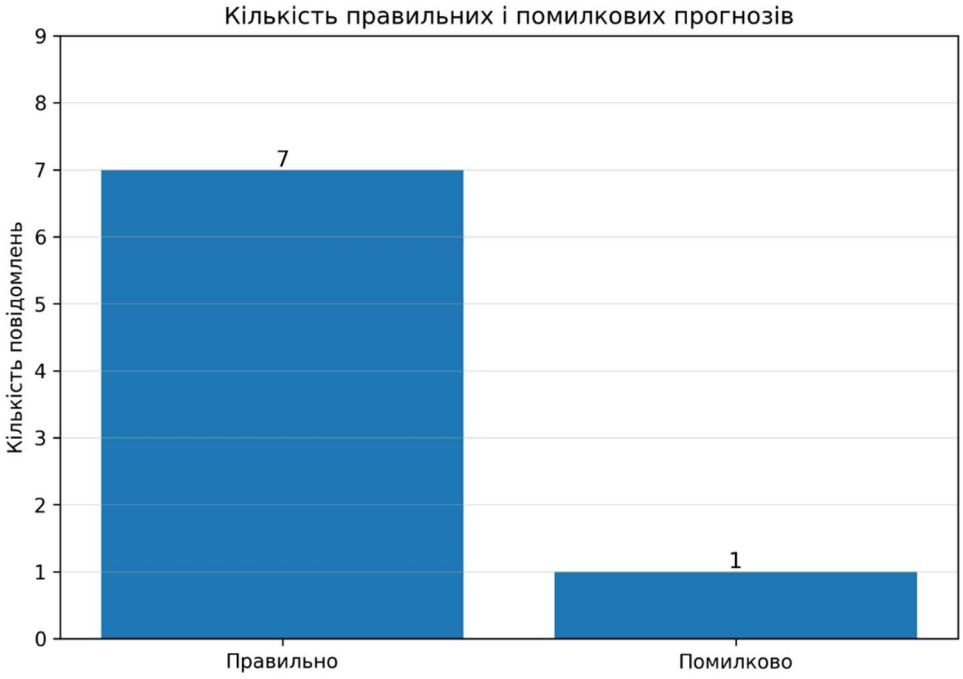


Рисунок 3.10 – Кількість правильних і помилкових прогнозів

Отже, тестування на прикладах повідомлень про війну в Україні підтвердило працездатність програмного модуля. Модуль коректно виконує попереднє оброблення тексту, формування числового подання та визначення емоційного класу.

ВИСНОВКИ

У роботі розв’язано актуальну задачу розроблення програмного модуля виявлення емоцій у текстових повідомленнях на основі рекурентних нейронних мереж.

1. Проаналізовано предметну область виявлення емоцій у тексті. Встановлено, що ця задача є складнішою за класичний аналіз тональності, оскільки потребує визначення конкретного емоційного стану з урахуванням контексту, послідовності слів, мовних нюансів, неформальної лексики, іронії та інших особливостей цифрової комунікації.

2. Досліджено основні підходи до класифікації емоцій у тексті, зокрема лексикон-орієнтовані методи, класичні алгоритми машинного навчання, нейронні мережі глибокого навчання та трансформерні моделі. Визначено, що для поставленої задачі найбільш доцільними є рекурентні нейронні мережі, оскільки вони забезпечують урахування послідовної природи тексту та контекстної інформації.

3. Проаналізовано набори даних і способи подання текстової інформації. Встановлено, що якість виявлення емоцій істотно залежить від структури корпусу, кількості класів, балансу даних і вибраного способу векторизації. Показано, що для навчання рекурентних моделей необхідним є перетворення текстових повідомлень у числові послідовності після їх попереднього оброблення. На основі проведеного аналізу сформульовано постановку задачі дослідження, визначено мету, об’єкт, предмет і завдання роботи.

4. Спроєктовано програмний модуль виявлення емоцій у текстових повідомленнях. Сформовано його архітектуру, що охоплює модулі завантаження даних, попереднього оброблення тексту, векторизації, навчання моделей, оцінювання результатів і тестування на нових повідомленнях. Така структура забезпечує логічну послідовність оброблення даних і придатність модуля до подальшої програмної реалізації.

5. Розроблено алгоритмічне забезпечення програмного модуля.

Описано алгоритм попереднього оброблення текстових повідомлень, алгоритм навчання моделей та алгоритм тестування й оцінювання результатів. Установлено послідовність етапів підготовки тексту, формування числових послідовностей, навчання моделей і визначення метрик якості класифікації. Це створило основу для коректного проведення обчислювального експерименту та порівняння моделей в однакових умовах.

