Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка
Навчально-науковий інститут філології
кафедра української мови та прикладної лінгвістики

АВТОМАТИЧНЕ ВИЯВЛЕННЯ ФЕЙКОВИХ НОВИН НА МАТЕРІАЛІ УКРАЇНСЬКОМОВНИХ ТЕКСТІВ

Кваліфікаційна робота
освітнього ступеня «магістр»
студентки ІІ курсу магістратури
освітньої програми
«Прикладна лінгвістика (редакторсько-перекладацька та експертна діяльність)»,
спеціальність – 035 Філологія
Софія Петрівна МАЦЬКОВИЧ
Наукові керівники:
д.філол.н., проф. Наталія ДАРЧУК,
Валентина РОБЕЙКО

«Допущено до захисту»
Протокол засідання
кафедри української мови та прикладної лінгвістики
протокол № 14 **від «** 8 **»** 05 **2024 року**
завідувач кафедри _____(підпис)
к.філол.н., доц. Сергій РІЗНИК

КИЇВ
2024

АНОТАЦІЯ

Велика кількість джерел інформації та складність їхнього аналізу заохочують використовувати методи автоматичної обробки природної мови для виявлення фейкових новин. Об’єктом дослідження є новинні тексти, написані українською мовою, а предметом — лінгвістичні ознаки фейків у них. Метою є створення застосунку для опрацювання фейкового контенту українською мовою. Для досягнення цієї мети було проаналізовано підходи до розуміння поняття “фейк”, описано загальні методи виявлення фейкових новин, досліджено способи розпізнавання оманливого контенту за допомогою обробки природної мови, проаналізовано мовні дані для створення класифікатора новин, навчено та порівняно кілька бінарних моделей для виявлення фейків та розроблено вебзастосунок. Під час здійснення дослідження було використано описовий та порівняльний методи, методи експерименту, кількісного аналізу та моделювання.

Результатом роботи є створений корпус з українськомовними новинними текстами та програма-детектор фейкових новин. У новинному корпусі представлено тексти, заголовки та класи, а також метадані: дата публікації, посилання. Застосунок для виявлення фейків поєднує інтерфейс користувача з мовною моделлю на основі нейронної мережі з архітектурою довгої короткочасної пам’яті. Для оцінювання ефективності навченої моделі у дослідженні використовуються такі показники: точність, повнота, влучність та міра F1. Основними компонентами цього застосунку є тренувальні дані, модуль попередньої обробки, цикл навчання бінарного класифікатора, backend-сервер та користувацький інтерфейс.

Дослідження складається з трьох розділів. Перший розділ розглядає теоретичні засади автоматичного виявлення фейкових новин. Другий розділ присвячений створенню корпусу текстів для навчання бінарних класифікаторів. Третій розділ описує розробку застосунку “Детектор фейкових новин”.

Ключові слова: автоматична обробка природної мови, класифікація тексту, виявлення фейкових новин, глибоке навчання.

The large number of information sources and the complexity of their analysis encourage the use of natural language processing to detect fake news. The object of the study is the news texts written in Ukrainian, and the subject is the linguistic features of fake news in those texts. The goal is to create an application for processing fake content in Ukrainian. To achieve this goal, the research analyses approaches to understanding the concept of a fake, describes general methods for detecting fake news, investigates ways to recognise misleading content using NLP, analyses speech data to create a news classifier; several binary models for fake news detection were trained and compared, and a web application was developed. The research was conducted using descriptive and comparative methods as well as experiment, quantitative analysis, and modelling.

The result of the work is a corpus of Ukrainian-language news texts and a fake news detector. The news corpus contains texts, headlines, and classes, as well as metadata such as publication date and links. The fake news detector combines a user interface with an LSTM-based language model. To evaluate the effectiveness of the trained model, the study uses the following metrics: accuracy, completeness, precision, and F1 measure. The main components of this application are training data, a preprocessing module, a binary classifier training cycle, a backend server, and a user interface.

The study consists of three chapters. The first section discusses the theoretical foundations of automatic fake news detection. The second section is devoted to the creation of a text corpus for training binary classifiers. The third section describes the development of the Fake News Detector application.

Keywords: natural language processing, text classification, fake news detection, deep learning.

ЗМІСТ

ВСТУП

У сучасну цифрову епоху поширення інформації в Інтернеті змінило спосіб, у який ми споживаємо новини. Така доступність підвищує масштаби та швидкість розповсюдження фейків. **Актуальність** цього дослідження зумовлена великою кількістю неправдивого та оманливого контенту в медійних текстах та потребою автоматизувати процес розпізнавання таких новин. Через російську агресію проти України виявлення фейкових українськомовних новин є критично важливим для захисту населення від пропаганди та дезінформації.

