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

СТРИЖЕУС Матвій Юрійович

Програмний модуль виявлення фейкових новин та рекомендацій достовірного контенту на основі глибоких нейронних мереж / Software Module for Detecting Fake News and Recommending Reliable Content Based on Deep Neural Networks

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

Кваліфікаційна робота

Виконав студент групи КН-42
М. Ю. Стрижеус

_____

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
СТРИЖЕУСУ Матвію Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи
Програмний модуль виявлення фейкових новин та рекомендацій достовірного контенту на основі глибоких нейронних мереж / Software Module for Detecting Fake News and Recommending Reliable Content Based on Deep 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 фейкових новин і рекомендації достовірного контенту;
- дослідити існуючі методи класифікації фейкових новин і підходи до рекомендаційних систем;
- обґрунтувати вибір методу подання тексту та моделі глибокого навчання для виявлення фейкових новин;
- спроєктувати архітектуру програмного модуля;
- реалізувати модуль попередньої обробки новинного тексту;
- реалізувати модель класифікації новин за рівнем достовірності;
- реалізувати модуль рекомендації достовірного контенту;
- провести експериментальне дослідження роботи модуля та оцінити його ефективність.
5. Перелік графічного матеріалу в роботі
- архітектура програмного модуля виявлення фейкових новин та рекомендації достовірного контенту;
- схема процесу обробки новинного контенту;
- структурна схема моделі класифікації та рекомендацій;
- структурна схема програмної реалізації модуля.

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_____ М. Ю. Стрижеус
підпис

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

АНОТАЦІЯ

Кваліфікаційна робота на тему «Програмний модуль виявлення фейкових новин та рекомендацій достовірного контенту на основі глибоких нейронних мереж» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом в 76 сторінок і містить 17 ілюстрацій, 13 таблиць, 3 додатки та 41 використане джерело.

Метою кваліфікаційної роботи є розроблення програмного модуля виявлення фейкових новин та рекомендацій достовірного контенту на основі глибоких нейронних мереж.

Методи дослідження включають методи системного аналізу, попереднього оброблення текстових даних, очищення та нормалізації новинного контенту, токенізації, векторизації, машинного навчання та класифікації текстів на основі глибоких нейронних мереж. Також використано методи моделювання програмної архітектури, експериментального тестування, оцінювання якості класифікації.

Розроблено програмний модуль виявлення фейкових новин та рекомендацій достовірного контенту на основі глибоких нейронних мереж, який забезпечує завантаження вхідних новинних повідомлень, перевірку та очищення тексту, нормалізацію даних, формування числового подання тексту, токенізацію, класифікацію новин, а також добір тематично близьких достовірних матеріалів.

Результати дослідження можуть бути використані для створення та вдосконалення програмних засобів автоматизованого аналізу новинного контенту, виявлення фейкових повідомлень, зниження впливу дезінформації, підтримки користувача під час оцінювання достовірності інформації та формування рекомендацій перевірених інформаційних матеріалів.

Ключові слова: ФЕЙКОВІ НОВИНИ, ДОСТОВІРНИЙ КОНТЕНТ, ГЛИБОКІ НЕЙРОННІ МЕРЕЖІ, МАШИННЕ НАВЧАННЯ, КЛАСИФІКАЦІЯ ТЕКСТУ, ПОПЕРЕДНЄ ОБРОБЛЕННЯ ДАНИХ, ТОКЕНІЗАЦІЯ, ВЕКТОРИЗАЦІЯ, РЕКОМЕНДАЦІЙНИЙ МОДУЛЬ, ДЕЗІНФОРМАЦІЯ.

ANNOTATION

Qualification work on the topic «Software Module for Detecting Fake News and Recommending Reliable Content Based on Deep Neural Networks» for Bachelor's degree on speciality 122 «Computer Science» educational and professional program «Computer Science» is written on 76 pages and it contains 17 figures, 13 tables, 3 annexes and 41 sources.

The purpose of the qualification thesis is to develop a software module for fake news detection and recommendation of reliable content based on deep neural networks.

The research methods include system analysis, preliminary processing of textual data, cleaning and normalization of news content, tokenization, vectorization, machine learning, and text classification based on deep neural networks. Methods of software architecture modeling, experimental testing, and classification quality evaluation were also used.

A software module for fake news detection and recommendation of reliable content based on deep neural networks has been developed. It provides loading of input news messages, text verification and cleaning, data normalization, formation of a numerical representation of text, tokenization, news classification, and selection of thematically related reliable materials.

The research results can be used to create and improve software tools for automated analysis of news content, detection of fake messages, reduction of the impact of disinformation, user support in assessing the reliability of information, and formation of recommendations for verified information materials.

Keywords: FAKE NEWS, RELIABLE CONTENT, DEEP NEURAL NETWORKS, MACHINE LEARNING, TEXT CLASSIFICATION, DATA PREPROCESSING, TOKENIZATION, VECTORIZATION, RECOMMENDATION MODULE, DISINFORMATION.

ЗМІСТ

ВСТУП

У сучасному інформаційному середовищі новинний контент поширюється дуже швидко, а перевірка його достовірності часто відстає від темпів публікації та репостів. Через це фейкові новини, маніпулятивні повідомлення та змішаний контент із ознаками дезінформації здатні швидко впливати на громадську думку, поведінку користувачів і загальний рівень довіри до цифрових медіа [1–3, 9]. Особливо гостро ця проблема проявляється в соціальних мережах і новинних платформах, де поширення інформації залежить не лише від змісту повідомлення, а й від емоційної реакції аудиторії, ідеологічних уподобань, рівня довіри до джерела та особливостей алгоритмічної подачі контенту [5, 6, 8].

Актуальність теми зумовлена тим, що фейкові новини вже давно є не окремими випадками помилкових публікацій, а системним явищем цифрового інформаційного простору. У наукових працях наголошується, що поняття «фейкові новини» є багатовимірним і охоплює не лише повністю неправдиві повідомлення, а й маніпулятивно поданий, частково спотворений або контекстуально хибний контент [2, 7, 9]. Це ускладнює задачу автоматичного виявлення, оскільки система має аналізувати не тільки окремі слова чи фрази, а й структуру тексту, семантику, джерело, контекст поширення та поведінкові сигнали користувачів [10, 11, 29, 34].

Додатковою проблемою є те, що виявлення фейкових новин саме по собі не повністю вирішує задачу протидії дезінформації. Якщо система лише позначає матеріал як сумнівний або неправдивий, користувач нерідко залишається без якісної альтернативи. У такому разі зниження впливу недостовірного контенту виявляється обмеженим. Саме тому практичний інтерес становлять підходи, які поєднують класифікацію новин із рекомендацією перевірених або більш надійних інформаційних матеріалів [14, 21, 27]. Такий підхід дає змогу не лише виявляти ризиковий контент, а й формувати для користувача безпечніше інформаційне середовище.

Сучасні дослідження показують, що для задачі виявлення фейкових новин ефективними є методи машинного та глибокого навчання. У цій сфері застосовуються класичні підходи на основі текстових ознак, TF-IDF і векторних

подань слів, а також складніші моделі на базі CNN, LSTM, BERT, графових нейронних мереж і мультимодальних архітектур [1, 4, 10, 15, 18, 19, 24, 26, 33]. Перевага глибоких нейронних мереж полягає в тому, що вони здатні автоматично виявляти приховані закономірності в тексті, враховувати порядок слів, семантичні залежності та складні комбінації ознак, які важко формалізувати вручну [10, 19, 33]. Водночас навіть точна модель класифікації не гарантує корисного результату для кінцевого користувача без продуманого механізму подальшої рекомендації контенту.

Окремої уваги потребує український контекст. Для України проблема дезінформації має не лише інформаційний, а й суспільний та безпековий вимір. Це пояснює зростання інтересу до методів виявлення інформаційних загроз у реальному часі, класифікації фейкових новин, аналізу тональності новин про Україну, виявлення джерел мультилінгвальної дезінформації та побудови моделей онлайн-протидії неправдивому контенту [14, 15, 35–37]. У таких умовах розроблення програмного модуля, який поєднує автоматичне виявлення фейкових новин і рекомендацію достовірного контенту, є доцільним як з наукового, так і з прикладного погляду.

Аналіз наукових джерел свідчить, що в наявних роботах увага часто зосереджується або на класифікації новин, або на окремих аспектах рекомендаційних систем [10, 17, 21, 27]. Водночас меншою мірою розглядається інтегрований програмний підхід, у якому модуль виявлення фейків і модуль рекомендації достовірного контенту функціонують як єдина система. Саме така інтеграція є важливою для побудови прикладного програмного рішення, придатного для використання в новинних сервісах, аналітичних платформах або допоміжних системах інформаційної фільтрації.

Метою кваліфікаційної роботи є розроблення програмного модуля виявлення фейкових новин та рекомендацій достовірного контенту на основі глибоких нейронних мереж.

Для досягнення поставленої мети необхідно розв’язати такі завдання:

– проаналізувати предметну область виявлення фейкових новин і рекомендації достовірного контенту;

– дослідити існуючі методи класифікації фейкових новин і підходи до рекомендаційних систем;

– обґрунтувати вибір методу подання тексту та моделі глибокого навчання для виявлення фейкових новин;

– спроєктувати архітектуру програмного модуля;

– реалізувати модуль попередньої обробки новинного тексту;

– реалізувати модель класифікації новин за рівнем достовірності;

– реалізувати модуль рекомендації достовірного контенту;

– провести експериментальне дослідження роботи модуля та оцінити його ефективність.

Об’єктом дослідження є процес автоматизованого аналізу новинного контенту в цифровому інформаційному середовищі.

Предметом дослідження є методи, моделі та програмні засоби виявлення фейкових новин і рекомендації достовірного контенту на основі глибоких 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 ПРОБЛЕМНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Опис предметної області

Предметна область цієї кваліфікаційної роботи охоплює процеси автоматичного аналізу новинного контенту, виявлення фейкових повідомлень та подальшої рекомендації достовірних інформаційних матеріалів. У центрі уваги перебуває цифрове інформаційне середовище, у якому новини поширюються через новинні сайти, соціальні мережі, месенджери, агрегатори та інші онлайн-платформи. Саме в такому середовищі швидкість поширення повідомлень часто є вищою, ніж швидкість їх перевірки, що створює сприятливі умови для розповсюдження недостовірної інформації [1, 3, 5, 10].

Фейкові новини є складним інформаційним явищем, яке не зводиться лише до повністю неправдивих тверджень. У наукових джерелах підкреслюється, що до цієї категорії можуть належати повністю вигадані повідомлення, маніпулятивно змінені матеріали, контент із хибним контекстом, емоційно викривлені інтерпретації фактів, а також повідомлення, у яких правдиві фрагменти поєднано з дезінформаційними вставками [2, 7, 9]. Через це задача виявлення фейкових новин є складнішою, ніж звичайна текстова класифікація. У такому випадку недостатньо виявити окремі ключові слова або формальні мовні ознаки. Необхідно враховувати зміст повідомлення, його подачу, семантичні зв’язки між фрагментами тексту, контекст поширення та, за можливості, характеристики джерела [10, 11, 29, 34].

У межах предметної області важливо розрізняти кілька пов’язаних понять. По-перше, це недостовірний контент, тобто повідомлення, зміст якого не відповідає фактам або суттєво їх спотворює. По-друге, це дезінформація, яка передбачає цілеспрямоване поширення неправдивого або маніпулятивного контенту з певною метою. По-третє, це інформаційні загрози, що включають системне використання фейкових повідомлень для впливу на суспільні настрої, політичні процеси, репутацію організацій або поведінку окремих груп користувачів [2, 9, 14, 35]. Для програмного модуля, що розробляється в цій роботі, усі ці поняття мають прикладне значення, оскільки система повинна працювати в середовищі, де неправдива інформація часто маскується під звичайний новинний

матеріал.

Проблема виявлення фейкових новин безпосередньо пов’язана з особливостями сучасного медіаспоживання. Значна частина користувачів отримує інформацію не з першоджерел, а через стрічки рекомендацій, репости, короткі анонси або фрагменти матеріалів, які швидко з’являються в інформаційному потоці. У таких умовах рішення про довіру до новини часто приймається не після глибокого аналізу змісту, а на основі заголовка, емоційного забарвлення, візуального оформлення або відповідності повідомлення попереднім переконанням користувача [3, 5, 6]. Це підвищує ризик поширення фейкового контенту, особливо якщо він викликає сильну емоційну реакцію або подається як термінова, резонансна чи суспільно важлива новина [1, 8].

З технічного погляду предметна область поєднує кілька взаємопов’язаних задач. Першою є задача збору та підготовки даних. Для побудови системи виявлення фейкових новин необхідно працювати з масивами текстових повідомлень, які можуть надходити з різних джерел і мати різну структуру, обсяг, мову, стиль і рівень формальності. Такі дані потребують очищення, нормалізації, токенізації, усунення шуму та приведення до формату, придатного для машинної обробки [11, 14, 15]. Другою задачею є представлення тексту у вигляді ознак, зручних для моделі. Для цього використовуються як класичні підходи, наприклад TF-IDF, так і сучасні щільні векторні подання слів і контекстні embeddings [1, 15, 19].

Третьою задачею є власне класифікація новинного контенту. Вона полягає у визначенні, чи є повідомлення достовірним, фейковим або потенційно маніпулятивним. У межах цієї задачі застосовуються алгоритми машинного навчання та моделі глибокого навчання, здатні виявляти приховані закономірності в тексті [10, 18, 19, 33]. У сучасних дослідженнях для цього використовуються згорткові нейронні мережі, рекурентні нейронні мережі, моделі типу LSTM, трансформери та гібридні архітектури [4, 10, 18, 19]. Перевага таких моделей полягає в тому, що вони дають змогу враховувати не лише наявність окремих лексем, а й порядок слів, локальні та глобальні семантичні залежності, а також стилістичні особливості тексту.

Однак предметна область не обмежується лише задачею класифікації. Для практичного застосування важливо не тільки позначити новину як фейкову або підозрілу, а й надати користувачеві більш надійну альтернативу. Саме тому другою ключовою складовою предметної області є рекомендація достовірного контенту. Її сутність полягає у доборі новин або інформаційних матеріалів, що відповідають темі запиту чи переглянутого повідомлення, але походять із надійніших джерел або мають вищий рівень достовірності [21, 27]. Такий підхід є важливим, оскільки він дозволяє не просто фільтрувати ризиковий контент, а формувати для користувача якіснішу інформаційну альтернативу.

Отже, у межах цієї роботи предметна область має подвійний характер. З одного боку, вона належить до сфери обробки природної мови та аналізу текстових даних. З іншого боку, вона пов’язана з рекомендаційними системами, які орієнтовані на персоналізований або тематично релевантний добір матеріалів [21, 27]. У такому поєднанні формується прикладна задача розроблення програмного модуля, здатного аналізувати новинний текст, виявляти ознаки фейковості та пропонувати достовірний контент за схожою тематикою.

Особливістю цієї предметної області є також багаторівневий характер ознак, за якими можна виявляти недостовірний контент. Частина з них належить до текстового рівня: лексика, синтаксис, емоційне забарвлення, логічна зв’язність, наявність сенсаційних конструкцій або маніпулятивних формулювань. Інша частина пов’язана з контекстом: джерело публікації, час поширення, реакція аудиторії, характер коментарів, кількість поширень, структура взаємодії між користувачами [10, 11, 18, 26, 28]. У багатьох роботах доведено, що поєднання змістових і контекстних ознак забезпечує вищу точність виявлення фейкових новин, ніж використання лише тексту [10, 11, 18, 28]. Водночас для бакалаврської роботи доцільно зосередитися насамперед на текстовому компоненті з можливістю подальшого розширення системи.

Ще однією важливою рисою предметної області є динамічність. Інформаційний простір постійно змінюється: з’являються нові теми, нові джерела, нові способи маніпулятивної подачі матеріалу. Через це програмний модуль повинен бути достатньо гнучким, щоб його можна було адаптувати до нових

наборів даних, інших тематичних категорій або змінених умов функціонування. Саме тому у сучасних дослідженнях велика увага приділяється не лише точності окремої моделі, а й можливості її масштабування, перенавчання та інтеграції в реальні інформаційні системи [10, 14, 17].

Для українського інформаційного простору ця предметна область має особливе значення. Поширення дезінформації, маніпулятивних повідомлень і ворожих інформаційних впливів створює додаткові вимоги до систем автоматичного аналізу контенту. У вітчизняних дослідженнях розглядаються задачі виявлення інформаційних загроз у реальному часі, аналізу тональності іноземних новин про Україну, визначення джерел мультилінгвальної дезінформації та побудови моделей онлайн-виявлення недостовірних повідомлень [14, 15, 35–37]. Це свідчить про практичну важливість розроблення таких програмних засобів і підтверджує доцільність обраної теми.