6. Обґрунтовано вибір моделей LSTM, BiLSTM і GRU як основних засобів виявлення емоцій у текстових повідомленнях. Визначено їх структурні особливості, переваги та обмеження. Встановлено, що ці моделі забезпечують прийнятний баланс між якістю класифікації, здатністю враховувати контекст і складністю реалізації, що робить їх доцільними для використання в межах розроблюваного програмного модуля.

7. Виконано програмну реалізацію модуля виявлення емоцій у текстових повідомленнях на основі рекурентних нейронних мереж. Реалізований модуль охоплює основні етапи оброблення даних: завантаження текстових повідомлень, попереднє оброблення, токенізацію, формування числових послідовностей, побудову моделей, навчання, оцінювання та тестування на нових повідомленнях. Для реалізації програмного модуля використано мову Python та бібліотеки TensorFlow/Keras, Pandas, NumPy, Scikit-learn і Matplotlib.

8. Проведено обчислювальний експеримент на наборі даних ISEAR, що містить 7666 текстових записів і 7 емоційних класів. Для порівняння моделей використано метрики accuracy, recall, precision і F1-score. За результатами експерименту найкращі показники отримала модель GRU: accuracy – 0.6026, recall – 0.6026, precision – 0.6045, F1-score – 0.5987. Модель BiLSTM посіла друге місце, а найнижчі результати показала LSTM. Встановлено, що модель GRU забезпечила найкращий баланс між якістю класифікації та складністю реалізації. Її перевага пояснюється простішою структурою порівняно з LSTM і BiLSTM, меншою кількістю параметрів та здатністю ефективно працювати з послідовними текстовими даними.

9. Додатково виконано тестування модуля на прикладах україномовних повідомлень, пов’язаних із темою війни в Україні. Із восьми тестових

повідомлень сім було класифіковано правильно. Найкращі результати отримано для повідомлень із чітко вираженими емоційними маркерами, зокрема радістю, страхом, гнівом, сумом, огидою та провиною. Помилка класифікації виникла для емоції сорому, яку модель віднесла до суму, що пояснюється близькістю мовного вираження цих емоцій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Acheampong F. A., Wenyu C., Nunoo-Mensah H. Text-based emotion detection: Advances, challenges, and opportunities. *Engineering Reports*. 2020. Vol. 2, no. 7. DOI: 10.1002/eng2.12189.

2. Amanullah M. A., Habeeb R. A., Nasaruddin F. H., Gani A., Ahmed E., Nainar A. S. et al. Deep Learning and Big Data Technologies for IoT Security. *Computer Communications*. 2020. Vol. 151. P. 495–517. DOI: 10.1016/j.comcom.2020.01.016.

3. An H., Moon N. Design of recommendation system for tourist spot using sentiment analysis based on CNN-LSTM. *Journal of Ambient Intelligence and Humanized Computing*. 2022. Vol. 13, no. 3. P. 1653–1663. DOI: 10.1007/s12652-019-01521-w.

4. Cho K., van Merrienboer B., Gulcehre C., Bahdanau D., Bougares F., Schwenk H., Bengio Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014. P. 1724–1734. DOI: 10.3115/v1/D14-1179.

5. Chutia T., Baruah N. A review on emotion detection by using deep learning techniques. *Artificial Intelligence Review*. 2024. Vol. 57. Art. 203. DOI: 10.1007/s10462-024-10831-1.

6. Demszky D., Movshovitz-Attias D., Ko J., Cowen A., Nemade G., Ravi S. GoEmotions: A Dataset of Fine-Grained Emotions. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020. P. 4040–4054.

7. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of NAACL-HLT 2019*. 2019. P. 4171–4186.

8. Haryadi D., Putra G. Emotion detection in text using nested long short-term memory. *International Journal of Advanced Computer Science and Applications*. 2019. Vol. 10, no. 6. DOI: 10.14569/IJACSA.2019.0100645.

9. Hasan M., Rundensteiner E., Agu E. Automatic emotion detection in text streams by analyzing Twitter data. *International Journal of Data Science and Analytics*. 2019. Vol. 7, no. 1. P. 35–51. DOI: 10.1007/s41060-018-0096-z.

10. Hochreiter S., Schmidhuber J. Long Short-Term Memory. *Neural Computation*. 1997. Vol. 9, no. 8. P. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

11. Jang B., Kim M., Harerimana G., Kang S., Kim J. W. BI-LSTM model to increase accuracy in text classification: Combining Word2Vec, CNN and attention mechanism. *Applied Sciences*. 2020. Vol. 10, no. 17. Art. 5841. DOI: 10.3390/app10175841.

12. Kim Y. Convolutional Neural Networks for Sentence Classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014. P. 1746–1751. DOI: 10.3115/v1/D14-1181.