Мета цієї роботи — створити програмний продукт для виявлення фейкових українськомовних новин. Для досягнення мети потрібно виконати такі **завдання**:

1) визначити поняття “фейк” та дотичні терміни, проаналізувати різні підходи до розуміння концепції фейку;
2) описати загальні методи виявлення фейкових новин;
3) усебічно дослідити способи розпізнавання оманливого контенту за допомогою засобів обробки природної мови;
4) створити українськомовний корпус анотованих новинних текстів;
5) навчити та порівняти між собою кілька бінарних моделей для виявлення фейкової інформації, проаналізувати їхню ефективність;
6) розробити ефективний та надійний інструмент, котрий можна використовувати для розпізнавання фейкових новин українською мовою.

Об’єктом цього дослідження є писемні українськомовні медійні тексти, а його **предметом** — лінгвістичні ознаки фейків у писемних українськомовних медійних текстах, які можна виявити за допомогою методів і алгоритмів автоматичної обробки природної мови.

Матеріалом дослідження є українськомовні новини, а саме 12956 текстів в основному корпусі та 78 у тестовій вибірці.

Для досягнення мети та виконання завдань роботи ми використовували такі **методи** дослідження: описовий та порівняльний, методи експерименту, кількісного аналізу та моделювання.

Методологічною основою цього дослідження є праці С. Афроза та ін. (2012), Т. Мітра та Е. Ґілберта (2015), С. Волкової (2017), Т. Алхінді та ін. (2018), Л. Борхеса та ін. (2019), М. Маянка та ін. (2022), Ч. Й. Пака та ін. (2022), Ч. Хелве та ін. (2023).

Новизна полягає в роботі з українськомовними текстами, адже більшість досліджень з виявлення фейкових новин аналізують головним чином англійськомовні медіа. Оскільки нашою кінцевою метою є розробка програмного продукту, ця робота має **практичне значення** для захисту українського інформаційного простору та підвищення медіаграмотності через ознайомлення користувачів з характеристиками фейкових новин.

Структура й **обсяг** кваліфікаційної роботи. Робота складається зі вступу, трьох розділів: теоретичної частини (Розділ 1. “Теоретичне підґрунтя автоматичного виявлення фейкових новин”) та практичних результатів (Розділ 2. “Створення корпусу українськомовних новинних текстів”; Розділ 3. “Створення застосунку для виявлення фейкових новин на основі стилю”), висновків до кожного розділу та загальних висновків, списку використаних джерел, додатків. Загальний обсяг магістерської роботи — 74 сторінки, із яких основний текст охоплює 55 сторінок, список використаних джерел (103 найменування) — 12, додатки — 7.

РОЗДІЛ 1. ТЕОРЕТИЧНЕ ПІДҐРУНТЯ АВТОМАТИЧНОГО ВИЯВЛЕННЯ ФЕЙКОВИХ НОВИН

У цьому розділі ми розглянемо проблему фейкових новин, дамо визначення ключовим поняттям у контексті дезінформації та фейків. Спираючись на результати психологічних та соціологічних досліджень, обговоримо психологічні упередження та вплив соціальних мереж як домінантного джерела інформації в Україні. Крім цього, ми опишемо використання методів обробки природної мови для виявлення фейкових новин, включаючи засоби, засновані на перевірці фактів та аналізі стилю тексту, що варіюються від простих статистичних моделей до складних нейронних мереж. Нарешті ми розглянемо найновіші дослідження у сфері виявлення фейкових новин, висвітлюючи методи покращення ефективності мовних моделей.

1.1 Обґрунтування проблеми

Сьогодення характеризується значною кількістю фейкової інформації в Інтернеті, і її швидке поширення є великою проблемою. Згідно з опитуванням USAID, у 2023 році основним джерелом новин для українців є соціальні мережі (76% респондентів) та вебсайти (41%). Обізнаність щодо дезінформації є високою (84%), але здатність розпізнавати такий контент дещо знизилася (з 72% до 67%) [18, с. 5–6].

Споживачі медіа схильні вірити у правдивість новинних текстів через психологічні упередження. Люди надають перевагу даним, що підтверджують їхні попередні погляди (так зване вибіркове сприйняття — англ. *selective exposure*), вважають більш переконливою ту інформацію, що їм подобається (упередження бажаності — англ. *desirability bias*) [89, с. 1095].