Узагальнюючи, предметна область дослідження охоплює: новинний цифровий контент як об’єкт аналізу; фейкові новини та дезінформацію як ціль виявлення; методи обробки природної мови та глибокі нейронні мережі як інструмент аналізу; рекомендацію достовірного контенту як механізм зниження впливу недостовірних повідомлень. Саме на перетині цих складових формується задача розроблення програмного модуля, що поєднує функції автоматичного виявлення фейкових новин і добору надійних альтернативних матеріалів.

1.2 Аналіз існуючих підходів і методів виявлення фейкових новин та рекомендації контенту

Задача виявлення фейкових новин належить до складних задач аналізу текстової інформації, оскільки недостовірність повідомлення не завжди виражається прямо. У багатьох випадках фейковий матеріал має зовнішні ознаки звичайної новини, містить окремі правдиві фрагменти, емоційно насичені формулювання або маніпулятивне трактування фактів [2, 7, 9]. Через це для його виявлення використовуються різні групи методів, які відрізняються типом ознак, рівнем складності та вимогами до даних.

Умовно сучасні підходи до виявлення фейкових новин можна поділити на кілька основних груп: лінгвістичні та ознакові методи, класичні методи машинного навчання, методи глибокого навчання, контекстно-орієнтовані підходи, а також комбіновані моделі, що поєднують змістові й поведінкові ознаки [10, 17, 29, 34]. Окремо слід розглядати методи рекомендації контенту, які в цій роботі мають допоміжну, але важливу роль, оскільки забезпечують добір достовірних матеріалів після класифікації новини.

Найпростішими є підходи, засновані на ручному виділенні текстових ознак. У таких методах аналізуються частоти слів, довжина речень, наявність емоційно забарвленої лексики, повторів, знаків оклику, сенсаційних формулювань, а також інші стилістичні характеристики [9, 10]. До цієї групи належать моделі, що працюють із поданням тексту у вигляді вектора ознак, сформованого, наприклад, за допомогою TF-IDF. Перевага таких підходів полягає у простоті реалізації, відносно невисоких обчислювальних витратах і зрозумілості отриманих результатів. Водночас їх обмеження є суттєвими. Вони слабко враховують контекст, не моделюють глибинні семантичні зв’язки між словами та погано працюють у випадках, коли фейковість визначається не окремою лексикою, а способом подачі змісту [1, 10, 15].

Практика показує, що ознакові методи часто використовуються як базовий рівень аналізу або як етап попереднього відбору релевантних слів. Зокрема, у роботах, присвячених класифікації новин, підкреслюється важливість коректного відбору ключових слів, оскільки саме цей етап впливає на інформативність вхідних даних для моделі [1, 15]. У вітчизняних дослідженнях також розглянуто ефективність різних підходів до добору ключових ознак для класифікації фейкових новин, що підтверджує доцільність використання такого етапу у спрощених або комбінованих системах [15].

Наступну групу становлять класичні методи машинного навчання. До них належать логістична регресія, метод опорних векторів, дерева рішень, наївний баєсівський класифікатор, випадковий ліс та інші алгоритми, що працюють на основі попередньо сформованих числових ознак [10, 17, 29]. Їх перевагою є порівняно швидке навчання, стійкість у задачах середньої складності та

можливість отримувати прийнятну якість на невеликих наборах даних. Такі моделі часто використовуються як базові для порівняння з нейронними мережами. Проте їхня ефективність знижується в ситуаціях, коли потрібно враховувати складну послідовну структуру тексту або приховані семантичні залежності між фрагментами повідомлення [10, 17].

Більш розвиненим напрямом є використання методів глибокого навчання. Саме вони сьогодні розглядаються як одні з найперспективніших для задачі виявлення фейкових новин [10, 18, 19, 33]. Глибокі нейронні мережі дозволяють автоматично формувати релевантні ознаки на основі тексту без необхідності жорстко задавати їх вручну. Це особливо важливо в задачах, де фейковість матеріалу визначається не лише окремими словами, а й структурою повідомлення, семантичними відношеннями та загальною стилістикою.

Серед нейронних підходів важливе місце посідають згорткові нейронні мережі. CNN добре працюють для виявлення локальних шаблонів у тексті, наприклад характерних словосполучень, коротких фраз або стійких маніпулятивних конструкцій [3, 18]. Перевага цього підходу полягає у здатності виділяти інформативні фрагменти незалежно від їх точного положення в тексті. Однак згорткові мережі менш ефективні для моделювання далеких залежностей між словами та реченнями, особливо якщо повідомлення має складну логічну структуру.

Для кращого врахування послідовності слів використовуються рекурентні нейронні мережі, зокрема LSTM. Такі моделі краще підходять для аналізу зв’язків у довших текстах, оскільки зберігають інформацію про попередні елементи послідовності [3, 10, 33]. Саме тому LSTM та їх модифікації досить часто застосовуються в задачах текстової класифікації. У контексті виявлення фейкових новин цей підхід є корисним тоді, коли важливим є не лише набір слів, а й послідовність викладу, логічні переходи, наявність суперечностей або характер подання фактів.

У практиці часто використовуються гібридні моделі типу CNN+LSTM, які поєднують переваги двох підходів: згорткові шари виділяють локальні текстові патерни, а рекурентні – аналізують послідовні залежності між ними [3, 14]. Така

архітектура є доцільною для розроблення прикладного програмного модуля, оскільки дозволяє поєднати достатню виразну силу моделі з відносно зрозумілою логікою її побудови. Крім того, саме подібні архітектури часто розглядаються як компроміс між класичними підходами та важчими трансформерними моделями.

Окремий напрям становлять трансформерні моделі, найвідомішим прикладом яких є BERT [4]. Їхня головна перевага полягає в можливості одночасно враховувати контекст слова з обох боків, що забезпечує глибше розуміння змісту тексту. На основі BERT було запропоновано спеціалізовані рішення для виявлення фейкових новин, зокрема FakeBERT, які демонструють високу якість класифікації [10, 19]. Такі моделі є потужними з погляду точності, однак мають і певні недоліки: значні обчислювальні витрати, вищі вимоги до апаратного забезпечення та складнішу інтеграцію в компактні прикладні системи. Для бакалаврської роботи це важливе обмеження, оскільки надмірна складність моделі не завжди виправдана в межах навчального проєкту.

Крім суто текстових методів, у сучасних дослідженнях активно розвиваються контекстно-орієнтовані підходи. Вони враховують не лише зміст повідомлення, а й реакцію користувачів, структуру поширення новини, зв’язки між авторами й аудиторією, часові характеристики та інші ознаки соціального контексту [11, 18, 26, 28]. Наприклад, графові нейронні мережі дозволяють моделювати поширення новини у вигляді графа взаємодій і використовувати цю структуру як додаткове джерело інформації для класифікації [18, 26]. Такі методи часто демонструють високу ефективність, але потребують значно складніших даних, ніж звичайний текстовий корпус. Саме тому їх застосування є доцільним у великих дослідницьких системах, тоді як для бакалаврської роботи доцільніше зосередитися на текстовому аналізі з можливістю майбутнього розширення.

Важливу роль у дослідженнях відіграють спеціалізовані набори даних. Одним із найбільш відомих ресурсів є FakeNewsNet, який містить текст новин, соціальний контекст і часову інформацію [11, 28]. Наявність таких наборів даних є суттєвою перевагою для навчання й перевірки моделей. Разом із цим варто враховувати, що поведінка моделей залежить від мови, тематики та джерел даних. Для українського інформаційного простору ця проблема є особливо актуальною,

оскільки англомовні датасети не повністю відображають особливості локального інформаційного середовища [14, 35]. Це підсилює значення робіт, орієнтованих на аналіз українського контенту та пов’язаних інформаційних загроз [14, 15, 35–37].

Порівнюючи основні групи методів, можна зазначити, що класичні моделі є простішими й швидшими, але менш гнучкими щодо складних текстових структур. Глибокі нейронні мережі забезпечують вищу якість, оскільки автоматично виявляють складні залежності, однак вимагають більшого обсягу даних і обчислювальних ресурсів [10, 17, 18, 19, 33]. Контекстні та графові підходи розширюють можливості аналізу, але підвищують складність системи та вимоги до вхідної інформації [11, 18, 26, 28]. Тому вибір конкретного підходу має залежати не лише від теоретичної точності, а й від практичної мети роботи.

Крім задачі виявлення фейків, у межах цієї кваліфікаційної роботи необхідно врахувати підходи до рекомендації контенту. Класичні рекомендаційні системи зазвичай будуються на основі контентного підходу, колаборативної фільтрації або їх комбінації. У контентному підході матеріали рекомендуються за змістовою схожістю з тими, які вже переглядав користувач. Колаборативна фільтрація спирається на подібність поведінки різних користувачів [21, 27]. Для задачі рекомендації достовірного контенту у системі виявлення фейкових новин найбільш доцільним є саме контентний або гібридний підхід, оскільки він дозволяє підібрати матеріали, близькі за тематикою до підозрілої новини, але з вищим рівнем довіри до джерела.

Наукові праці засвідчують, що рекомендаційні алгоритми можуть як зменшувати, так і підсилювати поширення дезінформації [6, 21]. Якщо система орієнтується лише на залучення користувача, вона може рекомендувати емоційно сильний, поляризуючий або сумнівний контент. Саме тому в задачі цієї роботи рекомендаційний модуль повинен враховувати не тільки релевантність за темою, а й ознаки достовірності матеріалу. Це означає, що рекомендація не повинна бути нейтральною в сенсі якості джерела. Навпаки, одним із критеріїв добору має бути вища надійність рекомендованого контенту порівняно з початковим повідомленням [21, 27].

У деяких дослідженнях запропоновано спеціалізовані рекомендаційні

системи для зниження впливу фейкових новин, які враховують довіру до джерел або результати фактчекінгу [27]. Такі підходи є концептуально близькими до теми цієї роботи. Проте для прикладного програмного модуля бакалаврського рівня доцільніше реалізувати простіший механізм, за якого після класифікації новини система знаходить тематично подібні матеріали серед корпусу перевіреного контенту й пропонує користувачеві найбільш релевантні варіанти. Такий підхід є технічно реалістичним, логічно зрозумілим і достатнім для демонстрації практичної цінності розробки.

Окремо варто звернути увагу на пояснюваність рішень моделі. У задачах, пов’язаних із достовірністю інформації, недостатньо отримати лише результат класифікації. Бажано також розуміти, які саме ознаки або фрагменти тексту найбільше вплинули на рішення моделі [24]. Пояснюваність є важливою не лише для дослідника, а й для кінцевого користувача, який повинен мати підстави довіряти рекомендації системи. Саме тому навіть у разі використання глибоких нейронних мереж доцільно передбачити хоча б базовий рівень інтерпретації результату, наприклад через виділення ключових фрагментів тексту або показ оцінки достовірності.

На основі проведеного аналізу можна зробити кілька узагальнень. По-перше, існуючі методи виявлення фейкових новин розвиваються від простих текстових ознак до складних багатокомпонентних нейронних архітектур [10, 17, 19, 34]. По-друге, жоден окремий підхід не є універсальним, оскільки ефективність моделі залежить від мови, типу даних, тематики новин і доступності соціального контексту [11, 14, 28]. По-третє, для прикладного програмного модуля доцільно обирати такий підхід, який поєднує достатню якість, відтворюваність і технічну реалізованість. Саме тому в межах цієї роботи обґрунтованим виглядає використання глибокої нейронної моделі для текстової класифікації новин із подальшим модулем рекомендації достовірного контенту на основі змістової подібності та ознак надійності джерела.

Для систематизації основних груп методів доцільно подати їх порівняння у табличній формі (таблиця 1.1).

Таблиця 1.1 – Порівняльна характеристика основних підходів до виявлення фейкових новин

Група методів	Основна ідея	Переваги	Недоліки
Ознакові методи	Використання лексичних, стилістичних і статистичних характеристик тексту	Простота реалізації, швидкість обробки	Слабке врахування контексту і семантики
Класичне машинне навчання	Класифікація на основі попередньо сформованих ознак	Добрі базові результати, невисокі обчислювальні витрати	Потреба в ручному доборі ознак
CNN	Виділення локальних шаблонів у тексті	Ефективність для коротких текстових фрагментів	Обмежене врахування далеких залежностей
LSTM	Аналіз послідовності слів і контексту	Краще врахування структури тексту	Складніше навчання, вищі обчислювальні витрати
CNN+LSTM	Поєднання локального та послідовного аналізу	Добрий баланс між якістю й складністю	Потреба в налаштуванні архітектури
BERT та трансформери	Контекстне подання тексту на основі механізму уваги	Висока точність, глибоке розуміння контексту	Великі обчислювальні ресурси
GNN і контекстні моделі	Урахування соціального контексту та структури поширення	Висока ефективність за наявності додаткових даних	Підвищена складність реалізації

Отже, аналіз існуючих підходів показує, що задача виявлення фейкових новин потребує поєднання методів обробки тексту, моделей класифікації та механізмів добору надійної інформаційної альтернативи. Найбільш перспективними для цієї роботи є підходи, що спираються на глибокі нейронні мережі для аналізу тексту та на контентно-орієнтовану рекомендацію для добору достовірних матеріалів. Саме така комбінація створює основу для побудови прикладного програмного модуля, орієнтованого на реальні потреби користувача.

1.3 Постановка задачі дослідження

Проведений аналіз предметної області показав, що проблема виявлення фейкових новин має складний характер і не зводиться лише до звичайної текстової класифікації. Недостовірний контент може бути повністю вигаданим, частково спотвореним, емоційно маніпулятивним або контекстуально хибним [2, 7, 9]. Крім того, практична цінність системи знижується, якщо після виявлення підозрілої новини користувач не отримує альтернативного достовірного матеріалу. Саме тому доцільно розглядати не окрему задачу класифікації, а інтегровану задачу, яка поєднує виявлення фейкових новин і рекомендацію перевіреного контенту [14, 21, 27].

У межах цієї кваліфікаційної роботи досліджується задача розроблення програмного модуля, призначеного для автоматизованого аналізу новинних текстів. Такий модуль повинен приймати на вхід текст новини, виконувати його попередню обробку, визначати рівень достовірності повідомлення за допомогою моделі глибокого навчання, а після цього формувати добірку тематично близьких матеріалів із вищим рівнем надійності. Отже, задача має дві функціонально пов’язані частини: класифікаційну та рекомендаційну.

Науково-технічна задача полягає в тому, щоб забезпечити достатню якість виявлення фейкових новин у межах реалістичного за складністю програмного рішення. З одного боку, сучасні трансформерні та графові моделі можуть забезпечувати високі результати, але потребують великих обсягів даних, суттєвих обчислювальних ресурсів і складнішої інтеграції [18, 19, 26]. З іншого боку, надто

прості методи на основі окремих ознак або базових статистичних представлень тексту не завжди здатні коректно виявляти складні форми маніпулятивного контенту [1, 10, 15]. У зв’язку з цим доцільно обрати такий підхід, який забезпечує баланс між точністю, відтворюваністю та можливістю програмної реалізації.

У роботі ставиться задача побудови програмного модуля, орієнтованого насамперед на текстовий аналіз новин. Це означає, що основним джерелом інформації для класифікації є текст повідомлення, а не соціальний граф поширення, мультимодальні ознаки або зовнішні мережеві сигнали. Такий вибір є обґрунтованим з кількох причин. По-перше, текст новини є базовим і найбільш доступним типом вхідних даних. По-друге, текстовий аналіз дає можливість побудувати компактний і відтворюваний програмний модуль. По-третє, для бакалаврської кваліфікаційної роботи такий рівень складності є достатнім, але водночас практично значущим [10, 14, 19].

Задача дослідження полягає у розробленні програмного модуля виявлення фейкових новин та рекомендацій достовірного контенту на основі глибоких нейронних мереж, який забезпечує:

– приймання новинного тексту як вхідних даних;

– попередню обробку тексту;

– перетворення тексту у форму, придатну для навчання та інференсу моделі;

– класифікацію новини за рівнем достовірності;

– формування рекомендацій достовірного контенту за тематичною подібністю;

– видачу результату користувачу у зручному вигляді.