13. Koufakou A., Nieves E. Review of recent emotion-annotated text corpora and resources. *Language Resources and Evaluation*. 2025. Vol. 59, no. 4. P. 4313–4347. DOI: 10.1007/s10579-025-09828-1.

14. Lipianina-Honcharenko K., Ihnatiev I., Boguta G., Drakokhrust T., Telka M. Method for Determining the Sentiment of Foreign News about Ukraine. *CEUR Workshop Proceedings*. 2025. Vol. 3974. P. 185–195.

15. Lipianina-Honcharenko K., Ihnatiev I., Bohuta H., Yurkiv K., Novosad S. Rule-based method for aligning media content with Ukrainian legislation. CEUR Workshop Proceedings. 2025. Vol. 4004. P. 301–311.

16. Lipianina-Honcharenko K., Komar M., Bohuta H., Ihnatiev I., Yurkiv K., Illiashenko O., Bilovus L. A general method for real-time detection of information threats with a Ukraine case study. *Radioelectronic and Computer Systems*. 2025. № 3. С. 202–230.

17. Lipianina-Honcharenko K., Lendiuk D., Melnyk N., Komar M., Lendiuk T. Evaluation of the Keyword Selection Methods Effectiveness for the Fake News Classification. *IT&I*. 2024. С. 109–122.

18. Lipianina-Honcharenko K., Melnyk N., Ivasechko A., Telka M., Illiashenko O. Neural Network Ensemble Method for Deepfake Classification Using

Golden Frame Selection. *Big Data and Cognitive Computing*. 2025. Vol. 9, no. 4. Art. 109.

19. Lipianina-Honcharenko K., Telka M., Melnyk N. Comparison of ResNet, EfficientNet, and Xception architectures for deepfake detection. *CEUR Workshop Proceedings*. 2025. Vol. 3899. P. 26–34.

20. Mohammad S. M., Bravo-Marquez F. Emotion Intensities in Tweets. *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*. 2017. P. 65–77.

21. Mohammad S. M., Bravo-Marquez F., Salameh M., Kiritchenko S. SemEval-2018 Task 1: Affect in Tweets. *Proceedings of The 12th International Workshop on Semantic Evaluation (SemEval-2018)*. 2018. P. 1–17.

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

23. Munir H. S., Ren S., Mustafa M., Siddique C. N., Qayyum S. Attention based GRU-LSTM for software defect prediction. *PLOS ONE*. 2021. Vol. 16, no. 3. DOI: 10.1371/journal.pone.0247444.

24. Nandwani P., Verma R. A review on sentiment analysis and emotion detection from text. *Social Network Analysis and Mining*. 2021. Vol. 11, art. 81. DOI: 10.1007/s13278-021-00776-6.

25. Öhman E., Pàmies M., Kajava K., Tiedemann J. XED: A Multilingual Dataset for Sentiment Analysis and Emotion Detection. *Proceedings of COLING 2020*. 2020. P. 1009–1018.

26. Pereira P., Moniz H., Carvalho J. P. Deep emotion recognition in textual conversations: a survey. *Artificial Intelligence Review*. 2025. Vol. 58, no. 1. Art. 10. DOI: 10.1007/s10462-024-11010-y.

27. Polignano M., Basile P., de Gemmis M., Semeraro G. A comparison of word-embeddings in emotion detection from text using BiLSTM, CNN and self-attention. *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*. 2019. DOI: 10.1145/3314183.3324983.

28. Prytula M., Olenych I. Detection of Aggressive Rhetoric in Text Using Machine Learning Algorithms. *Electronics and Information Technologies*. 2023. Issue 22. DOI: 10.30970/eli.22.4.

29. Ray S. A Quick Review of Machine Learning Algorithms. *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*. 2019.

30. Sachin S., Tripathi A., Mahajan N., Aggarwal S., Nagrath P. Sentiment analysis using gated recurrent neural networks. *SN Computer Science*. 2020. Vol. 1, no. 2. DOI: 10.1007/s42979-020-0076-y.

31. Sailunaz K., Dhaliwal M., Rokne J., Alhajj R. Emotion detection from text and speech: A survey. *Social Network Analysis and Mining*. 2018. Vol. 8, no. 1. DOI: 10.1007/s13278-018-0505-2.

32. Shahid F., Zameer A., Muneeb M. Predictions for COVID-19 with Deep Learning Models of LSTM, GRU and Bi-LSTM. *Chaos, Solitons & Fractals*. 2020. Vol. 140. Art. 110212. DOI: 10.1016/j.chaos.2020.110212.

33. Suhaimi N. S., Mountstephens J., Teo J. EEG-based emotion recognition: A state-of-the-art review of current trends and opportunities. *Computational Intelligence and Neuroscience*. 2020. Vol. 2020. P. 1–19. DOI: 10.1155/2020/8875426.

34. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser Ł., Polosukhin I. Attention Is All You Need. *Advances in Neural Information Processing Systems*. 2017. Vol. 30. P. 5998–6008.

35. Yu S., Wu N., Liu S., Gong X. Job insecurity and employees' extra-role behavior: Moderated mediation model of negative emotion and workplace friendship. *Frontiers in Psychology*. 2021. Vol. 12. DOI: 10.3389/fpsyg.2021.631062.

36. Залуцька О., Молчанова М., Мазурець О., Мельник О., Скрипник Т. Метод інтелектуального аналізу емоційної тональності текстової інформації для визначення поведінкових намірів нейромережевими засобами. *Вісник Хмельницького національного університету. Технічні науки*. 2023. Т. 1, № 5 (325). С. 67–73. DOI: 10.31891/2307-5732-2023-325-5-67-73.

37. Катрук К. О., Бабюк Н. П. Аналіз сучасних методів розпізнавання емоційних станів у текстових повідомленнях. *LIV Всеукраїнська науково-технічна конференція факультету інформаційних технологій та комп'ютерної інженерії*. Вінниця: ВНТУ, 2025.

38. Комар М., Ліпяніна-Гончаренко Х., Кіт І., Мадараш Р., Юрків Х. Інтелектуальний метод виявлення джерел мультилінгвальної дезінформації. *Вимірювальні та обчислювальні прилади в технологічних процесах*. 2023. № 2. С. 221–230.

39. Піцун О., Мельник Н., Ліпяніна-Гончаренко Х. Гібридний підхід до визначення діпфейків на основі людського обличчя. *Herald of Khmelnytskyi National University. Technical Sciences*. 2024. Vol. 341, no. 5. P. 371–375.

40. Мельник Н.Б., Шклярук Н.А. Виявлення емоцій у текстових повідомленнях на основі рекурентних нейронних мереж. Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення: матеріали Міжнародної наукової інтернет-конференції, м. Тернопіль, Україна, м. Ополе, Польща, 7-8 травня 2026 р. 2026. С. 57-59.

41. ISEAR Dataset. URL: https://www.kaggle.com/datasets/faisalsanto007/isear-dataset/data (дата звернення:04.05.2026).

42. Комар М.П., Саченко А.О., Васильків Н.М та ін. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.

43. Островерхов В. М., Біловус Л. І., Возьний К. З. та ін. Загальні методичні рекомендації з підготовки, оформлення, захисту та оцінювання кваліфікаційних робіт здобувачів вищої освіти першого (бакалаврського) і другого (магістерського) рівнів. Тернопіль: ЗУНУ, 2024. 83 с.

Додаток А

Лістинги програмного коду

Лістинг А.1 – Імпорт бібліотек

```
import re
import string
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report, accuracy_score,
precision_score, recall_score, f1_score

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, GRU, Dense, Dropout,
Bidirectional
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.callbacks import EarlyStopping
```

Лістинг А.2 – Завантаження набору даних

```
def load_dataset(file_path):
    """
    Завантаження набору даних із текстовими повідомленнями та мітками емоцій.
    Очікувані стовпці: 'text' і 'emotion'.
    """
    data = pd.read_csv(file_path)

    if 'text' not in data.columns or 'emotion' not in data.columns:
        raise ValueError("Набір даних повинен містити стовпці 'text' та
'emotion'.")

    data = data.dropna(subset=['text', 'emotion'])
    data['text'] = data['text'].astype(str)

    return data
```

Лістинг А.3 – Попереднє оброблення текстових повідомлень

```
def clean_text(text):
    """
    Попереднє оброблення тексту:
    - приведення до нижнього регістру;
    - видалення URL;
    - видалення цифр;
    - видалення пунктуації;
    - видалення зайвих пробілів.
    """
    text = text.lower()
    text = re.sub(r"http\S+|www\S+", " ", text)
    text = re.sub(r"\d+", " ", text)
    text = text.translate(str.maketrans("", "", string.punctuation))
    text = re.sub(r"\s+", " ", text).strip()

    return text
```

```
def preprocess_dataset(data):
    """
    Застосування попереднього оброблення до всіх текстових повідомлень.
    """
    data['clean_text'] = data['text'].apply(clean_text)
    return data
```

Лістинг А.4 – Кодування емоційних класів

```
def encode_labels(labels):
    """
    Кодування текстових міток емоцій у числовий формат.
    """
    label_encoder = LabelEncoder()
    integer_labels = label_encoder.fit_transform(labels)

    num_classes = len(label_encoder.classes_)
    categorical_labels = to_categorical(integer_labels, num_classes=num_classes)

    return categorical_labels, label_encoder, num_classes
```