Серйозність ситуації підсилюється ще й тим, що створити вебсайт, котрий виглядає, як професійний новинний ресурс, відносно нескладно: це не потребує значних фінансових вкладень; контент легко монетизувати завдяки онлайн-рекламі; алгоритми соціальних мереж сприяють поширенню такої інформації [89, с. 1096].

Окрім традиційних вебсторінок, що імітують серйозні видання, значною проблемою є кількість шкідливого контенту в Telegram-каналах. Українці відзначають, що під час війни кількість дезінформації на платформі Telegram збільшилася, через що аудиторія відписується від таких джерел [18, с. 18]. Найпоширенішими каналами у 2023 році стали "Труха Україна", "Украина сейчас", "Лачен пише", "ТСН новини" [18, с. 5], у яких неодноразово публікувалася фейкова інформація [8].

Причини створення фейкових новин різні. Серед них: поширення паніки, розпалювання ворожнечі, заплутування та відволікання від правди, реклама, прибуток, маніпуляції свідомістю, заплямування репутації, для розваги [10, с. 30]. Згідно з Г. Олкоттом та ін., є дві основні мотивації для поширення фейкових новин: фінансова (фейкові новини стають "вірусними" та приносять значний дохід від реклами, коли користувачі переходять на сторінку, де опубліковано оригінал) та ідеологічна (наприклад, під час виборчих кампаній медійні ресурси використовують фейкові новини для просування "своїх" кандидатів) [25, с. 217].

В українському інформаційному просторі згадана ідеологічна мотивація найбільш яскраво виражається в контексті російсько-української війни, особливо з початком повномасштабного воєнного вторгнення росії в лютому 2022 року. Новини є засобом ведення інформаційної війни й використовуються для розповсюдження пропаганди, деморалізації цивільного населення та військових, підривання якості інформаційних ресурсів тощо [21, с. 275].

Беручи до уваги зазначені вище причини поширення фейкових новин, масовість цифрових джерел, а також складність аналізувати прочитану інформацію, зазначимо, що проблема фейків у медійних текстах є нагальною та потребує розв'язання. Згідно з Д. Лейзером та ін., заходи з перешкоджання поширенню фейкових новин можна поділити на дві категорії: ті, що допомагають пересічним користувачам ідентифікувати такий контент, та фундаментальні структурні зміни, що мають на меті унеможливити доступ до фейкових новин як таких [89, с. 1095]. Друге розв'язання стосується рішень на рівні урядів країн та організацій і виходить за межі нашого дослідження.

Що ж стосується першого розв’язання, то це включає фактчекінгові сайти й організації, а також інформаційні кампанії, спрямовані на інформування суспільства щодо фейкових новин, як-от порадник, виданий Фондом Фрідріха Науманна за Свободу [61].

Однак покладатися на спроби вручну розпізнавати фейки — не найкращий вихід. Як справедливо зазначають автори книги “Foundations of Statistical Natural Language Processing” (укр. “Основи статистичної обробки природної мови”), в епоху поширення цифрової інформації компанії, урядові організації та окремі особи мають справу з великими обсягами тексту, однак не завжди добре розуміють, як видобути з них приховану суть [63, с. 29].

Способом розв’язання проблем, пов’язаних з опрацюванням великої кількості текстової інформації (куди входить і вище описаний сплеск дезінформації та фейкових новин) є використання методів автоматичної обробки природної мови (АОПМ). Методи АОПМ можна використовувати для автоматичної класифікації медійного тексту як фейкового чи справжнього, екстракції тверджень з тексту і порівняння їх з фактами, екстракції джерел, на які посилається текст, тощо.

Методи АОПМ та інші способи автоматизації виявлення фейкових новин можуть інтегруватися з фактчекінговими сайтами, соціальними мережами, форумами тощо. Ці платформи можуть подавати споживачам сигнали про якість джерела та правдивість контенту, а також використовувати розроблені інструменти в алгоритмах ранжування та рекомендування інформації [89, с. 1096].

1.2 Дефініції

Автори публікації “Approaches to Identify Fake News: A Systematic Literature Overview” (укр. “Підходи до виявлення фейкових новин: систематичний огляд літератури”), намагаючись витягнути за допомогою пошукових систем різні дослідження на тему фейкових новин на основі методів АОПМ, використовують такі ключові слова в пошуку: фейкові новини (*fake*

news), неточна інформація (*inaccurate report*), ненадійна інформація (*untrustworthy information*), хибні новини (*false news*) тощо [36, с. 15]. Лексикон пошукового запиту є таким широким через те, що серед дослідників немає згоди щодо називання одного й того самого явища спільним терміном. Отже, даючи визначення поняття "фейкова новина", будемо брати до уваги й інші терміни.