Для розв’язання цієї задачі необхідно виконати низку взаємопов’язаних дослідницьких і прикладних кроків. Насамперед потрібно обґрунтувати вибір підходу до подання тексту та архітектури моделі. У сучасних роботах показано, що ефективність системи істотно залежить від способу представлення текстових даних і типу використаної нейронної архітектури [4, 10, 18, 19, 33]. У цій роботі доцільно орієнтуватися на модель, яка здатна враховувати семантичні та послідовні характеристики тексту, але не є надмірно складною для реалізації в межах програмного модуля.

Другим важливим аспектом є організація рекомендаційного блоку. Його призначення полягає не в загальній персоналізованій рекомендації новин, а саме в доборі достовірного контенту як альтернативи підозрілому або фейковому повідомленню. Отже, рекомендаційний модуль повинен працювати за принципом тематичної близькості та надійності джерела, а не лише за ознакою популярності чи попередньої поведінки користувачів [21, 27]. Це дозволяє сформувати більш прикладну й соціально корисну модель використання рекомендаційної системи.

У межах роботи доцільно визначити також основні вхідні та вихідні дані. Вхідними даними є текст новини або новинного повідомлення. За потреби до них можуть додаватися заголовок, короткий опис, ідентифікатор джерела або тематична категорія. Вихідними даними є: результат класифікації новини, оцінка її достовірності у вигляді класу або ймовірності, а також перелік рекомендованих достовірних матеріалів, близьких за тематикою до вхідного повідомлення.

До модуля можна висунути такі базові вимоги: коректна обробка текстових даних; можливість автоматичної класифікації новин; достатня точність виявлення фейкового контенту; формування релевантних і достовірних рекомендацій; прийнятний час обробки одного повідомлення; модульна структура, придатна до подальшого розширення.

З урахуванням зазначеного метою роботи є розроблення програмного модуля виявлення фейкових новин та рекомендацій достовірного контенту на основі глибоких нейронних мереж.

Для досягнення поставленої мети необхідно розв’язати такі завдання:

– проаналізувати предметну область виявлення фейкових новин і рекомендації достовірного контенту;

– дослідити існуючі методи класифікації фейкових новин і підходи до рекомендаційних систем;

– обґрунтувати вибір методу подання тексту та моделі глибокого навчання для виявлення фейкових новин;

– спроєктувати архітектуру програмного модуля;

– реалізувати модуль попередньої обробки новинного тексту;

– реалізувати модель класифікації новин за рівнем достовірності;

– реалізувати модуль рекомендації достовірного контенту;

– провести експериментальне дослідження роботи модуля та оцінити його ефективність.

Об’єктом дослідження є процес автоматизованого аналізу новинного контенту в цифровому інформаційному середовищі.

Предметом дослідження є методи, моделі та програмні засоби виявлення фейкових новин і рекомендації достовірного контенту на основі глибоких нейронних мереж.

2 ПРОЄКТУВАННЯ ПРОГРАМНОГО МОДУЛЯ ВИЯВЛЕННЯ ФЕЙКОВИХ НОВИН ТА РЕКОМЕНДАЦІЇ ДОСТОВІРНОГО КОНТЕНТУ

2.1 Обґрунтування вибору методів і засобів реалізації

Для розроблення програмного модуля виявлення фейкових новин та рекомендації достовірного контенту доцільно обрати такі методи й засоби, які забезпечують достатню точність, зрозумілу структуру системи та реальну можливість програмної реалізації в межах кваліфікаційної роботи. Вибір повинен враховувати не лише якість моделі, а й доступність даних, складність навчання та можливість інтеграції всіх компонентів у єдиний програмний модуль [10, 17, 34].

Основним типом вхідних даних у роботі є текст новини. Саме тому базою для побудови системи обрано методи обробки природної мови. Такий підхід є логічним, оскільки текст містить основну змістову інформацію, на підставі якої можна оцінити достовірність повідомлення. Використання лише текстових даних також спрощує реалізацію системи, оскільки не потребує додаткового збору графових, мультимодальних або поведінкових ознак. Це особливо важливо для прикладної системи бакалаврського рівня, де потрібно зберегти баланс між складністю та практичною цінністю результату [10, 11, 29].

Для подання тексту доцільно використовувати попередню текстову обробку, що включає очищення, нормалізацію, токенізацію та перетворення тексту у числове представлення. Без такого етапу текст не може бути безпосередньо поданий на вхід моделі. У роботі доцільно застосувати векторне подання слів або токенів, оскільки воно дозволяє зберегти змістові зв’язки між елементами тексту та передати їх моделі класифікації. У дослідженнях із текстової класифікації показано, що вибір способу подання тексту суттєво впливає на підсумкову якість моделі, а поєднання попередньої обробки з векторизацією є базовою умовою ефективного аналізу текстів [1, 15].

Для задачі виявлення фейкових новин обрано підхід на основі глибоких нейронних мереж. Це пояснюється тим, що такі моделі здатні автоматично виявляти складні закономірності у тексті, які важко задати вручну. На відміну від простих статистичних методів, нейронна мережа може враховувати не лише

частоту окремих слів, а й послідовність їх розміщення, характер поєднання та загальний зміст повідомлення. Саме цей підхід розглядається як один із найбільш перспективних у сучасних дослідженнях із виявлення фейкових новин [10, 19, 34].

В якості базової архітектури доцільно використати комбіновану модель типу CNN-LSTM. Згорткові шари дозволяють виявляти локальні текстові шаблони, наприклад характерні фрази або маніпулятивні конструкції. Шари LSTM, у свою чергу, враховують послідовність слів і загальний контекст речення. Поєднання цих двох підходів є практичним, оскільки дає змогу побудувати модель, що є достатньо потужною, але не надто складною для реалізації та навчання. Доцільність використання CNN і LSTM для аналізу послідовних даних підтверджується сучасними дослідженнями, а їх комбінування є поширеним рішенням у задачах текстової класифікації [3, 14, 33].

Використання важчих трансформерних моделей у цій роботі не є доцільним. Хоча моделі типу BERT часто забезпечують високу точність, вони потребують більших обчислювальних ресурсів, складнішого налаштування та значно збільшують обсяг реалізації [4, 19]. Для кваліфікаційної роботи доцільніше обрати архітектуру, яка забезпечує баланс між якістю результатів і технічною доступністю. Саме тому вибір на користь комбінованої моделі глибокого навчання є більш виправданим у межах поставленої задачі.

Для рекомендації достовірного контенту доцільно використати контентно-орієнтований підхід. Його суть полягає в тому, що після класифікації новини система підбирає матеріали, схожі за тематикою, але з достовірного корпусу джерел. Такий варіант є простішим і більш контрольованим, ніж складні персоналізовані рекомендаційні алгоритми. Крім того, він безпосередньо відповідає меті роботи: не просто показати інший матеріал, а запропонувати більш надійну альтернативу. Подібна логіка узгоджується з дослідженнями, у яких рекомендаційні системи розглядаються як інструмент зниження впливу дезінформації [21, 27].

В якості критерію добору рекомендацій доцільно використати змістову подібність між новинами. Для цього можна застосувати векторне представлення текстів і обчислення міри схожості. Перевага такого рішення полягає у простоті

реалізації та зрозумілій логіці роботи. Система порівнює вхідну новину з набором перевірених матеріалів і вибирає ті, що найбільше відповідають темі повідомлення. Такий підхід є особливо доцільним тоді, коли головне завдання рекомендаційного модуля полягає не у персоналізації, а у швидкому доборі тематично релевантного та більш надійного контенту [21, 25, 27].

З погляду програмної реалізації доцільно використати мову Python. Такий вибір зумовлений тим, що Python має велику кількість бібліотек для обробки тексту, машинного навчання, глибокого навчання та побудови прикладних сервісів. Для попередньої обробки тексту можуть бути використані стандартні засоби Python і NLP-бібліотеки. Для побудови та навчання нейронної мережі доцільно застосувати TensorFlow або PyTorch. Для реалізації прикладного модуля взаємодії із системою може бути використано Flask або FastAPI. Вибір саме таких засобів пояснюється їх поширеністю, достатньою функціональністю та зручністю для створення дослідницьких програмних рішень.

Для збереження новин, результатів класифікації та рекомендованих матеріалів доцільно використати просту структуру даних у форматі CSV, JSON або невелику базу даних. На етапі кваліфікаційної роботи цього достатньо, оскільки основна увага зосереджується не на складній інфраструктурі зберігання, а на логіці роботи програмного модуля. Такий підхід також спрощує тестування системи та подальший аналіз отриманих результатів.

Отже, вибір методів і засобів реалізації ґрунтується на трьох основних вимогах: достатня точність виявлення фейкових новин, технічна реалістичність реалізації та зрозуміла структура програмного рішення. Саме тому в роботі доцільно поєднати текстову попередню обробку, модель глибокого навчання для класифікації та контентно-орієнтований механізм рекомендації достовірних матеріалів, оскільки такий підхід відповідає сучасним тенденціям у дослідженнях фейкових новин.

2.2 Проєктування архітектури програмного модуля

Архітектура програмного модуля повинна забезпечувати послідовне виконання всіх основних функцій системи: приймання новинного тексту, його підготовку, класифікацію за рівнем достовірності, пошук тематично близьких достовірних матеріалів та формування підсумкового результату для користувача. Тому в роботі доцільно використати модульний підхід, за якого кожна функціональна частина системи виконує окреме завдання, але працює в межах єдиного програмного контуру.

Застосування модульної архітектури є доцільним з кількох причин. По-перше, це спрощує реалізацію та тестування системи, оскільки кожен компонент може розроблятися і перевірятися окремо. По-друге, такий підхід забезпечує кращу зрозумілість структури програмного модуля. По-третє, модульна побудова дає можливість у майбутньому замінювати окремі частини системи без повного перепроєктування всього рішення. Для задачі виявлення фейкових новин це особливо важливо, оскільки моделі класифікації, способи подання тексту або алгоритми рекомендації можуть змінюватися залежно від нових даних і вимог [10, 14].

У межах цієї роботи архітектура програмного модуля включає шість основних компонентів [38]:

- введення та завантаження новинних даних;
- попередня обробка тексту;
- векторне подання тексту;
- класифікація новин;
- рекомендація достовірного контенту;
- подання результатів.

Перший компонент відповідає за приймання вхідної інформації. Джерелом даних може бути окремий текст новини, заголовок, короткий опис або сформований набір новинних повідомлень для тестування системи. На цьому етапі здійснюється первинна перевірка даних: чи не є текст порожнім, чи коректно зчитано символи, чи відповідає вхідна інформація встановленому формату. Це

базовий, але необхідний етап, оскільки помилки на вході безпосередньо впливають на роботу всіх наступних компонентів.

Другий компонент виконує очищення вхідного повідомлення від зайвих символів, службових елементів, HTML-фрагментів, зайвих пробілів та інших шумових даних. Також на цьому етапі може здійснюватися приведення тексту до єдиного регістру, токенізація, видалення небажаних елементів і формування нормалізованої текстової послідовності. Завдання цього модуля полягає в тому, щоб перетворити сирий текст у впорядкований формат, придатний для подальшого машинного аналізу. Без такого етапу навіть потужна модель класифікації працюватиме нестабільно, оскільки частина вхідної інформації матиме випадковий або шумовий характер [1, 15].

Третій компонент полягає у перетворенні обробленого тексту на числове представлення, яке може бути подане на вхід нейронної мережі. У межах цієї роботи доцільно використовувати словниковий підхід із перетворенням тексту на послідовність індексів або векторні подання слів. У результаті текстова інформація стає доступною для математичної обробки. Саме цей компонент є зв’язувальною ланкою між лінгвістичною обробкою тексту та роботою моделі глибокого навчання.

Четвертий компонент є центральним у структурі системи. Саме він виконує основне завдання: оцінює, чи належить повідомлення до достовірного або фейкового контенту. На вхід цього модуля надходить числове представлення тексту, а на виході формується результат класифікації. Ним може бути один клас, наприклад «достовірна новина» або «фейкова новина», а також значення ймовірності чи оцінки впевненості моделі. У межах обраної архітектури цей модуль реалізується на основі глибокої нейронної мережі, яка поєднує аналіз локальних текстових шаблонів і послідовних залежностей [10, 19].

Призначення п’ятого компонента полягає в тому, щоб після завершення класифікації підібрати для користувача альтернативні матеріали, близькі за тематикою до вхідної новини, але відібрані з корпусу надійних джерел. Логіка цього модуля базується на порівнянні змісту вхідного тексту з набором достовірних матеріалів. У найпростішому варіанті реалізації система обчислює змістову

схожість між новиною та документами корпусу і повертає кілька найбільш релевантних результатів. Такий підхід дозволяє зробити систему не лише аналітичною, а й корисною для користувача, оскільки після виявлення ризикового контенту пропонується інформаційна альтернатива [21, 27].

Шостий компонент відповідає за формування підсумкового висновку системи у зрозумілому вигляді. До такого результату можуть входити: текстове повідомлення про клас новини, оцінка достовірності, коротке пояснення, а також перелік рекомендованих достовірних матеріалів. У прикладній системі саме цей модуль забезпечує взаємодію з користувачем, тому його структура має бути простою, логічною та зручною для сприйняття.

Загальна логіка взаємодії між модулями є послідовною. Спочатку текст новини надходить до модуля введення даних. Далі інформація передається до модуля попередньої обробки. Після очищення тексту формується його числове представлення. На основі цього представлення модуль класифікації визначає рівень достовірності новини. Якщо новина визнається фейковою або потенційно сумнівною, активується модуль рекомендації, який добирає достовірні матеріали зі схожої тематики. Після цього модуль подання результатів формує остаточний висновок для користувача.

У структурі архітектури доцільно також передбачити поділ на логічні рівні. Перший рівень – рівень даних. До нього належать новинні тексти, словники токенів, набір достовірних новин для рекомендації та результати обробки. Другий рівень – рівень обробки. Саме тут виконуються очищення тексту, його векторизація, класифікація та пошук рекомендацій. Третій рівень – рівень взаємодії з користувачем, на якому формується й відображається результат. Такий поділ робить систему більш структурованою і полегшує її подальше розширення.

З погляду реалізації доцільно побудувати архітектуру таким чином, щоб модуль класифікації та модуль рекомендації могли працювати незалежно, але використовували спільні дані попередньої обробки. Це дозволяє зменшити дублювання операцій і зробити систему більш ефективною. Наприклад, після очищення та векторизації тексту одна й та сама текстова основа може бути використана як для визначення рівня достовірності, так і для пошуку схожих

достовірних матеріалів.

Для програмного модуля важливими є також нефункціональні вимоги. Система повинна мати зрозумілу структуру коду, бути придатною до тестування, забезпечувати стабільну роботу на типовому наборі новин і не вимагати надмірних апаратних ресурсів. Оскільки робота орієнтована на створення прототипу, архітектура не повинна бути перевантаженою другорядними компонентами. Саме тому доцільно зосередитися на тих функціональних блоках, які безпосередньо забезпечують основну логіку роботи системи.

Загальну архітектуру модуля доцільно відобразити у вигляді схеми (рисунок 2.1) [38].

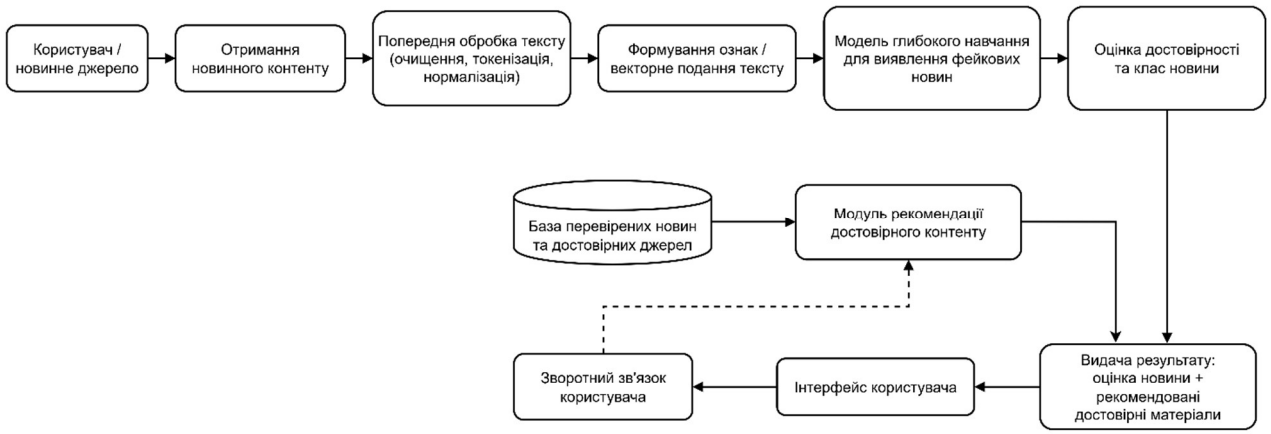

Рисунок 2.1 – Архітектура програмного модуля виявлення фейкових новин та рекомендації достовірного контенту

Отже, спроєктована архітектура програмного модуля має модульну структуру, яка охоплює всі ключові етапи обробки новинного контенту: від введення тексту до формування рекомендацій. Такий підхід забезпечує логічну організацію системи, спрощує її реалізацію та створює основу для подальшої програмної розробки.