Лістинг А.5 – Токенізація та формування числових послідовностей

```
def tokenize_texts(texts, max_words=10000, max_len=100):
    """
    Перетворення текстових повідомлень у числові послідовності.
    """
    tokenizer = Tokenizer(num_words=max_words, oov_token="<OOV>")
    tokenizer.fit_on_texts(texts)

    sequences = tokenizer.texts_to_sequences(texts)
    padded_sequences = pad_sequences(
        sequences,
        maxlen=max_len,
        padding='post',
        truncating='post'
    )

    return padded_sequences, tokenizer
```

Лістинг А.6 – Поділ даних на навчальну, валідаційну та тестову вибірки

```
def split_data(X, y):
    """
    Поділ даних на навчальну, валідаційну та тестову вибірки.
    Співвідношення: 70% / 15% / 15%.
    """
    X_train, X_temp, y_train, y_temp = train_test_split(
        X,
        y,
        test_size=0.30,
        random_state=42,
        stratify=y.argmax(axis=1)
    )

    X_val, X_test, y_val, y_test = train_test_split(
        X_temp,
        y_temp,
        test_size=0.50,
        random_state=42,
```

```
        stratify=y_temp.argmax(axis=1)
    )

    return X_train, X_val, X_test, y_train, y_val, y_test
```

Лістинг А.7 – Побудова моделі LSTM

```
def build_lstm_model(vocab_size, embedding_dim, max_len, num_classes):
    """
    Побудова моделі LSTM для класифікації емоцій.
    """
    model = Sequential()

    model.add(Embedding(
        input_dim=vocab_size,
        output_dim=embedding_dim,
        input_length=max_len
    ))

    model.add(LSTM(128))
    model.add(Dropout(0.3))

    model.add(Dense(64, activation='relu'))
    model.add(Dropout(0.2))

    model.add(Dense(num_classes, activation='softmax'))

    model.compile(
        loss='categorical_crossentropy',
        optimizer='adam',
        metrics=['accuracy']
    )

    return model
```

Лістинг А.8 – Побудова моделі BiLSTM

```
def build_bilstm_model(vocab_size, embedding_dim, max_len, num_classes):
    """
    Побудова моделі BiLSTM для класифікації емоцій.
    """
    model = Sequential()

    model.add(Embedding(
        input_dim=vocab_size,
        output_dim=embedding_dim,
        input_length=max_len
    ))

    model.add(Bidirectional(LSTM(128)))
    model.add(Dropout(0.3))

    model.add(Dense(64, activation='relu'))
    model.add(Dropout(0.2))

    model.add(Dense(num_classes, activation='softmax'))

    model.compile(
        loss='categorical_crossentropy',
        optimizer='adam',
        metrics=['accuracy']
    )
```

```
    return model
```

Лістинг А.9 – Побудова моделі GRU

```
def build_gru_model(vocab_size, embedding_dim, max_len, num_classes):
    """
    Побудова моделі GRU для класифікації емоцій.
    """
    model = Sequential()

    model.add(Embedding(
        input_dim=vocab_size,
        output_dim=embedding_dim,
        input_length=max_len
    ))

    model.add(GRU(128))
    model.add(Dropout(0.3))

    model.add(Dense(64, activation='relu'))
    model.add(Dropout(0.2))

    model.add(Dense(num_classes, activation='softmax'))

    model.compile(
        loss='categorical_crossentropy',
        optimizer='adam',
        metrics=['accuracy']
    )

    return model
```

Лістинг А.10 – Навчання моделі

```
def train_model(model, X_train, y_train, X_val, y_val, epochs=10,
batch_size=32):
    """
    Навчання моделі з використанням валідаційної вибірки.
    """
    early_stopping = EarlyStopping(
        monitor='val_loss',
        patience=3,
        restore_best_weights=True
    )

    history = model.fit(
        X_train,
        y_train,
        validation_data=(X_val, y_val),
        epochs=epochs,
        batch_size=batch_size,
        callbacks=[early_stopping],
        verbose=1
    )

    return history
```

Лістинг А.11 – Оцінювання якості моделі

```
def evaluate_model(model, X_test, y_test, label_encoder):
    """
```

```
    Оцінювання якості моделі на тестовій вибірці.
    """
    y_pred_prob = model.predict(X_test)
    y_pred = np.argmax(y_pred_prob, axis=1)
    y_true = np.argmax(y_test, axis=1)

    accuracy = accuracy_score(y_true, y_pred)
    precision = precision_score(y_true, y_pred, average='weighted',
zero_division=0)
    recall = recall_score(y_true, y_pred, average='weighted', zero_division=0)
    f1 = f1_score(y_true, y_pred, average='weighted', zero_division=0)

    print("Accuracy:", round(accuracy, 4))
    print("Precision:", round(precision, 4))
    print("Recall:", round(recall, 4))
    print("F1-score:", round(f1, 4))

    print("\nClassification report:")
    print(classification_report(
        y_true,
        y_pred,
        target_names=label_encoder.classes_,
        zero_division=0
    ))

    return {
        "accuracy": accuracy,
        "precision": precision,
        "recall": recall,
        "f1_score": f1
    }
```