Згідно зі ще одним оглядом досліджень з виявлення фейкових новин, аж 42% таких робіт не містять визначення поняття "фейкова новина" [58, с. 8]. У нашій роботі спробуємо розтлумачити терміни "фейк" та "фейкова новина" та з'ясувати кореляцію між ними та такими поняттями, як місінформація, дезінформація, дипфейк, сатира, чутки.

Фейк — це "будь-яка підробка, яку хтось намагається видати за оригінал" [11, с. 282]. Це може бути акаунт у соцмережі або одяг [11, с. 283]. Фейкову новину вважаємо різновидом фейку.

Щодо самого визначення **фейкових новин** однозначної згоди немає. Одна група дослідників (як лінгвісти, так і науковці з інших сфер) стверджують, що в основі поняття фейкової новини лежить злий намір. М. Кіца вважає, що фейк — це саме навмисне перекручення фактів, тоді як будь-яку хибну інформацію, подану випадково або через неуважність, названо **неправдивою інформацією** [11, с. 283]. Синонімом до поняття фейкові новини в одній із українськомовних публікацій є термін **псевдоновини**, "навмисно оприлюднена в мас-медіа неправдива інформація, що має на меті дезінформувати, ввести в оману споживачів інформації" [16, с. 46].

О. Грищенко теж погоджується, що основна функція фейків — саме маніпулятивна; за допомогою цієї функції відбувається навмисний вплив на свідомість [5, с. 40].

Друга група дослідників вважає фейковою будь-яку новину з хибними твердженнями. Згідно з Д. Лейзером та ін., фейкова новина — це сфабрикована інформація, що імітує контент новинних медіа за формою, але не за організаційним процесом чи наміром [89, с. 1094]. Автори брошури "Як можна

протидіяти фейковим новинам?” за фейк вважають будь-якого роду фальшиву інформацію в Інтернеті [14].

Третім підходом можна вважати такий, за якого ми погоджуємося з існуванням наміру при публікації фейкових новин, однак ціль може бути позитивною. Наприклад, в українськомовному медійному просторі існує низка джерел, котрі публікують інформацію для морального піднесення: про близьку смерть президента росії В. Путіна, здобутки українських військових, постачання зброї Україні [11, с. 285].

Також, як справедливо зазначає Л. Доскіч, хоча саме поняття “фейк” відносно нове, саме явище існувало завжди, однак раніше для цього слугував термін “газетна качка” [7, с. 74].

Дезінформація — це “спосіб психологічного впливу, котрий полягає в поданні об’єктові такої інформації, яка вводить його в оману щодо справжнього стану справ та створює викривлену реальність” [10, с. 29].

Місінформація — це неправильна інформація [67]. Як бачимо з визначення, місінформація не має стосунку до наміру; навмисне подання є характеристикою дезінформації. Фейки є різновидом місінформації та можуть підпадати під дезінформацію залежно від того, яке визначення слова “фейк” ми оберемо.

Чутки — це непідтверджені цікаві історії або новини, що швидко поширюються від людини до людини [75]. **Сатира** використовується для гумористичної критики людей чи ідей, щоб показати їхні помилки або недоліки [21]. Прикладом сатиричних новин в Україні є UaReview [93]. У літературі терміни “чутки” та “сатиричні новини” на позначення оманливого контенту вживаються рідше, ніж “фейк”, “місінформація” та “дезінформація”. Однак дослідження цих понять є для нас теж релевантними.

Дипфейк — це технологія, що з'явилася у 2017 році й поєднує в собі елементи глибокого навчання та фейку: створює дуже переконливі маніпулятивні відео, синтезуючи кілька фотографій або відео людей, їхні рухи та емоції [7, с. 74].

У нашому дослідженні будемо спиратися на таке визначення фейкових новин, де головним є не злий умисел, а факт непідтвердженої, хибної інформації, адже для кінцевого користувача часто важливою є не причина недостовірності, а бажання споживати лише факти. Ми також не відкидаємо важливість інших дотичних термінів, адже вони є в літературі про фейкові новини.

1.3 Методи автоматичного виявлення фейкових новин

С. Чжоу та ін. пропонують поділити методи розпізнавання фейків на чотири категорії: на основі знання (*knowledge-based*), стилю (*style-based*), способу поширення (*propagation-based*) та джерела (*source-based*) [103, с. 7]. Хоча в методах на основі способу поширення та джерела використовується подекуди АОПМ, під інтересом дослідників з цієї сфери знаходяться головним чином методи на основі стилю та знання — спробуємо описати їх в наступних підрозділах.