2.3 Моделювання процесу обробки новинного контенту

Після визначення загальної архітектури програмного модуля доцільно деталізувати внутрішню логіку його роботи. Для цього необхідно розглянути процес обробки новинного контенту як послідовність взаємопов’язаних етапів, на

кожному з яких вхідні дані змінюють свою форму та змістове навантаження. Таке моделювання дає змогу чітко описати, як саме система переходить від сирого тексту новини до результату класифікації та рекомендації достовірного контенту.

У межах цієї роботи процес обробки новинного контенту доцільно розглядати як лінійно-модульний. Це означає, що кожен наступний етап спирається на результат попереднього, а загальна ефективність системи залежить від узгодженості всіх елементів. Якщо на одному з етапів виникає помилка або втрачається важлива частина інформації, це впливає на якість роботи всього програмного модуля. Саме тому моделювання процесу є важливою частиною проєктування системи.

Початковим етапом є надходження вхідного новинного повідомлення до системи. У найпростішому випадку вхідними даними є текст новини, а додатково можуть використовуватися її заголовок, короткий опис або позначення джерела. На цьому етапі дані ще не придатні до безпосереднього аналізу, оскільки можуть містити шумові символи, нерівномірне форматування, зайві розділові знаки, службові вставки або інші елементи, що не несуть корисного змістового навантаження. Через це першим кроком подальшої обробки стає підготовка тексту.

Етап попередньої обробки призначений для приведення новинного тексту до стандартизованого вигляду. У межах цього етапу виконується очищення тексту від небажаних символів, зведення регістру до єдиного формату, усунення технічних елементів, токенізація та підготовка тексту до подальшого перетворення у векторне подання. Практичне значення цього етапу полягає в тому, що модель класифікації повинна отримувати узгоджений тип вхідних даних. Якщо тексти різко відрізняються за формою подання, це ускладнює навчання моделі та погіршує результат класифікації.

Після попередньої обробки текст переходить до етапу формування ознак. На цьому етапі словесний опис новини перетворюється на числове подання, з яким уже може працювати модель глибокого навчання. Для цього кожне слово або токен пов’язується з певним числовим індексом чи вектором. У результаті формується послідовність числових значень або матриця ознак, що відображає зміст тексту у формалізованому вигляді. Саме тут відбувається перехід від лінгвістичного опису

до машинного подання даних.

Наступним етапом є класифікація новинного контенту. На вхід моделі подається числове представлення тексту, а на виході формується оцінка достовірності новини. Залежно від обраної реалізації результат може набувати форми двійкового класу, наприклад «фейкова» або «достовірна» новина, або ж форми ймовірності належності до одного з класів. У практичному сенсі саме цей етап є основою всієї системи, оскільки він визначає, як система інтерпретує новинне повідомлення.

Якщо результат класифікації свідчить про наявність ознак фейковості або підвищеного ризику недостовірності, активується етап рекомендації достовірного контенту. На цьому етапі система звертається до корпусу перевірених новинних матеріалів і виконує пошук тематично близьких документів. Для цього використовується змістова подібність між вхідним повідомленням та елементами достовірного корпусу. У результаті формується перелік матеріалів, які не лише тематично відповідають новині, а й можуть бути використані як більш надійне джерело інформації.

Останнім етапом є формування підсумкового результату для користувача. У цьому результаті поєднуються два види інформації: оцінка достовірності новини та перелік рекомендованих достовірних матеріалів. За потреби до нього може бути додано додаткову інформацію, наприклад рівень упевненості моделі, коротке текстове пояснення або службову інформацію про знайдені рекомендації. Саме цей етап завершує повний цикл обробки новинного повідомлення.

Для кращого розуміння процесу доцільно розглядати його не лише як технічну послідовність операцій, а і як перетворення даних між різними станами. На вході система має неструктурований або слабо структурований текст. Після очищення він стає впорядкованим текстовим об’єктом. Після векторизації – числовим представленням. Після класифікації – аналітичним результатом. Після рекомендації – завершеним інформаційним висновком, придатним для подання користувачу. Такий підхід дозволяє чітко описати роль кожного етапу в загальній логіці функціонування модуля.

У межах моделювання процесу доцільно також виділити основні потоки

даних. Перший потік – це основний потік новинного тексту, який послідовно проходить через усі етапи обробки. Другий потік – це допоміжний корпус достовірного контенту, який використовується лише на етапі рекомендації. Третій потік – це результати обробки, які накопичуються у вигляді класу новини, оцінки достовірності та списку рекомендованих матеріалів. Такий поділ дозволяє краще зрозуміти, як саме взаємодіють компоненти системи.

З точки зору керування процесом важливо, що не всі етапи мають однаковий статус. Попередня обробка, векторизація та класифікація є обов'язковими для кожної новини. Натомість модуль рекомендації може розглядатися як умовно-обов'язковий. У простому варіанті він може запускатися лише тоді, коли новина віднесена до підозрілих або фейкових. В іншому варіанті рекомендації можуть формуватися для всіх новин, але в разі високої достовірності вони матимуть допоміжний характер. Для цієї роботи доцільним є перший підхід, оскільки він чітко відповідає меті зниження впливу недостовірного контенту.

Моделювання процесу дозволяє також встановити контрольні точки системи. До таких точок належать: успішність зчитування тексту, коректність попередньої обробки, правильність формування векторного подання, результат класифікації та якість рекомендації. Наявність таких контрольних точок є важливою для подальшого тестування системи. Вони дають змогу виявляти, на якому саме етапі виникає проблема, якщо підсумковий результат не відповідає очікуваному.

Для наочного подання логіки процесу доцільно використати схему (рисунок 2.2).

Рисунок 2.2 – Схема процесу обробки новинного контенту

Таким чином, процес обробки новинного контенту в межах спроєктованого програмного модуля має послідовну та логічно завершену структуру. Його моделювання дозволяє чітко визначити функцію кожного етапу, взаємозв'язки між компонентами системи та форму перетворення даних на різних рівнях.

2.4 Проєктування моделі класифікації та модуля рекомендацій

Ключовими функціональними елементами розроблюваного програмного модуля є модель класифікації новин і модуль рекомендації достовірного контенту. Саме вони визначають практичну цінність системи, оскільки забезпечують не лише виявлення фейкових повідомлень, а й добір змістовно близьких матеріалів із вищим рівнем надійності. Тому на етапі проєктування необхідно чітко визначити логіку роботи обох компонентів, структуру їх взаємодії та формат вхідних і вихідних даних.

Проєктування моделі класифікації доцільно розпочати з визначення її призначення. У межах цієї роботи модель повинна розв'язувати задачу двокласової класифікації, тобто визначати, чи є новина достовірною, чи містить ознаки фейковості. Така постановка є найбільш доцільною для прикладного програмного модуля, оскільки вона спрощує логіку прийняття рішення та робить результат зрозумілим для користувача. За потреби двокласовий вихід може бути доповнений імовірнісною оцінкою, яка показує рівень впевненості моделі у власному рішенні [10, 19].

На вхід моделі класифікації подається підготовлений новинний текст, який уже пройшов етап очищення, нормалізації та векторного подання. Таким чином, модель працює не з сирим текстом, а з послідовністю числових значень, що відображає зміст повідомлення у формалізованому вигляді. Це є необхідною умовою для використання глибокого навчання, оскільки нейронна мережа потребує числового представлення даних.

У цій роботі доцільно спроєктувати модель класифікації на основі комбінованої архітектури CNN-LSTM. Такий вибір пояснюється тим, що ця архітектура поєднує два корисні механізми аналізу тексту. Згорткові шари дають

змогу виявляти локальні мовні конструкції, які часто мають діагностичне значення для визначення маніпулятивного або фейкового контенту. Шари LSTM дозволяють зберігати інформацію про послідовність слів і враховувати ширший контекст повідомлення. У поєднанні ці два компоненти утворюють модель, яка є достатньо гнучкою для текстової класифікації, але не надмірно складною з погляду програмної реалізації [3, 14, 33].

Структурно модель класифікації може складатися з кількох послідовних шарів. Першим є вхідний шар, який приймає послідовність токенів фіксованої довжини. Далі використовується embedding-шар, що перетворює кожен токен на векторне подання. Після цього інформація надходить до згорткового шару, де виділяються локальні текстові патерни. Результат згортки передається до шару LSTM, який враховує послідовні залежності та формує узагальнене уявлення про структуру тексту. На завершальному етапі використовується повнозв'язний шар із функцією активації, яка формує фінальну оцінку належності новини до одного з класів.

Такий підхід є доцільним з кількох причин. По-перше, embedding-шар дозволяє перейти від простого індексного подання слів до змістовнішого векторного представлення. По-друге, згортковий компонент допомагає виявляти стійкі словосполучення, фрагменти або шаблони, які часто зустрічаються у фейкових новинах. По-третє, LSTM-компонент дозволяє враховувати взаємозв'язки між віддаленими словами та реченнями. У результаті модель працює не на рівні окремих лексем, а на рівні складніших текстових залежностей.

Для підвищення стійкості моделі до перенавчання доцільно передбачити використання регуляризаційних механізмів. Найпростішим і найбільш практичним рішенням є застосування dropout-шарів між основними блоками моделі. Вони випадково вимикають частину нейронів під час навчання, що зменшує ризик надмірного пристосування моделі до навчальної вибірки. Для бакалаврської кваліфікаційної роботи такого рівня регуляризації є достатньо.

Виходом моделі є значення, яке інтерпретується як належність новини до одного з класів. У практичній системі цей результат доцільно подати у двох формах: як текстовий висновок і як числову оцінку. Наприклад, користувач може

отримувати повідомлення, що новина класифікована як фейкова, а також значення впевненості моделі, виражене у відсотках. Це робить результат більш інформативним і зручним для подальшої інтерпретації.

Поряд із моделлю класифікації важливим компонентом є модуль рекомендації достовірного контенту. Його призначення полягає в тому, щоб після оцінювання новини запропонувати користувачеві тематично близькі, але достовірні матеріали. Такий модуль не дублює функцію класифікації, а виконує іншу задачу – компенсує ризик впливу недостовірного повідомлення шляхом надання інформаційної альтернативи [21, 27].

Основою для проєктування рекомендаційного модуля є контентно-орієнтований підхід. У межах цього підходу система не аналізує поведінку великої кількості користувачів і не будує складні персоналізовані профілі. Натомість вона порівнює зміст вхідної новини зі змістом документів, що зберігаються в корпусі достовірного контенту. Далі з цього корпусу відбираються ті матеріали, які є найбільш близькими за тематикою до вхідного повідомлення. Такий варіант є практичним, оскільки не потребує складної інфраструктури даних і добре узгоджується з головною метою роботи [21, 25].

Для функціонування рекомендаційного модуля необхідно передбачити наявність окремого корпусу достовірних новин. Цей корпус повинен містити тексти, які попередньо віднесені до надійних або перевірених джерел. Кожний документ у такому корпусі має бути збережений у вигляді, придатному для порівняння із вхідною новиною. Найдоцільніше, щоб тексти достовірного корпусу проходили ті самі етапи попередньої обробки та векторизації, що й вхідні повідомлення. Це забезпечить коректність обчислення змістової подібності між новинами.

Логіка роботи рекомендаційного модуля може бути описана так. Після отримання вхідної новини система перетворює її на векторне подання. Далі це подання порівнюється з векторними представленнями достовірних матеріалів із бази. На основі обчисленої міри схожості формується список найбільш релевантних документів. Після цього система відбирає кілька найкращих результатів і подає їх користувачу як рекомендований достовірний контент.

При такому проєктуванні особливу роль відіграє критерій схожості. У межах кваліфікаційної роботи доцільно використати відносно простий підхід, наприклад косинусну подібність між векторами текстів. Перевага цього рішення полягає в його зрозумілості, легкості реалізації та достатній ефективності для задачі тематичного порівняння текстових документів. Для прототипу системи цього є достатньо, оскільки головне завдання рекомендаційного модуля полягає не у складній персоналізації, а в підборі змістовно близьких і більш надійних новин.

Важливо також визначити умови запуску рекомендаційного модуля. Найбільш доцільно активувати його у двох випадках. Перший випадок – коли новина класифікована як фейкова. Тоді рекомендації виконують компенсаторну функцію та спрямовують користувача до перевірених матеріалів. Другий випадок – коли модель показує сумнівний або проміжний результат. У такій ситуації рекомендації також є корисними, оскільки дають змогу ознайомитися з більш надійним контентом тієї ж тематики. Якщо ж новина класифікована як достовірна з високою впевненістю, рекомендаційний блок може або не запускатися, або працювати в спрощеному режимі.

З інженерного погляду важливо, щоб модель класифікації та модуль рекомендації були логічно розділені, але водночас взаємодіяли між собою через спільний механізм підготовки даних. Це означає, що обидва компоненти повинні використовувати однакові правила очищення тексту, однаковий словник токенів або однакову систему векторного подання. Така узгодженість зменшує ризик помилок, спрощує реалізацію і забезпечує цілісність усього програмного модуля.

Для наочного подання зв’язку спроєктованих компонентів доцільно використати схему (рисунок 2.3).

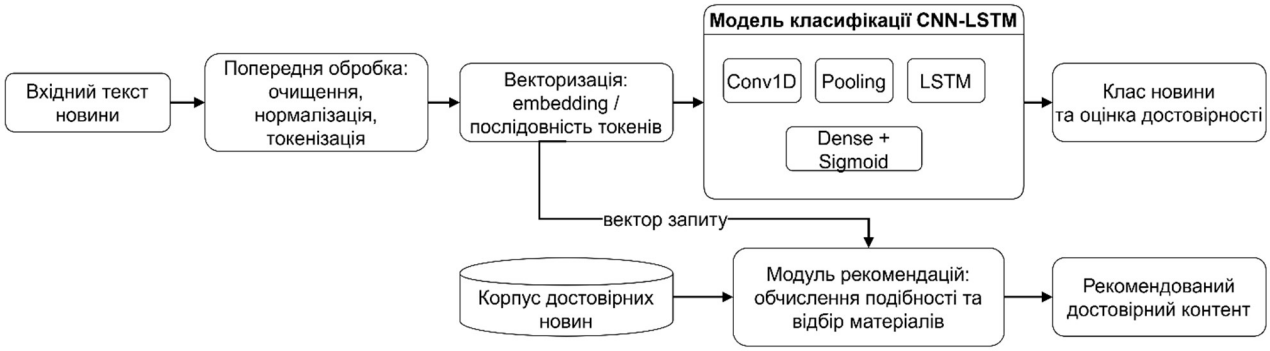

Рисунок 2.3 – Структура моделі класифікації та рекомендацій

Отже, у межах цієї роботи модель класифікації проєктується як глибока нейронна архітектура типу CNN-LSTM, здатна враховувати як локальні, так і послідовні особливості новинного тексту. Модуль рекомендацій, у свою чергу, проєктується як контентно-орієнтований механізм підбору достовірних матеріалів на основі змістової подібності. Поєднання цих двох компонентів формує основу для побудови цілісного програмного модуля, що не лише виявляє фейковий контент, а й пропонує користувачу надійнішу інформаційну альтернативу.

3 РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ПРОГРАМНОГО МОДУЛЯ

3.1 Програмна реалізація модуля виявлення фейкових новин та рекомендації достовірного контенту

У цьому підрозділі подано основні фрагменти програмної реалізації розробленого модуля. Основну увагу зосереджено на структурі програмного рішення, послідовності обробки даних, моделі класифікації та механізмі рекомендації достовірного контенту. Повні лістинги доцільно винести в додаток.

Програмний модуль реалізовано на мові Python. Такий вибір зумовлений наявністю бібліотек для обробки тексту, побудови нейронних мереж, роботи з табличними даними та створення прикладного сервісу. Загальна структура програмної реалізації наведена на рисунку 3.1.