Лістинг А.12 – Побудова графіків навчання

```
def plot_training_history(history, model_name):
    """
    Побудова графіків accuracy та loss для процесу навчання.
    """
    plt.figure(figsize=(8, 5))
    plt.plot(history.history['accuracy'], label='Training accuracy')
    plt.plot(history.history['val_accuracy'], label='Validation accuracy')
    plt.title(f'Accuracy for {model_name}')
    plt.xlabel('Epoch')
    plt.ylabel('Accuracy')
    plt.legend()
    plt.grid(True)
    plt.show()

    plt.figure(figsize=(8, 5))
    plt.plot(history.history['loss'], label='Training loss')
    plt.plot(history.history['val_loss'], label='Validation loss')
    plt.title(f'Loss for {model_name}')
    plt.xlabel('Epoch')
    plt.ylabel('Loss')
    plt.legend()
    plt.grid(True)
    plt.show()
```

Лістинг А.13 – Класифікація нового текстового повідомлення

```
def predict_emotion(text, model, tokenizer, label_encoder, max_len=100):
    """
    Визначення емоційного класу для нового текстового повідомлення.
```

```
    """
    cleaned_text = clean_text(text)
    sequence = tokenizer.texts_to_sequences([cleaned_text])
    padded_sequence = pad_sequences(
        sequence,
        maxlen=max_len,
        padding='post',
        truncating='post'
    )

    prediction = model.predict(padded_sequence)
    predicted_class_index = np.argmax(prediction, axis=1)[0]
    predicted_emotion =
label_encoder.inverse_transform([predicted_class_index])[0]

    confidence = float(np.max(prediction))

    return predicted_emotion, confidence
```

Лістинг А.14 – Основний сценарій роботи програмного модуля

```
def main():
    """
    Основний сценарій роботи програмного модуля.
    """
    file_path = "emotion_dataset.csv"

    max_words = 10000
    max_len = 100
    embedding_dim = 100
    epochs = 10
    batch_size = 32

    data = load_dataset(file_path)
    data = preprocess_dataset(data)

    y, label_encoder, num_classes = encode_labels(data['emotion'])

    X, tokenizer = tokenize_texts(
        data['clean_text'],
        max_words=max_words,
        max_len=max_len
    )

    X_train, X_val, X_test, y_train, y_val, y_test = split_data(X, y)

    vocab_size = min(max_words, len(tokenizer.word_index) + 1)

    models = {
        "LSTM": build_lstm_model(vocab_size, embedding_dim, max_len,
num_classes),
        "BiLSTM": build_bilstm_model(vocab_size, embedding_dim, max_len,
num_classes),
        "GRU": build_gru_model(vocab_size, embedding_dim, max_len, num_classes)
    }

    results = {}

    for model_name, model in models.items():
        print(f"\nTraining model: {model_name}")

        history = train_model(
            model,
            X_train,
```

```
            y_train,
            X_val,
            y_val,
            epochs=epochs,
            batch_size=batch_size
        )

        plot_training_history(history, model_name)

        metrics = evaluate_model(
            model,
            X_test,
            y_test,
            label_encoder
        )

        results[model_name] = metrics

        model.save(f"{model_name.lower()}_emotion_model.h5")

    results_df = pd.DataFrame(results).T
    results_df.to_csv("model_comparison_results.csv", index=True)

    print("\nПорівняльні результати моделей:")
    print(results_df)


if __name__ == "__main__":
    main()
```

Лістинг А.15 – Приклад тестування навченої моделі

```
test_messages = [
    "I am very happy today",
    "This situation makes me angry",
    "I feel sad and lonely",
    "I am afraid of what can happen next"
]

for message in test_messages:
    emotion, confidence = predict_emotion(
        text=message,
        model=models["GRU"],
        tokenizer=tokenizer,
        label_encoder=label_encoder,
        max_len=max_len
    )

    print("Text:", message)
    print("Predicted emotion:", emotion)
    print("Confidence:", round(confidence, 4))
    print("-" * 40)
```

Лістинг А.16 – Збереження результатів тестування

```
def save_predictions(messages, model, tokenizer, label_encoder, max_len=100):
    """
    Збереження результатів класифікації нових повідомлень у CSV-файл.
    """
    prediction_results = []

    for message in messages:
        emotion, confidence = predict_emotion(
```

```
            text=message,
            model=model,
            tokenizer=tokenizer,
            label_encoder=label_encoder,
            max_len=max_len
        )

        prediction_results.append({
            "text": message,
            "predicted_emotion": emotion,
            "confidence": round(confidence, 4)
        })

    result_df = pd.DataFrame(prediction_results)
    result_df.to_csv("emotion_predictions.csv", index=False)

    return result_df
```