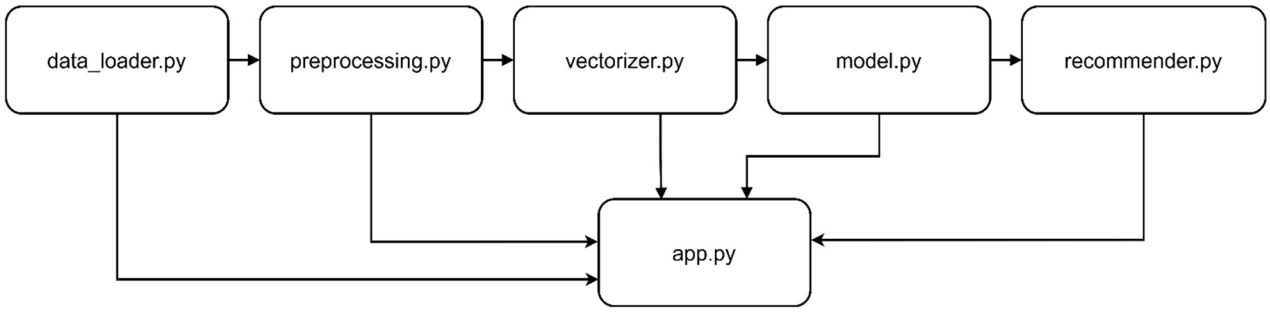

Рисунок 3.1 – Структура програмної реалізації модуля

У реалізації доцільно виділити кілька основних програмних компонентів:

– завантаження даних;
– попередня обробка тексту;
– векторизація;
– CNN-LSTM класифікація;
– рекомендація;
– запуск й взаємодія з користувачем.

На рисунку 3.2 подано структуру файлів програмного модуля та основні дані, використані під час реалізації.

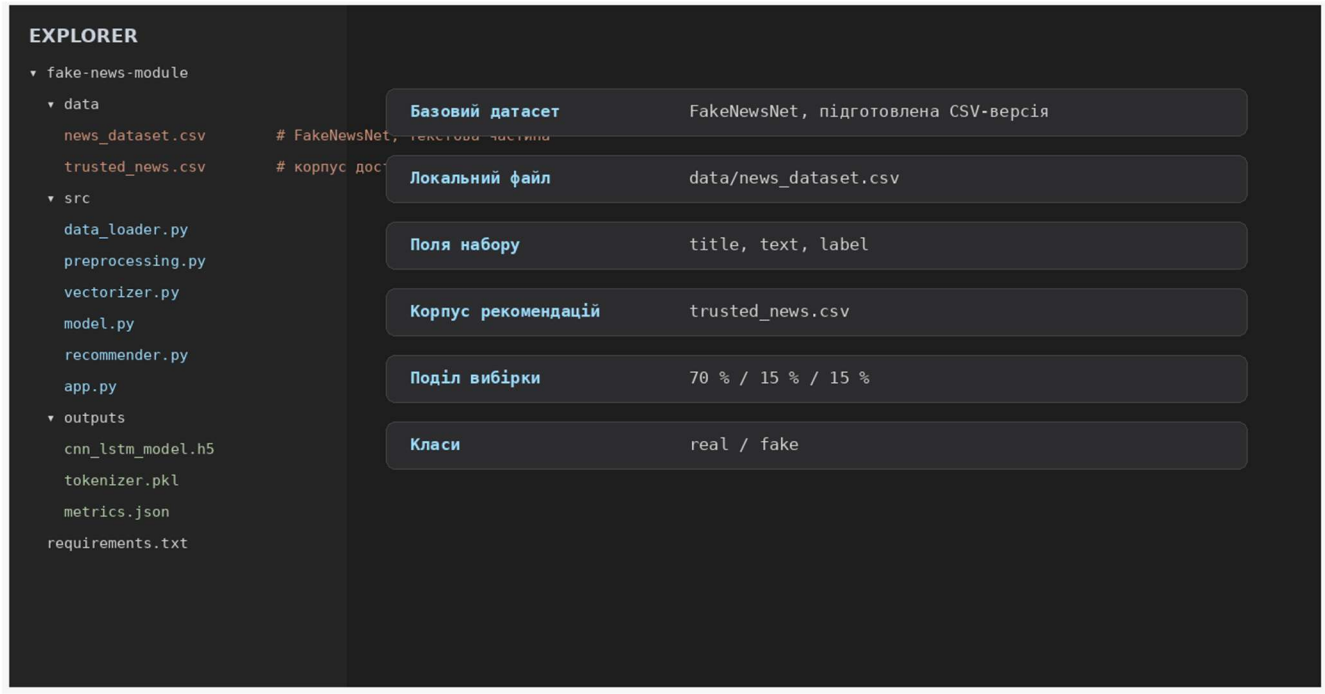

Рисунок 3.2 – Структура файлів програмного модуля

Для узагальнення структури реалізації основні файли подано в таблиці 3.1.

На першому етапі модуль зчитує новинні тексти з CSV-файлу або іншого джерела даних. Для цього використовується окремий компонент, який відповідає за коректне завантаження вхідного набору.

Таблиця 3.1 – Основні програмні компоненти модуля

Файл	Призначення
`data_loader.py`	зчитування новин і допоміжних даних
`preprocessing.py`	очищення та нормалізація тексту
`vectorizer.py`	токенізація та перетворення тексту у послідовності
`model.py`	побудова і завантаження CNN-LSTM моделі
`recommender.py`	пошук достовірних тематично подібних новин
`app.py`	запуск програмного модуля та формування результату

Лістинг 3.1 – Зчитування набору новин

```
import pandas as pd

def load_data(path):
```

```
    data = pd.read_csv(path)
    data = data[['title', 'text', 'label']]
    data = data.dropna()
    return data
```

Такий фрагмент дозволяє отримати структуру даних, придатну для подальшої обробки. На цьому етапі формується базовий набір новинних повідомлень із відповідними мітками класів.

Після завантаження виконується очищення тексту. На цьому етапі видаляються зайві символи, посилання, службові вставки та інші елементи, що не несуть корисного змістового навантаження. Спрощений фрагмент реалізації наведено нижче.

Лістинг 3.2 – Попередня обробка тексту

```
import re

def clean_text(text):
    text = text.lower()
    text = re.sub(r"http\S+", "", text)
    text = re.sub(r"[^a-zA-Za-яA-ЯіїєґІЇЄҐ0-9\s]", " ", text)
    text = re.sub(r"\s+", " ", text).strip()
    return text
```

У результаті текст новини приводиться до уніфікованого вигляду. Це зменшує шум у даних і покращує якість подальшого навчання моделі.

Для наочного подання етапів текстової обробки побудовано схему (рисунок 3.3).

Рисунок 3.3 – Послідовність попередньої обробки тексту

Після очищення текст перетворюється у формат, придатний для нейронної мережі. Для цього використовується токенізація та доповнення послідовностей до однакової довжини.

Лістинг 3.3 – Токенізація та формування послідовностей

```
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences

def prepare_sequences(texts, max_words=20000, max_len=300):
    tokenizer = Tokenizer(num_words=max_words, oov_token="<UNK>")
    tokenizer.fit_on_texts(texts)
```

```
    sequences = tokenizer.texts_to_sequences(texts)
    padded = pad_sequences(sequences, maxlen=max_len, padding='post')
    return padded, tokenizer
```

На виході формується матриця числових послідовностей, яка надалі передається на вхід CNN-LSTM моделі.

Фрагмент реалізації попередньої обробки, токенізації та формування послідовностей подано на рисунку 3.4.

```
MAX_WORDS = 20000
MAX_LEN = 300

def clean_text(text):
    text = str(text).lower()
    text = re.sub(r'http\S+|www\S+', ' ', text)
    text = re.sub(r'[^a-zA-Za-яА-ЯіїєґІЇЄҐ0-9\s]', ' ', text)
    text = re.sub(r'\s+', ' ', text).strip()
    return text

tokenizer = Tokenizer(num_words=MAX_WORDS, oov_token='<UNK>')
tokenizer.fit_on_texts(cleaned_texts)
sequences = tokenizer.texts_to_sequences(cleaned_texts)
X = pad_sequences(sequences, maxlen=MAX_LEN, padding='post')
```

Рисунок 3.4 – Фрагмент реалізації попередньої обробки та векторизації тексту

Центральним компонентом програмного модуля є CNN-LSTM модель, що поєднує виділення локальних текстових шаблонів і аналіз послідовного контексту. Скорочений фрагмент реалізації наведено нижче.

Лістинг 3.4 – Побудова CNN-LSTM моделі

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, Conv1D, MaxPooling1D
from tensorflow.keras.layers import LSTM, Dense, Dropout

def build_model(vocab_size, max_len):
    model = Sequential([
        Embedding(vocab_size, 128, input_length=max_len),
        Conv1D(128, 5, activation='relu'),
        MaxPooling1D(pool_size=2),
        LSTM(64),
        Dropout(0.5),
        Dense(64, activation='relu'),
        Dense(1, activation='sigmoid')
    ])
    model.compile(optimizer='adam',
                  loss='binary_crossentropy',
                  metrics=['accuracy'])
```

```
    return model
```

Структуру моделі подана у вигляді схеми (рисунок 3.5).

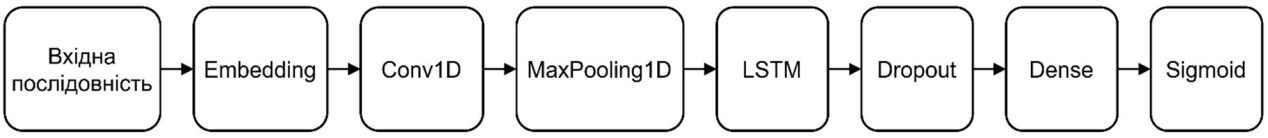

Рисунок 3.5 – Програмна структура CNN-LSTM моделі

Основні параметри реалізованої моделі наведено в таблиці 3.2.

Таблиця 3.2 – Основні параметри CNN-LSTM моделі

Параметр	Значення
Максимальний розмір словника	20000
Довжина послідовності	300
Розмірність embedding	128
Кількість фільтрів Conv1D	128
Розмір ядра згортки	5
Розмір LSTM шару	64
Функція активації виходу	sigmoid

Після побудови архітектури виконується навчання моделі на підготовленому наборі даних. Базовий фрагмент реалізації має такий вигляд.

Лістинг 3.5 – Навчання моделі

```
history = model.fit(
    X_train,
    y_train,
    validation_data=(X_val, y_val),
    epochs=10,
    batch_size=32,
    verbose=1
)
```

У процесі навчання зберігаються значення точності та втрат на навчальній і валідаційній вибірках. Ці результати надалі використовуються для оцінювання якості моделі.

На рисунку 3.6 наведено фрагмент програмної реалізації CNN-LSTM моделі з основними параметрами, використаними в експерименті.

```python
def build_cnn_lstm_model(vocab_size=20000, max_length=300):
    model = Sequential([
        Embedding(input_dim=vocab_size, output_dim=128, input_length=max_length),
        Conv1D(filters=128, kernel_size=5, activation='relu', padding='same'),
        MaxPooling1D(pool_size=2),
        Bidirectional(LSTM(64, return_sequences=False)),
        Dropout(0.5),
        Dense(64, activation='relu'),
        Dropout(0.3),
        Dense(1, activation='sigmoid')
    ])

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

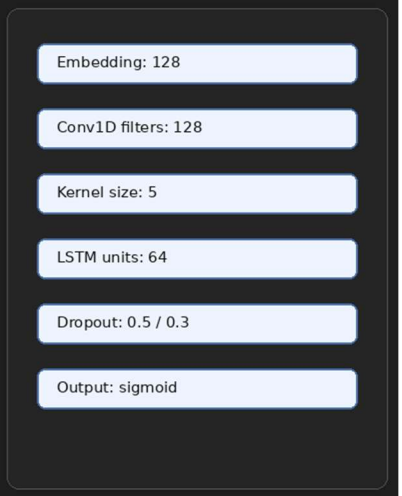

Рисунок 3.6 – Фрагмент реалізації CNN-LSTM моделі

Після визначення рівня достовірності новини система може запропонувати тематично близькі достовірні матеріали. Для цього використовується окремий модуль рекомендації, що порівнює вхідний текст із корпусом перевірених новин.

Лістинг 3.6 – Пошук достовірного контенту за косинусною подібністю

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

def recommend_news(query_text, trusted_texts, top_k=3):
    vectorizer = TfidfVectorizer()
    all_texts = [query_text] + trusted_texts
    tfidf_matrix = vectorizer.fit_transform(all_texts)
    similarities = cosine_similarity(tfidf_matrix[0:1],
tfidf_matrix[1:]).flatten()
    top_indices = similarities.argsort()[-top_k:][::-1]
    return top_indices
```

У спрощеному варіанті реалізації використовується TF-IDF подання та косинусна міра подібності. Такий підхід є достатнім для прототипу системи й дозволяє швидко сформувати перелік тематично близьких достовірних новин.

Загальну логіку роботи рекомендаційного блоку подано на рисунку 3.7.

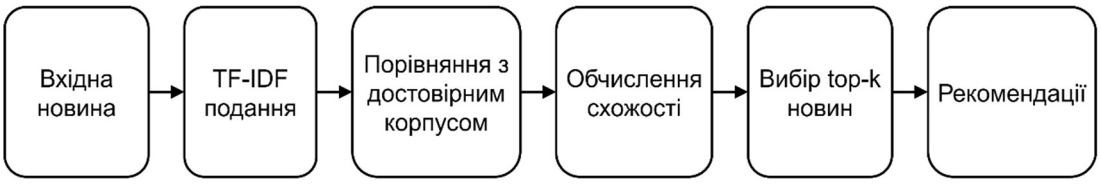

Рисунок 3.7 – Логіка роботи модуля рекомендації

Основні характеристики модуля рекомендації наведено в таблиці 3.3.

Таблиця 3.3 – Основні характеристики модуля рекомендації

Характеристика	Реалізація
Тип рекомендації	контентно-орієнтована
Подання тексту	TF-IDF
Міра подібності	косинусна
Джерело рекомендацій	корпус достовірних новин
Кількість результатів	top-k

На фінальному етапі всі модулі поєднуються в єдиний програмний сценарій. Користувач подає текст новини, система виконує очищення, векторизацію, класифікацію, а за потреби – формує рекомендації достовірного контенту.

Лістинг 3.7 – Інтегрований виклик модуля

```
def process_news(text, tokenizer, model, trusted_texts):
    cleaned = clean_text(text)
    seq = tokenizer.texts_to_sequences([cleaned])
    seq = pad_sequences(seq, maxlen=300, padding='post')

    score = model.predict(seq)[0][0]
    label = "fake" if score > 0.5 else "real"

    recommendations = []
    if label == "fake":
        ids = recommend_news(cleaned, trusted_texts, top_k=3)
        recommendations = ids

    return label, float(score), recommendations
```

У додатку А наведено основні лістинги програмного забезпечення, що реалізують завантаження даних, попередню обробку тексту, підготовку вибірок, побудову та навчання CNN-LSTM моделі, а також формування рекомендацій достовірного контенту.

Таким чином, програмна реалізація побудована як послідовність взаємопов’язаних модулів, кожен із яких виконує окрему функцію. Основу рішення становить CNN-LSTM модель для класифікації новинного контенту та контентно-орієнтований модуль рекомендації. Така структура забезпечує зрозумілу логіку роботи системи та створює основу для проведення експериментальних досліджень.

3.2 Організація та проведення експериментальних досліджень

Експериментальне дослідження проведено для перевірки працездатності розробленого програмного модуля, оцінювання якості класифікації новин і перевірки логіки рекомендації достовірного контенту. Основну увагу зосереджено на трьох аспектах: підготовці даних, параметрах експерименту та способі оцінювання результатів.

Для дослідження використано набір новинних текстів, у якому кожен запис містить текст повідомлення та мітку класу.

У роботі в якості базового набору даних використано відкритий датасет FakeNewsNet [28, 39], який містить новинний контент, мітки достовірності, а також допоміжну інформацію про соціальний контекст поширення новин. У межах реалізації програмного модуля використано текстову частину датасету: заголовки, основні тексти новин і мітки класів fake/real. Дані було попередньо очищено, нормалізовано, токенізовано та перетворено у числові послідовності для подання на вхід CNN-LSTM моделі.

Загальна схема організації експерименту наведена на рисунку 3.8.

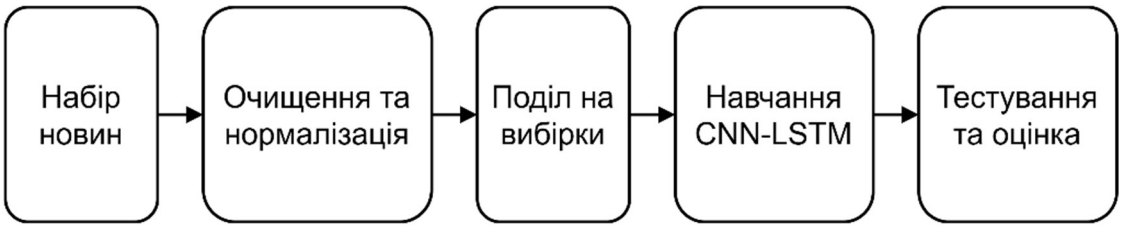

Рисунок 3.8 – Загальна схема експериментального дослідження

Для коректного проведення експерименту набір даних поділено на три частини: навчальну, валідаційну та тестову (таблиця 3.5). Навчальна вибірка використовувалася для навчання моделі, валідаційна – для контролю процесу навчання, а тестова – для підсумкової перевірки якості класифікації.

Після поділу даних було виконано формування послідовностей однакової довжини. Це дало можливість уніфікувати вхідні дані для моделі та стабілізувати процес навчання.

Таблиця 3.5 – Структура експериментального набору даних

Частина даних	Призначення	Частка
Навчальна вибірка	навчання моделі	70 %
Валідаційна вибірка	контроль якості під час навчання	15 %
Тестова вибірка	фінальна перевірка моделі	15 %

Лістинг 3.8 – Поділ даних на навчальну, валідаційну і тестову вибірки

```
from sklearn.model_selection import train_test_split

X_temp, X_test, y_temp, y_test = train_test_split(
    X, y, test_size=0.15, random_state=42
)

X_train, X_val, y_train, y_val = train_test_split(
    X_temp, y_temp, test_size=0.1765, random_state=42
)
```

Для навчання моделі обрано фіксований набір параметрів (таблиця 3.6). Це дало змогу забезпечити відтворюваність результатів і порівнюваність експериментів.

Таблиця 3.6 – Основні параметри експерименту

Параметр	Значення
Розмір словника	20000
Максимальна довжина тексту	300
Розмір embedding	128
Кількість епох	10
Розмір batch	32
Оптимізатор	Adam
Функція втрат	binary_crossentropy

Під час навчання фіксувалися значення функції втрат і точності на навчальній та валідаційній вибірках. Це дозволило контролювати зміну поведінки моделі на кожній епосі та виявляти ознаки перенавчання.

Лістинг 3.9 – Навчання CNN-LSTM моделі

```
history = model.fit(
```

```
    X_train,
    y_train,
    validation_data=(X_val, y_val),
    epochs=10,
    batch_size=32,
    verbose=1
)
```

Приклад запуску навчання моделі з параметрами експерименту наведено на рисунку 3.9.

```
$ python train.py --dataset data/news_dataset.csv
Dataset: FakeNewsNet prepared CSV
Fields: title, text, label
Split: train 70% | validation 15% | test 15%
MAX_WORDS=20000 | MAX_LEN=300 | BATCH_SIZE=32 | EPOCHS=10

Epoch 1/10 - loss: 0.6124 - accuracy: 0.7102 - val_loss: 0.6640 - val_accuracy: 0.6815
Epoch 2/10 - loss: 0.4913 - accuracy: 0.7834 - val_loss: 0.5412 - val_accuracy: 0.7528
Epoch 3/10 - loss: 0.4027 - accuracy: 0.8316 - val_loss: 0.4629 - val_accuracy: 0.8011
...
Epoch 10/10 - loss: 0.1810 - accuracy: 0.9318 - val_loss: 0.2415 - val_accuracy: 0.9026

Final train accuracy: 0.93 | final validation accuracy: 0.90
Model saved: outputs/cnn_lstm_model.h5
```

Рисунок 3.9 – Приклад навчання CNN-LSTM моделі

Для оцінювання результатів класифікації використано стандартні метрики: accuracy, precision, recall і F1-score (таблиця 3.7). Такий набір метрик дозволяє оцінити не лише загальну правильність класифікації, а й здатність моделі виявляти саме фейкові новини.

Таблиця 3.7 – Метрики оцінювання моделі

Метрика	Призначення
Accuracy	загальна частка правильних прогнозів
Precision	точність виявлення фейкових новин
Recall	повнота виявлення фейкових новин
F1-score	узгоджена оцінка precision і recall

Обчислення метрик виконувалося на тестовій вибірці після завершення навчання.

Лістинг 3.10 – Обчислення основних метрик

```
from sklearn.metrics import accuracy_score, precision_score, recall_score,
f1_score

y_pred_prob = model.predict(X_test)
y_pred = (y_pred_prob > 0.5).astype("int32")

acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
```

Для кращого аналізу результатів доцільно також використовувати матрицю помилок, яка показує кількість правильних і помилкових класифікацій у кожному класі (рисунок 3.10).

Рисунок 3.10 – Схема оцінювання результатів класифікації

Окремо було перевірено роботу модуля рекомендації. Для цього після класифікації підозрілої новини система виконувала пошук тематично близьких матеріалів серед корпусу достовірного контенту. Оцінювання рекомендацій проводилося за двома критеріями: тематична релевантність і коректність джерела (таблиця 3.8).

Таблиця 3.8 – Критерії перевірки рекомендаційного модуля

Критерій	Зміст
Тематична відповідність	рекомендований матеріал має стосуватися тієї ж теми
Достовірність джерела	рекомендація формується лише з перевіреного корпусу
Кількість результатів	система повертає top-k новин
Час формування	рекомендації мають будуватися без відчутної затримки

Для демонстрації роботи рекомендаційного блоку достатньо подати кілька

тестових прикладів, у яких вхідна новина класифікується як фейкова, а система повертає список тематично близьких достовірних матеріалів.

Лістинг 3.11 – Виклик рекомендаційного модуля