Додаток Б

Апробація результатів роботи

УДК 001 (063)

Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення (випуск 110)*:* матеріали Міжнародної наукової інтернет-конференції, (м. Тернопіль, Україна, м. Ополе, Польща, 7-8 травня 2026 р.) / редкол. : О. Патряк та ін. ГО "Наукова спільнота", WSZIA w Opolu. Тернопіль : ФО-П Шпак В.Б. 2026. 195 с. – ISSN 2522-932X

Збірник доповідей підготовлено за матеріалами Міжнародної наукової інтернет-конференції (випуск 110) 7-8 травня 2026 р. на сайті www.konferenciaonline.org.ua

Наша адреса: Оргкомітет МНІК "Конференція онлайн"
а/с 797, м. Тернопіль 46005
тел. моб. 068 366 0 525
e-mail: inetkonf@ukr.net

URL Інтернет-конференції: http://www.konferenciaonline.org.ua/
ISSN 2522-932X

Мельник Назар Борисович,
молодший науковий співробітник, викладач,
Західноукраїнський національний університет, м. Тернопіль

Шклярук Назарій Анатолійович, *бакалавр,*
Західноукраїнський національний університет, м. Тернопіль

ВИЯВЛЕННЯ ЕМОЦІЙ У ТЕКСТОВИХ ПОВІДОМЛЕННЯХ НА ОСНОВІ РЕКУРЕНТНИХ НЕЙРОННИХ МЕРЕЖ

Інтернет-адреса публікації на сайті:
http://www.konferenciaonline.org.ua/ua/article/id-2578/

Виявлення емоцій у тексті є складнішою задачею, ніж звичайний аналіз тональності. Якщо під час аналізу тональності зазвичай визначається загальна полярність висловлювання, то в задачі emotion detection необхідно встановити конкретний емоційний стан або емоційний клас, наприклад радість, страх, сум, гнів, провину чи відразу. Такий підхід потребує точнішого урахування контексту, послідовності слів, синтаксичних зв'язків і семантичних особливостей висловлювання [1]. Додаткову складність формують багатозначність природної мови, іронія, сарказм, скорочення, неформальна лексика, змішані емоції та залежність інтерпретації від предметної області [2].

На рисунку 1 зображено чотири логічні рівні функціонування програмного модуля: рівень вхідних даних, рівень підготовки тексту, рівень моделей класифікації та рівень вихідних результатів.

Рисунок 1 – Архітектура програмного модуля

Для підвищення наочності також подано рисунок 2, на якому зображено структурну схему взаємодії компонентів програмного модуля.

На рисунку 3 подано загальну схему алгоритмічного забезпечення програмного модуля. У межах цієї схеми відображено послідовність

переходу від вхідного текстового повідомлення до визначення емоційного класу та формування метрик якості.