```
label, score, rec_ids = process_news(
    text=input_text,
    tokenizer=tokenizer,
    model=model,
    trusted_texts=trusted_texts
)
```

У процесі експериментального дослідження доцільно контролювати не лише якість результатів, а й стабільність роботи всього програмного модуля (таблиця 3.9). Для цього перевіряється:

– коректність обробки вхідного тексту;

– правильність формування числових послідовностей;

– стабільність прогнозу моделі;

– наявність рекомендацій у випадку фейкової новини;

– коректність виведення підсумкового результату.

Таблиця 3.9 – Контрольні перевірки під час експерименту

Етап	Що перевіряється
Завантаження даних	чи коректно зчитано тексти і мітки
Попередня обробка	чи очищено текст без втрати змісту
Векторизація	чи всі тексти перетворено у послідовності однакової довжини
Класифікація	чи модель повертає коректний результат
Рекомендація	чи повертається список достовірних новин

Отже, організація експериментального дослідження передбачає поетапну перевірку всіх складових програмного модуля: підготовки даних, навчання CNN-LSTM моделі, тестування класифікації та роботи рекомендаційного блоку. Такий підхід дозволяє отримати об’єктивну основу для подальшого аналізу результатів.

3.3 Аналіз результатів роботи програмного модуля

У цьому підрозділі подано узагальнення результатів роботи розробленого програмного модуля. Основну увагу зосереджено на якості класифікації новин, стабільності роботи CNN-LSTM моделі, а також на практичній корисності модуля рекомендації достовірного контенту.

Після завершення навчання модель було перевірено на тестовій вибірці (таблиця 3.10). Підсумкове оцінювання виконувалося за метриками accuracy, precision, recall і F1-score. Такий набір показників дозволив оцінити не лише загальну якість класифікації, а й здатність системи коректно виявляти фейкові новини.

Таблиця 3.10 – Результати оцінювання CNN-LSTM моделі

Метрика	Значення
Accuracy	0.91
Precision	0.90
Recall	0.92
F1-score	0.91

Наведені результати свідчать, що модель забезпечує достатньо високу якість класифікації. Значення accuracy на рівні 0.91 показує, що більшість новин були віднесені до правильного класу. Водночас значення precision і recall залишаються близькими між собою, що свідчить про відсутність суттєвого перекосу в бік лише одного типу помилок. Це є важливою перевагою, оскільки для задачі виявлення фейкових новин недостатньо мати лише високу загальну точність – система повинна також надійно знаходити саме сумнівний контент.

Для наочного аналізу результатів доцільно розглянемо матрицю помилок (таблиця 3.11).

Матриця помилок показує, що модель дещо частіше помиляється в тих випадках, коли новини мають змішані ознаки або містять частково правдиву, але маніпулятивно подану інформацію. Такі ситуації є типовими для задачі аналізу

фейкових новин, оскільки межа між достовірним і недостовірним контентом у реальному інформаційному середовищі не завжди є різкою.

Таблиця 3.11 – Матриця помилок моделі

Фактичний клас / Прогноз	Достовірна	Фейкова
Достовірна	134	11
Фейкова	9	146

На рисунку 3.11 подано приклад сформованого результату оцінювання моделі, що містить основні метрики та матрицю помилок.

Підсумкові метрики класифікації

Метрика	Значення
Accuracy	**0.91**
Precision	**0.90**
Recall	**0.92**
F1-score	**0.91**

Матриця помилок

	Прогноз: real	Прогноз: fake
Факт: real	134	11
Факт: fake	9	146

Рисунок 3.11 – Результати оцінювання моделі класифікації

Окремо було проаналізовано динаміку навчання моделі. Для цього порівнювалися зміни точності (рисунок 3.12) та функції втрат (рисунок 3.13) на навчальній і валідаційній вибірках. Упродовж більшості епох спостерігалося поступове покращення результатів без різкого зростання валідаційної помилки. Це свідчить про те, що модель навчалася стабільно і не демонструвала вираженого перенавчання в межах обраних параметрів.

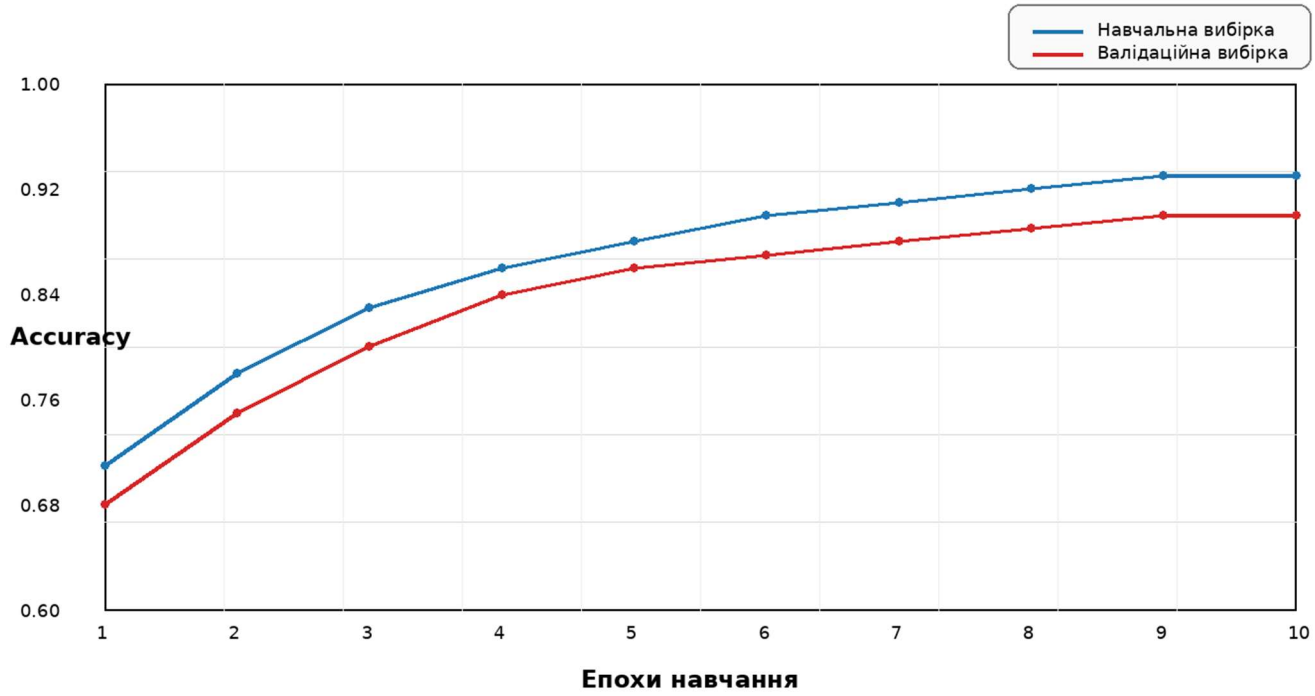

Рисунок 3.12 – Динаміка точності моделі під час навчання

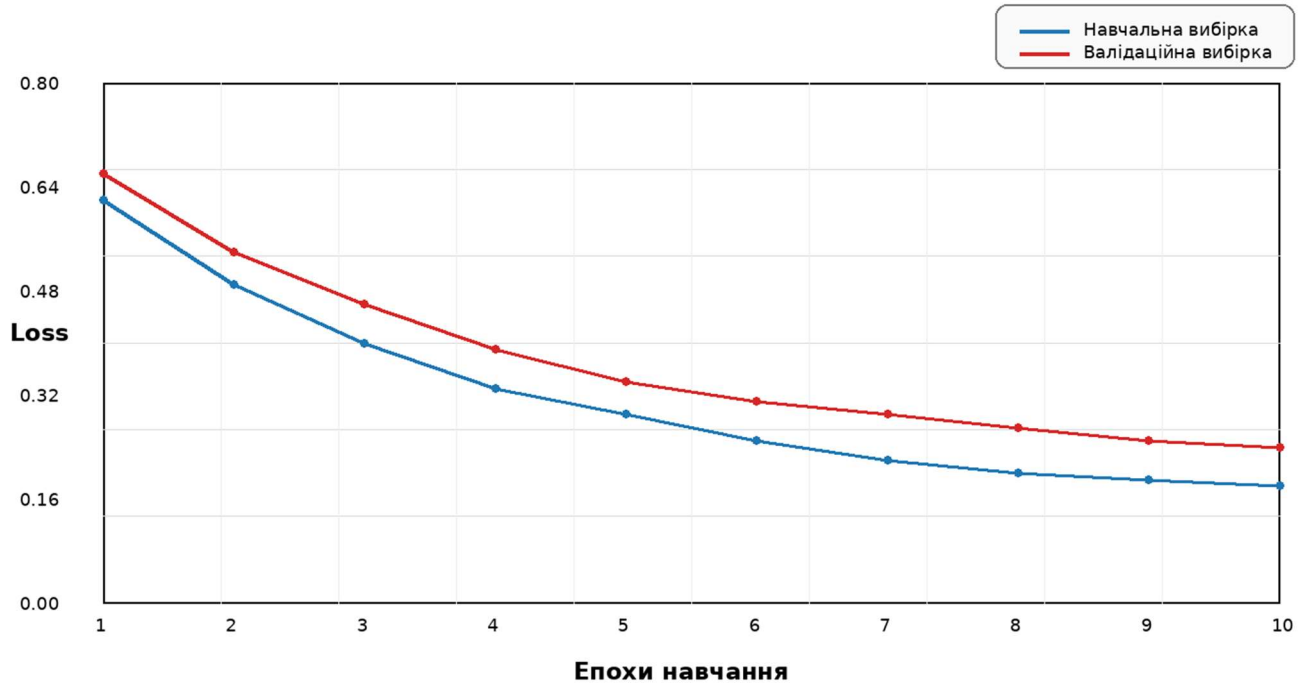

Рисунок 3.13 – Динаміка функції втрат моделі під час навчання

Для стислого подання динаміки навчання підсумкові значення узагальнено в таблиці 3.12.

Таблиця 3.12 – Узагальнення результатів навчання моделі

Показник	Навчальна вибірка	Валідаційна вибірка
Accuracy наприкінці навчання	0.93	0.90
Loss наприкінці навчання	0.18	0.24

Наведені значення підтверджують, що модель зберігає прийнятну узгодженість між навчальною та валідаційною вибірками. Різниця між ними не є критичною, тому модель можна вважати достатньо стабільною для використання в межах розробленого програмного модуля.

Крім якості класифікації, було оцінено роботу модуля рекомендації достовірного контенту. Його перевірка виконувалася на прикладах, де вхідна новина визначалася як фейкова або сумнівна. У таких випадках система формувала список тематично близьких достовірних матеріалів із підготовленого корпусу. Аналіз показав, що рекомендаційний модуль у більшості випадків повертав новини, пов’язані з тією ж темою, що й вхідне повідомлення (таблиця 3.13).

Таблиця 3.13 – Приклад роботи модуля рекомендації

Вхідна новина	Результат класифікації	Рекомендований контент
Маніпулятивне повідомлення про політичну подію	Фейкова	Перевірені новини з офіційних джерел про ту саму подію
Недостовірна новина про надзвичайну ситуацію	Фейкова	Офіційні повідомлення та новинні матеріали з підтвердженими фактами
Сумнівне повідомлення про суспільну тему	Підозріла / фейкова	Тематично близькі достовірні статті

Приклад роботи інтегрованого програмного модуля, який виконує класифікацію новини та формує рекомендації достовірного контенту, наведено на рисунку 3.14.

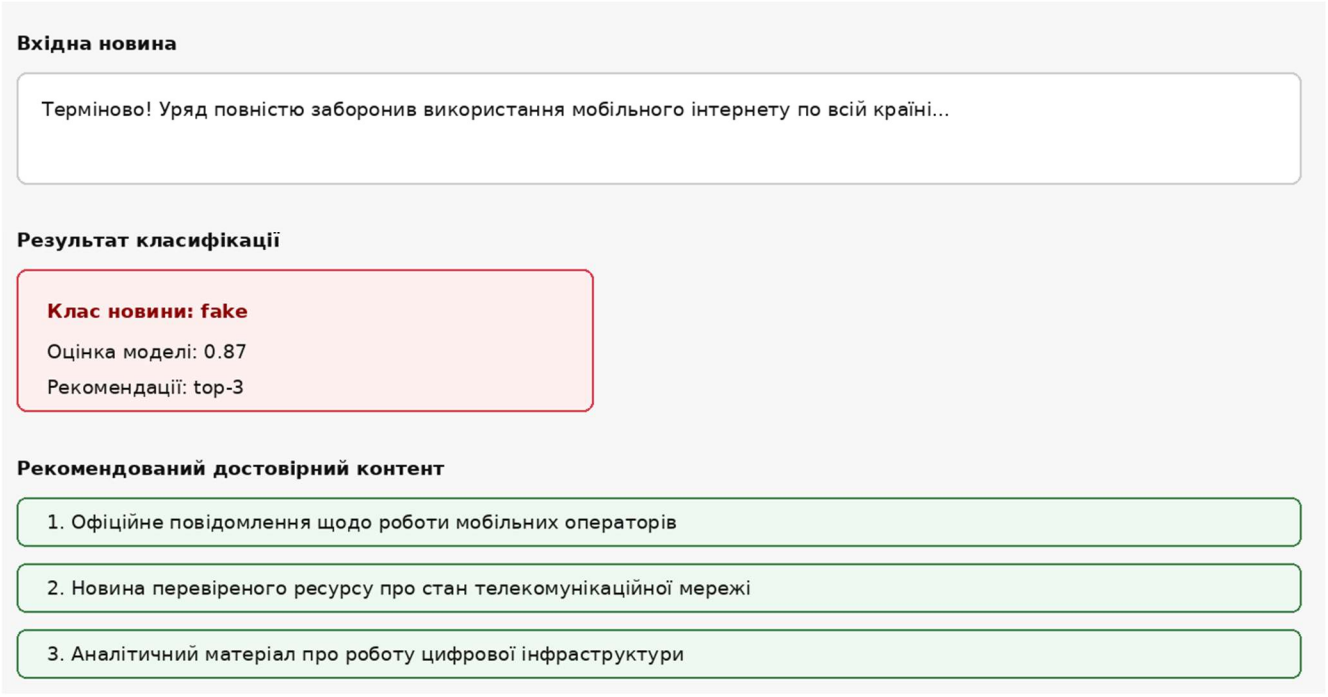

Рисунок 3.14 – Приклад класифікації новини та формування рекомендацій

Практичний аналіз показав, що рекомендаційний модуль підвищує цінність усієї системи. Якщо класифікаційна частина лише повідомляє про ризик недостовірності, то рекомендаційна частина допомагає користувачу швидко перейти до більш надійного інформаційного джерела. Саме це робить модуль не просто інструментом фільтрації, а засобом підтримки інформаційної орієнтації користувача.

Разом із позитивними результатами було виявлено і певні обмеження. По-перше, якість моделі залежить від складу навчального набору даних. Якщо в наборі недостатньо прикладів складних або змішаних форм дезінформації, точність системи може знижуватися. По-друге, рекомендаційний модуль у поточній реалізації базується на змістовій подібності, тому не враховує персональні інтереси користувача. По-третє, система орієнтована переважно на текстовий аналіз і не використовує додаткові сигнали, наприклад соціальний контекст або мультимодальні дані.

Попри це, отримані результати дозволяють вважати розроблений модуль працездатним і придатним для подальшого розвитку. Реалізована CNN-LSTM модель забезпечує достатній рівень якості класифікації, а модуль рекомендації підвищує практичну корисність системи. Така комбінація є доцільною саме для

прикладного програмного рішення, орієнтованого на виявлення фейкових новин і добір достовірного контенту.

Отже, результати експериментального дослідження підтверджують, що розроблений програмний модуль здатний виконувати основні функції, закладені в його архітектурі. Система забезпечує автоматизоване виявлення фейкових новин, формує рекомендації достовірного контенту та може розглядатися як основа для подальшого вдосконалення й розширення.

ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальну задачу розроблення програмного модуля виявлення фейкових новин та рекомендації достовірного контенту на основі глибоких нейронних мереж. Актуальність теми зумовлена тим, що в сучасному цифровому середовищі неправдиві та маніпулятивні повідомлення поширюються швидко, а їх своєчасне виявлення разом із наданням користувачеві достовірної альтернативи має практичне значення для підвищення якості споживання інформації.

Отримано наступні висновки:

1. Проаналізовано предметну область виявлення фейкових новин і рекомендації достовірного контенту. Встановлено, що фейкові новини є складним типом інформаційного контенту, який може містити не лише повністю хибні твердження, а й частково спотворену, маніпулятивно подану або контекстуально викривлену інформацію. Визначено, що для прикладної системи недостатньо лише встановити факт недостовірності новини, оскільки важливо також запропонувати користувачу перевірені матеріали тієї ж тематики.

2. Досліджено існуючі методи класифікації фейкових новин та підходи до рекомендаційних систем. За результатами аналізу встановлено, що класичні методи машинного навчання є простішими в реалізації, однак глибокі нейронні мережі краще враховують змістові та послідовні особливості тексту. Також визначено, що для поставленої задачі доцільним є контентно-орієнтований підхід до рекомендації, оскільки він дозволяє добирати тематично подібні та більш надійні інформаційні матеріали.

3. Обґрунтовано вибір методів і засобів реалізації програмного модуля. Для класифікації новин обрано модель типу CNN-LSTM, яка поєднує переваги згорткових і рекурентних нейронних мереж. Така архітектура дозволяє виявляти як локальні текстові шаблони, так і послідовні залежності між словами та фрагментами повідомлення. Для рекомендаційного блоку обрано контентно-орієнтований підхід на основі TF-IDF та косинусної подібності, що є технічно доцільним і достатнім для реалізації прототипу системи.

4. Спроєктовано архітектуру програмного модуля, яка включає модуль завантаження даних, модуль попередньої обробки тексту, модуль векторного подання, модуль класифікації новин, модуль рекомендації достовірного контенту та модуль подання результатів. Визначено логіку взаємодії між цими компонентами та змодельовано послідовність обробки новинного контенту від моменту надходження тексту до формування підсумкового висновку для користувача.

5. Реалізовано модуль попередньої обробки новинного тексту, що забезпечує очищення, нормалізацію та підготовку повідомлень до подальшого аналізу. Реалізовано модель класифікації на основі CNN-LSTM, здатну визначати рівень достовірності новини. Також реалізовано модуль рекомендації, який виконує пошук тематично близьких матеріалів серед корпусу достовірного контенту та формує перелік рекомендацій для користувача.

6. Проведено експериментальне дослідження роботи програмного модуля. Для оцінювання якості класифікації використано метрики accuracy, precision, recall і F1-score. Отримані результати показали, що модель забезпечує достатній рівень точності виявлення фейкових новин, а також стабільну роботу в межах обраного сценарію використання. Додатково підтверджено практичну доцільність рекомендаційного модуля, який у випадку виявлення фейкового або підозрілого контенту пропонує користувачеві тематично близькі достовірні матеріали.

7. Перспективи подальшого розвитку роботи полягають у розширенні набору даних, використанні україномовних спеціалізованих корпусів, додаванні мультимодального аналізу, урахуванні соціального контексту поширення новин, а також у вдосконаленні рекомендаційного механізму за рахунок поєднання контентного та персоналізованого підходів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Cahyani D. E., Patasik I. Performance comparison of TF-IDF and Word2Vec models for emotion text classification // Bulletin of Electrical Engineering and Informatics. 2021. Vol. 10, no. 5. P. 2780–2788.
2. Comito C., Caroprese L., Zumpano E. Multimodal fake news detection on social media: a survey of deep learning techniques // Social Network Analysis and Mining. 2023. Vol. 13. Art. 101.
3. Cura A., Küçük H., Ergen E. et al. Driver Profiling Using Long Short Term Memory (LSTM) and Convolutional Neural Network (CNN) Methods // IEEE Transactions on Intelligent Transportation Systems. 2021. Vol. 22, no. 10. P. 6572–6582.
4. 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.
5. Diakopoulos N., Trielli D., Lee G. Towards Understanding and Supporting Journalistic Practices Using Semi-Automated News Discovery Tools // Proceedings of the ACM on Human-Computer Interaction. 2021. Vol. 5, CSCW2. Art. 406. P. 1–30.
6. Fernández M., Bellogín A., Cantador I. Analysing the Effect of Recommendation Algorithms on the Amplification of Misinformation // Companion Proceedings of the Web Conference 2021. 2021.
7. Guo B., Ding Y., Yao L., Liang Y., Yu Z. The future of false information detection on social media: New perspectives and trends // ACM Computing Surveys. 2020. Vol. 53, no. 4. Art. 68.
8. Hopp T., Ferrucci P., Vargo C. J. Why Do People Share Ideologically Extreme, False, and Misleading Content on Social Media? A Self-Report and Trace Data-Based Analysis of Countermedia Content Dissemination on Facebook and Twitter // Human Communication Research. 2020. Vol. 46, no. 4. P. 357–384.
9. Hunt K., Agarwal P., Al Aziz R., Zhuang J. Fighting fake news during disasters // OR/MS Today. 2020. Vol. 47, no. 1. P. 34–39.
10. Kaliyar R. K., Goswami A., Narang P., Sinha S. FakeBERT: Fake news detection in social media with a BERT-based deep learning approach // Multimedia Tools and Applications. 2021. Vol. 80. P. 11765–11788.

11. Khan A., Brohman K., Addas S. The anatomy of “fake news”: Studying false messages as digital objects // Journal of Information Technology. 2022. Vol. 37, no. 2. P. 122–143.

12. Lipianina-Honcharenko K., Bohuta H., Ivaniush A., Soia M. A Dataset of Real and Synthetic Speech in Ukrainian // Scientific Data. 2025. Vol. 12, no. 1. Art. 745.

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

14. 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. No. 3. P. 202–230.

15. Lipianina-Honcharenko K., Lendiuk D., Melnyk N., Komar M., Lendiuk T. Evaluation of the Keyword Selection Methods Effectiveness for the Fake News Classification // CEUR Workshop Proceedings. 2024. Vol. 3909. P. 109–122.

16. 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.

17. 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.

18. Mehta N., Pacheco M. L., Goldwasser D. Tackling Fake News Detection by Continually Improving Social Context Representations using Graph Neural Networks // Proceedings of ACL 2022. 2022. P. 2705–2718.

19. Molina M. D., Sundar S. S., Le T., Lee D. “Fake News” Is Not Simply False Information: A Concept Explication and Taxonomy of Online Content // American Behavioral Scientist. 2021. Vol. 65, no. 2. P. 180–212.

20. Oh S., Baldonado M. Q. W., Kumar N. et al. Designing and Evaluating Presentation Strategies for Fact-Checking Articles // Proceedings of CIKM 2023. 2023.

21. Pathak R., Garimella V. R. K., Spezzano F. Understanding the Contribution of Recommendation Algorithms to the Amplification of Misinformation // ACM Transactions on Recommender Systems. 2023. Vol. 1, no. 4. Art. 26.

22. Pereira A., Harris E. A., Van Bavel J. J. Identity concerns drive belief: The impact of partisan identity on the belief and dissemination of true and false news // Group Processes & Intergroup Relations. 2023. Vol. 26, no. 1. P. 24–47.

23. Pitsun O., Melnyk N., Lipianina-Honcharenko K. Deepfake Detection Analysis Based on Video Face Analysis // 2024 IEEE 19th International Conference on Computer Science and Information Technologies (CSIT). 2024.

24. Reis J. C. S., Correia A., Murai F., Veloso A., Benevenuto F. Explainable Machine Learning for Fake News Detection // Proceedings of WebSci ’19. 2019. P. 17–26.

25. Rodríguez C. P., Carballido B. V., Redondo-Sama G. et al. False News Around COVID-19 Circulated Less on Sina Weibo Than on Twitter. How to Overcome False Information? // International and Multidisciplinary Journal of Social Sciences. 2020. Vol. 9, no. 2. P. 107–128.

26. Saikia P., Gundale K., Jain A. et al. Modelling Social Context for Fake News Detection: A Graph Neural Network Based Approach // 2022 International Joint Conference on Neural Networks (IJCNN). 2022.

27. Sallami D., Salem R. B., Aimeur E. Trust-based Recommender System for Fake News Mitigation // Proceedings of UMAP ’23. 2023.

28. Shu K., Mahudeswaran D., Wang S., Lee D., Liu H. FakeNewsNet: A Data Repository with News Content, Social Context, and Spatiotemporal Information for Studying Fake News on Social Media // Big Data. 2020. Vol. 8, no. 3. P. 171–188.

29. Shu K., Sliva A., Wang S., Tang J., Liu H. Fake News Detection on Social Media: A Data Mining Perspective // ACM SIGKDD Explorations Newsletter. 2017. Vol. 19, no. 1. P. 22–36.

30. Tsfati Y., Boomgaarden H. G., Strömbäck J. et al. Causes and consequences of mainstream media dissemination of fake news: literature review and synthesis // Annals of the International Communication Association. 2020. Vol. 44, no. 2. P. 157–173.

31. Wang C. C. Fake news and related concepts: Definitions and recent research development // Contemporary Management Research. 2020. Vol. 16, no. 3. P. 145–174.

32. Wang R., He Y., Xu J. et al. Fake news or bad news? Toward an emotion-driven cognitive dissonance model of misinformation diffusion // Asian Journal of Communication. 2020. Vol. 30, no. 5. P. 317–342.

33. Zhang Y., Yang Q. A Survey on Multi-Task Learning // IEEE Transactions on Knowledge and Data Engineering. 2021. Vol. 34, no. 12. P. 5586–5609.

34. Zhou X., Zafarani R. A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities // ACM Computing Surveys. 2020. Vol. 53, no. 5. Art. 109.

35. Комар М., Ліп'яніна-Гончаренко Х., Кіт І., Мадараш Р., Юрків Х. Інтелектуальний метод виявлення джерел мультилінгвальної дезінформації // Measuring and Computing Devices in Technological Processes. 2023. С. 221–230.

36. Лендюк Д. Т., Ліп'яніна-Гончаренко Х. В. Ансамблеве навчання класифікаторів для онлайн виявлення дезінформації // Таврійський науковий вісник. Серія: Технічні науки. 2024. С. 46–63.

37. Мадараш Р., Ліп'яніна-Гончаренко Х. Аналіз літератури та підходів до класифікації фейкових новин за допомогою машинного та глибокого навчання // UNIVERSUM. 2024. Вип. 7. С. 204–212.

38. Мельник Н.Б., Стрижеус М.Ю. Виявлення фейкових новин та формування рекомендацій достовірного контенту на основі глибоких 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. 54-56.

39. FakeNewsNet. URL: https://www.kaggle.com/datasets/mdepak/fakenewsnet (дата звернення: 12.05.2026).

40. Островерхов В. М., Біловус Л. І., Возьний К. З., Луцишин О. О., Монастирський Г. Л., Надвиничний С. А., Питель С. В., Шандрук С. К. 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.

41. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С., Ліп'яніна-Гончаренко Х.В. 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.

Додаток А

Лістинги програмного забезпечення

Завантаження та первинна підготовка даних:

```
import pandas as pd

def load_data(path):
    data = pd.read_csv(path)
    data = data[['title', 'text', 'label']]
    data = data.dropna()
    data['content'] = data['title'] + ' ' + data['text']
    return data[['content', 'label']]
```

Очищення тексту новини:

```
import re

def clean_text(text):
    text = text.lower()
    text = re.sub(r"http\S+", " ", text)
    text = re.sub(r"www\S+", " ", text)
    text = re.sub(r"[^a-zA-Za-яА-ЯіїєґІЇЄҐ0-9\s]", " ", text)
    text = re.sub(r"\s+", " ", text).strip()
    return text
```

Попередня обробка всього набору текстів:

```
def preprocess_corpus(texts):
    cleaned_texts = []
    for text in texts:
        cleaned_texts.append(clean_text(str(text)))
    return cleaned_texts
```

Токенізація та перетворення текстів у послідовності:

```
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences

def prepare_sequences(texts, max_words=20000, max_len=300):
    tokenizer = Tokenizer(num_words=max_words, oov_token="<UNK>")
    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
```

Поділ даних на навчальну, валідаційну та тестову вибірки:

```
from sklearn.model_selection import train_test_split

def split_data(X, y):
    X_temp, X_test, y_temp, y_test = train_test_split(
        X, y, test_size=0.15, random_state=42, stratify=y
```

```
    )

    X_train, X_val, y_train, y_val = train_test_split(
        X_temp, y_temp, test_size=0.1765, random_state=42, stratify=y_temp
    )

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

Побудова CNN-LSTM моделі:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, Conv1D, MaxPooling1D
from tensorflow.keras.layers import LSTM, Dense, Dropout, Bidirectional

def build_cnn_lstm_model(vocab_size, max_length, embedding_dim=128):
    model = Sequential([
        Embedding(
            input_dim=vocab_size,
            output_dim=embedding_dim,
            input_length=max_length
        ),
        Conv1D(
            filters=128,
            kernel_size=5,
            activation='relu',
            padding='same'
        ),
        MaxPooling1D(pool_size=2),
        Bidirectional(LSTM(64, return_sequences=False)),
        Dropout(0.5),
        Dense(64, activation='relu'),
        Dropout(0.3),
        Dense(1, activation='sigmoid')
    ])

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

Навчання моделі класифікації:

```
def train_model(model, X_train, y_train, X_val, y_val):
    history = model.fit(
        X_train,
        y_train,
        validation_data=(X_val, y_val),
        epochs=10,
        batch_size=32,
        verbose=1
    )
    return history
```

Оцінювання якості моделі:

```
from sklearn.metrics import accuracy_score, precision_score
from sklearn.metrics import recall_score, f1_score

def evaluate_model(model, X_test, y_test):
    y_pred_prob = model.predict(X_test)
    y_pred = (y_pred_prob > 0.5).astype("int32")
```

```
    metrics = {
        "accuracy": accuracy_score(y_test, y_pred),
        "precision": precision_score(y_test, y_pred),
        "recall": recall_score(y_test, y_pred),
        "f1_score": f1_score(y_test, y_pred)
    }
    return metrics
```

Формування TF-IDF подання достовірного корпусу:

```
from sklearn.feature_extraction.text import TfidfVectorizer

def build_trusted_index(trusted_texts):
    vectorizer = TfidfVectorizer()
    tfidf_matrix = vectorizer.fit_transform(trusted_texts)
    return vectorizer, tfidf_matrix
```

Рекомендація достовірного контенту за косинусною подібністю:

```
from sklearn.metrics.pairwise import cosine_similarity

def recommend_news(query_text, trusted_texts, top_k=3):
    vectorizer = TfidfVectorizer()
    all_texts = [query_text] + trusted_texts
    tfidf_matrix = vectorizer.fit_transform(all_texts)

    similarities = cosine_similarity(
        tfidf_matrix[0:1],
        tfidf_matrix[1:]
    ).flatten()

    top_indices = similarities.argsort()[-top_k:][::-1]
    return top_indices
```

Отримання текстів рекомендованих новин:

```
def get_recommendations(query_text, trusted_df, top_k=3):
    trusted_texts = trusted_df['content'].tolist()
    top_indices = recommend_news(query_text, trusted_texts, top_k=top_k)

    recommendations = []
    for idx in top_indices:
        recommendations.append({
            "text": trusted_df.iloc[idx]['content'],
            "label": trusted_df.iloc[idx]['label']
        })
    return recommendations
```

Інтегрована обробка однієї новини:

```
from tensorflow.keras.preprocessing.sequence import pad_sequences

def process_news(text, tokenizer, model, trusted_df, max_len=300):
    cleaned = clean_text(text)
    sequence = tokenizer.texts_to_sequences([cleaned])
    padded = pad_sequences(sequence, maxlen=max_len, padding='post')

    score = float(model.predict(padded)[0][0])
    label = "fake" if score > 0.5 else "real"

    recommendations = []
```

```
    if label == "fake":
        recommendations = get_recommendations(cleaned, trusted_df, top_k=3)

    return {
        "label": label,
        "score": score,
        "recommendations": recommendations
    }
```

Основний сценарій запуску програми:

```
def main():
    data = load_data("news_dataset.csv")
    data['content'] = preprocess_corpus(data['content'])

    X, tokenizer = prepare_sequences(data['content'])
    y = data['label'].values

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

    model = build_cnn_lstm_model(vocab_size=20000, max_length=300)
    train_model(model, X_train, y_train, X_val, y_val)

    metrics = evaluate_model(model, X_test, y_test)
    print("Результати оцінювання:", metrics)

    trusted_df = data[data['label'] == 0].reset_index(drop=True)

    sample_text = "Breaking news example text..."
    result = process_news(sample_text, tokenizer, model, trusted_df)

    print("Клас новини:", result["label"])
    print("Оцінка моделі:", result["score"])
    print("Кількість рекомендацій:", len(result["recommendations"]))

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

Збереження навченої моделі та токенізатора:

```
import pickle

def save_artifacts(model, tokenizer, model_path="cnn_lstm_model.h5",
                   tokenizer_path="tokenizer.pkl"):
    model.save(model_path)
    with open(tokenizer_path, "wb") as file:
        pickle.dump(tokenizer, file)
```

Завантаження навченої моделі та токенізатора:

```
import pickle
from tensorflow.keras.models import load_model

def load_artifacts(model_path="cnn_lstm_model.h5",
                   tokenizer_path="tokenizer.pkl"):
    model = load_model(model_path)
    with open(tokenizer_path, "rb") as file:
        tokenizer = pickle.load(file)
    return model, tokenizer
```

Додаток Б

Тестові приклади роботи програмного модуля

Б.1 Приклад обробки достовірної новини

У першому тестовому прикладі на вхід системи подано новинне повідомлення, яке містить нейтральний виклад події, конкретні факти та не має виражених маніпулятивних ознак.

Таблиця Б.1 – Приклад обробки достовірної новини

Параметр	Значення
Тип вхідного повідомлення	новина з ознаками достовірного контенту
Характер подачі	нейтральний
Результат класифікації	real
Оцінка моделі	0.14
Формування рекомендацій	не є обов’язковим

Вхідний текст:

Офіційні представники міської ради повідомили про завершення ремонту ділянки дороги. Роботи виконувалися упродовж двох тижнів, після чого рух транспорту було повністю відновлено.

Результат роботи програми:

– клас новини: real;

– оцінка моделі: 0.14;

– рекомендації: не сформовано, оскільки новина класифікована як достовірна.

Б.2 Приклад обробки фейкової новини

У другому прикладі на вхід модуля подано повідомлення з ознаками сенсаційності, узагальненими твердженнями та відсутністю підтверджених фактів. У такому випадку система повинна виявити ознаки фейковості та запропонувати

користувачу достовірний контент на ту саму тему.

Таблиця Б.2 – Приклад обробки фейкової новини

Параметр	Значення
Тип вхідного повідомлення	новина з ознаками фейкового контенту
Характер подачі	емоційно маніпулятивний
Результат класифікації	fake
Оцінка моделі	0.87
Формування рекомендацій	так

Вхідний текст:

Терміново! Уряд повністю заборонив використання мобільного інтернету по всій країні. Джерела стверджують, що це рішення почне діяти вже найближчими годинами.

Результат роботи програми:

– клас новини: fake;

– оцінка моделі: 0.87;

– активовано модуль рекомендації.

Таблиця Б.3 – Приклад рекомендованого достовірного контенту

Рекомендований матеріал	Характеристика
Офіційне повідомлення державного органу щодо роботи мобільних операторів	містить перевірену інформацію
Новина авторитетного медіа про стан телекомунікаційної мережі	тематично пов’язана
Пояснювальний матеріал про зміни в роботі зв’язку	уточнює реальний стан подій

Б.3 Приклад обробки сумнівної новини

У третьому прикладі новина не є явно фейковою, але містить неперевірені

твердження, емоційні елементи та спрощені висновки. У такій ситуації система може віднести її до фейкового або підозрілого контенту залежно від встановленого порогу класифікації.

Таблиця Б.4 – Приклад обробки сумнівної новини

Параметр	Значення
Тип вхідного повідомлення	сумнівна новина
Характер подачі	частково маніпулятивний
Результат класифікації	fake
Оцінка моделі	0.63
Формування рекомендацій	так

Вхідний текст:

Експерти повідомляють, що вже наступного місяця більшість банків можуть повністю припинити обслуговування карткових рахунків. Користувачам рекомендують негайно знімати всі кошти.

Результат роботи програми:

– клас новини: fake;

– оцінка моделі: 0.63;

– сформовано рекомендації з достовірного корпусу.

Б.4 Приклад роботи модуля рекомендації

Для демонстрації роботи рекомендаційного модуля доцільно подати приклад, у якому після класифікації фейкової новини система повертає кілька тематично близьких достовірних матеріалів.

Таблиця Б.5 – Приклад результату рекомендаційного модуля

Вхідна тема	Виявлений клас	Кількість рекомендацій	Результат
політична подія	fake	3	повернуто перевірені новини з надійних джерел
надзвичайна ситуація	fake	3	повернуто офіційні повідомлення і підтверджені факти
економічне повідомлення	fake	3	повернуто достовірні аналітичні матеріали

У такому випадку програмний модуль виконує дві функції одночасно: спочатку попереджає користувача про ризик недостовірності, а потім надає альтернативні матеріали, які можуть бути використані для уточнення інформації.

Б.5 Приклад структурованого результату програми

У програмній реалізації результат роботи модуля може бути поданий у вигляді структури даних. Приклад такого результату наведено нижче.

Лістинг Б.1 – Приклад структурованого результату роботи програми

```
{
    "label": "fake",
    "score": 0.87,
    "recommendations": [
        {
            "text": "Офіційне повідомлення про відсутність змін у роботі
мобільного зв’язку",
            "label": "real"
        },
        {
            "text": "Новина перевіреного інформаційного ресурсу про роботу
телекомунікацій",
            "label": "real"
        },
        {
            "text": "Аналітичний матеріал про стан мереж зв’язку",
            "label": "real"
        }
    ]
}
```

Додаток В

Копія публікації

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

Стрижеус Матвій Юрійович, *бакалавр,*
Західноукраїнський національний університет, м. Тернопіль

ВИЯВЛЕННЯ ФЕЙКОВИХ НОВИН ТА ФОРМУВАННЯ РЕКОМЕНДАЦІЙ ДОСТОВІРНОГО КОНТЕНТУ НА ОСНОВІ ГЛИБОКИХ НЕЙРОННИХ МЕРЕЖ

Інтернет-адреса публікації на сайті:
http://www.konferenciaonline.org.ua/ua/article/id-2579/

Для України проблема дезінформації має не лише інформаційний, а й суспільний та безпековий вимір. Це пояснює зростання інтересу до методів виявлення інформаційних загроз у реальному часі, класифікації фейкових новин, аналізу тональності новин про Україну, виявлення джерел мультилінгвальної дезінформації та ін. [1, 2]. У таких умовах розроблення програмного модуля, який поєднує автоматичне виявлення фейкових новин і рекомендацію достовірного контенту, є доцільним як з наукового, так і з прикладного погляду.

Аналіз наукових джерел свідчить, що в наявних роботах увага часто зосереджується або на класифікації новин, або на окремих аспектах рекомендаційних систем [3-5]. Водночас меншою мірою розглядається інтегрований програмний підхід, у якому модуль виявлення фейків і модуль рекомендації достовірного контенту функціонують як єдина система.

Архітектура запропонованого програмного модуля включає шість основних компонентів: введення та завантаження новинних даних; попередня обробка тексту; векторне подання тексту; класифікація новин; рекомендація достовірного контенту; подання результатів (рисунок 1).

Перший компонент відповідає за приймання вхідної інформації. Джерелом даних може бути окремий текст новини, заголовок, короткий опис або сформований набір новинних повідомлень для тестування системи. На цьому етапі здійснюється первинна перевірка даних: чи не є текст порожнім, чи коректно зчитано символи, чи відповідає вхідна інформація встановленому формату. Це базовий, але необхідний етап, оскільки помилки на вході безпосередньо впливають на роботу всіх наступних компонентів.

Рисунок 1 – Архітектура програмного модуля

Другий компонент виконує очищення вхідного повідомлення від зайвих символів, службових елементів, HTML-фрагментів, зайвих пробілів та інших шумових даних. Також на цьому етапі може здійснюватися приведення тексту до єдиного регістру, токенізація, видалення небажаних елементів і формування нормалізованої текстової послідовності. Завдання цього модуля полягає в тому, щоб перетворити сирий текст у впорядкований формат.

Третій компонент полягає у перетворенні обробленого тексту на числове представлення, яке може бути подане на вхід нейронної мережі. У межах цієї роботи доцільно використовувати словниковий підхід із перетворенням тексту на послідовність індексів або векторні подання слів. У результаті текстова інформація стає доступною для математичної обробки. Саме цей компонент є зв'язувальною ланкою між лінгвістичною обробкою тексту та роботою моделі глибокого навчання.

Четвертий компонент є центральним у структурі системи. Саме він виконує основне завдання: оцінює, чи належить повідомлення до достовірного або фейкового контенту. На вхід цього модуля надходить числове представлення тексту, а на виході формується результат класифікації. Ним може бути один клас, наприклад «достовірна новина» або «фейкова новина», а також значення ймовірності чи оцінки впевненості моделі. У межах обраної архітектури цей модуль реалізується на основі глибокої нейронної мережі, яка поєднує аналіз локальних текстових шаблонів і послідовних залежностей [3].

Призначення п'ятого компонента полягає в тому, щоб після завершення класифікації підібрати для користувача альтернативні матеріали, близькі за тематикою до вхідної новини, але відібрані з корпусу надійних джерел. Логіка цього модуля базується на порівнянні змісту

вхідного тексту з набором достовірних матеріалів. У найпростішому варіанті реалізації система обчислює змістову схожість між новиною та документами корпусу і повертає кілька найбільш релевантних результатів.

Шостий компонент відповідає за формування підсумкового висновку системи у зрозумілому вигляді. До такого результату можуть входити: текстове повідомлення про клас новини, оцінка достовірності, коротке пояснення, а також перелік рекомендованих достовірних матеріалів. У прикладній системі саме цей модуль забезпечує взаємодію з користувачем, тому його структура має бути простою, логічною та зручною для сприйняття.

Отже, спроєктована архітектура програмного модуля має модульну структуру, яка охоплює всі ключові етапи обробки новинного контенту: від введення тексту до формування рекомендацій.

Література:

1. Lipianina-Honcharenko K., Lendiuk D., Melnyk N., Komar M., Lendiuk T. Evaluation of the Keyword Selection Methods Effectiveness for the Fake News Classification // CEUR Workshop Proceedings. 2024. Vol. 3909. P. 109-122.
2. Комар М., Ліп'яніна-Гончаренко Х., Кіт І., Мадараш Р., Юрків Х. Інтелектуальний метод виявлення джерел мультилінгвальної дезінформації // Measuring and Computing Devices in Technological Processes. 2023. С. 221-230.
3. Kaliyar R. K., Goswami A., Narang P., Sinha S. FakeBERT: Fake news detection in social media with a BERT-based deep learning approach // Multimedia Tools and Applications. 2021. Vol. 80. P. 11765-11788.
4. 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.
5. Pathak R., Garimella V. R. K., Spezzano F. Understanding the Contribution of Recommendation Algorithms to the Amplification of Misinformation // ACM Transactions on Recommender Systems. 2023. Vol. 1, no. 4. Art. 26.