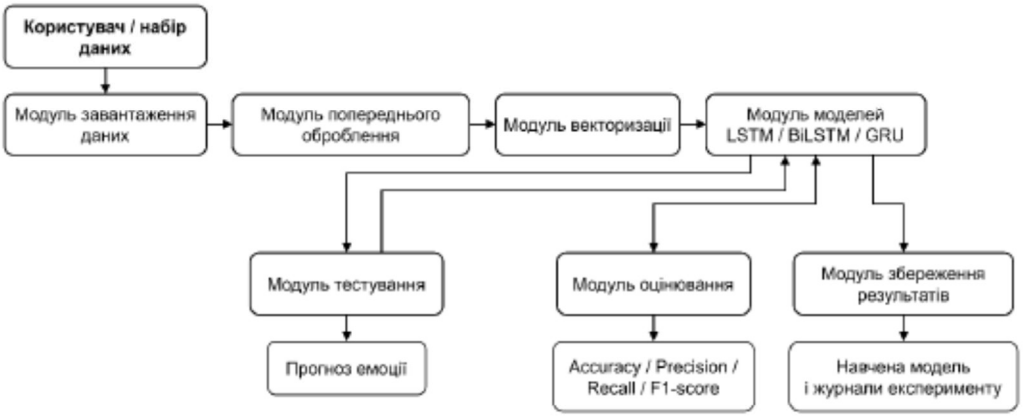


Рисунок 2 – Схема взаємодії компонентів програмного модуля

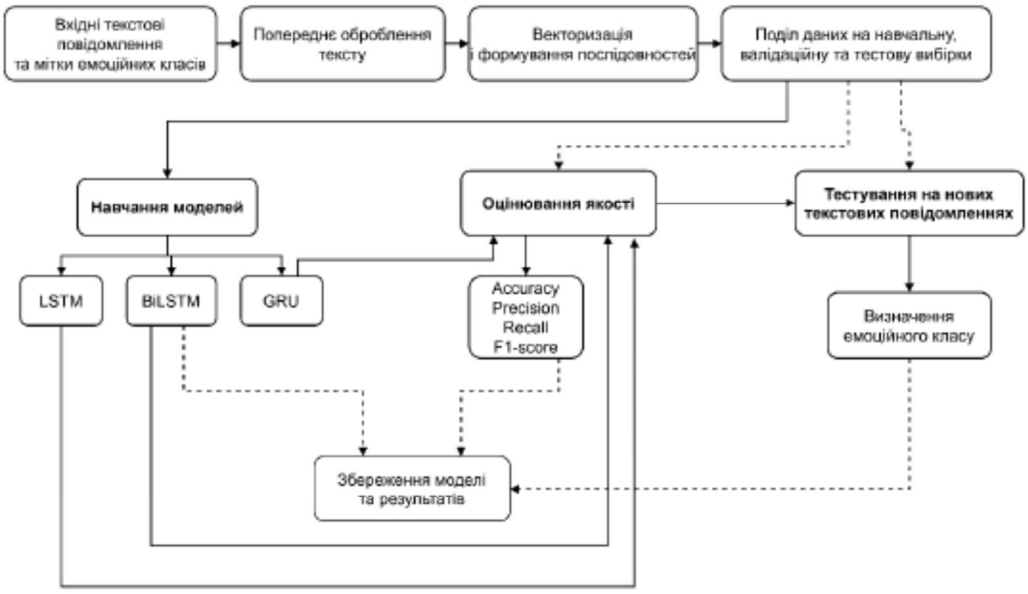


Рисунок 3 – Схема алгоритмічного забезпечення програмного модуля

Першим етапом є попереднє оброблення текстових повідомлень. Його призначення полягає у приведенні вхідного тексту до уніфікованого формату, придатного для подальшого машинного аналізу. На цьому етапі виконуються очищення тексту від зайвих символів, нормалізація регістру, токенізація, видалення надлишкових елементів та перетворення тексту у послідовність, що може бути подана на вхід нейронній мережі. Саме

якість цього етапу впливає на стабільність навчання і підсумкову точність класифікації [1, 2].

Другим етапом є алгоритм навчання моделей. Після формування числового подання тексту дані надходять до одного з рекурентних класифікаторів. У межах роботи передбачено використання трьох моделей – LSTM, BiLSTM і GRU. Для кожної з них застосовується однакова схема навчання, що забезпечує коректність подальшого порівняння. На цьому етапі виконується побудова архітектури моделі, задання функції втрат, вибір оптимізатора, запуск процесу навчання та збереження найкращих параметрів [3, 4].

Третім етапом є алгоритм тестування та оцінювання. Його завдання полягає у використанні навченої моделі для класифікації нових текстових повідомлень і визначенні якості роботи системи. Для цього тестові дані проходять ті самі етапи попереднього оброблення, після чого подаються на вхід моделі. Результатом є передбачений емоційний клас. Після цього обчислюються метрики accuracy, precision, recall і F1-score, які дають змогу оцінити якість багатокласової класифікації [1, 2].

Отже, сформовано архітектуру програмного модуля виявлення емоцій у текстових повідомленнях. Вона базується на модульному принципі, охоплює всі основні етапи оброблення текстових даних і забезпечує можливість реалізації, навчання та порівняння моделей LSTM, BiLSTM і GRU.

Література:

1. Acheampong F. A., Wenyu C., Nunoo-Mensah H. Text-based emotion detection: Advances, challenges, and opportunities. *Engineering Reports*. 2020. Vol. 2, no. 7. DOI: 10.1002/eng2.12189.

2. Chutia T., Baruah N. A review on emotion detection by using deep learning techniques. *Artificial Intelligence Review*. 2024. Vol. 57. Art. 203. DOI: 10.1007/s10462-024-10831-1.

3. Jang B., Kim M., Harerimana G., Kang S., Kim J. W. BI-LSTM model to increase accuracy in text classification: Combining Word2Vec, CNN and attention mechanism. *Applied Sciences*. 2020. Vol. 10, no. 17. Art. 5841. DOI: 10.3390/app10175841.

4. Munir H. S., Ren S., Mustafa M., Siddique C. N., Qayyum S. Attention based GRU-LSTM for software defect prediction. *PLOS ONE*. 2021. Vol. 16, no. 3. DOI: 10.1371/journal.pone.0247444.