Методи виявлення фейкових новин **на основі способу поширення** (синоніми: аналіз мережі, *network analysis*) полягають у шаблонах, якими інформація поширюється в мережах, аби виявити характерні для дезінформації закономірності [103, с. 20]. Такі методи можуть базуватися, наприклад, на тому, що статті з авторитетних джерел часто містять дезінформаційні наративи, коли вони поширюються разом з фейковими новинами; така одночасна публікація призводить до ширшого розповсюдження й потенційно більшого впливу дезінформації в публічному дискурсі [62, с. 20].

Одним з прикладів розв’язань на основі аналізу мережі є фреймворк SAFER [55]. Система SAFER моделює складні взаємодії та поведінку в онлайн-спільнотах, аналізуючи обмін контентом і мережеві зв'язки між користувачами за допомогою архітектур GCN (*graph convolution network*) та GAT (*graph attention network*), розширюючи можливості виявлення фейкових новин за межі аналізу стилю тексту шляхом інтеграції соціального контексту з текстовими даними [55, с. 4].

Іншим прикладом є технологія на основі геометричного глибокого навчання [47]. Цей метод бере до уваги архітектуру соціальних мереж і те, як новини поширюються в них, щоб розрізняти справжні та фейкові новини, застосовуючи методи глибокого навчання на основі графів, що враховують соціальний контекст і мережеву структуру поширення новин [47, с. 2].

Методи **на основі джерела** (їх ще називають мережевими підходами (англ. *network approaches* [79]) часто використовують метадані, як-от URL-адреса, автор, вподобання в соціальних мережах тощо, для аналізу того, наскільки джерело є надійним [79, с. 3].

1.3.1 Методи на основі перевірки фактів

Перевірка фактів (фактчекінг, англ. *fact-checking*) початково зародилася в журналістиці й полягає в оцінці достовірності новини, порівнюючи витягнуті з контенту знання з відомими фактами [103, с. 8].

Традиційна, або ж мануальна, перевірка фактів передбачає залучення невеликої групи експертів для перевірки змісту новин та створення інформаційних ресурсів (фактчекінгових вебсайтів). Це є точним, але дорогим методом, особливо враховуючи збільшення обсягу контенту [103, с. 8]. Прикладами фактчекінгових ресурсів англійською мовою є Associated Press Fact Check [26], FactCheck.org [44], PolitiFact [72], Reuters Fact Check [45], The Washington Post Fact Checker [43] та ін. В Україні одним з таких ресурсів є проєкт “Реєстр фейків України”, котрий є ініціативою Ірпінської організації НСЖУ і де можна знайти перелік “інфосмітників” та посилання на фейкові новини [20]. Ще одним прикладом є вебсайт StopFake, створений організацією “Центр Медіареформи”, на якому розвінчуються фейкові новини [83]. Також “Вокс Україна” має свою фактчекінгову сторінку VoxCheck [98]. Фейки й наративи досліджує і громадська організація “Детектор медіа” [19]. Ще одним прикладом є рубрика “Боротьба з фейками та цифрова гігієна” на ресурсі DeepState, де волонтери спростовують фейкову інформацію про російсько-українську війну [2].

Перевірити текст вручну може кожна людина. Якщо користувач помітив підозрілу інформацію, варто перевірити її автора (якщо це профіль у соцмережі — чи є у нього справжні друзі та чи не є він ботом) [13, с. 244].

Автоматична перевірка фактів складається з трьох етапів: виявлення тверджень (англ. *claim detection*) — тобто розпізнавання інформації, яку треба перевірити; збір доказів (англ. *evidence retrieval*), що полягає в пошуку ресурсів, щоб підтвердити або спростувати твердження; перевірка твердження (англ. *claim verification*) — оцінювання правдивості твердження на основі зібраних ресурсів [56, с. 179]. Розглянемо кожен із цих етапів.

І. Виявлення тверджень. На цьому етапі відбувається екстракція всіх тверджень, котрі треба перевірити. У новинному тексті не всі речення включають в цей процес, а лише ті, що містять важливу інформацію і щодо яких широка публіка хотіла б дізнатися правду. Наприклад, твердження “понад шість мільйонів американців захворіли на COVID-19 у січні” варте перевірки більше, ніж “вода мокра” [56, с. 179–180].

У фактчекінгових системах, що працюють на основі АОПМ, подібні важливі твердження представлені як набір іменованих сутностей, що пов'язані між собою. Іменовані сутності — це термін на позначення власних назв (імена людей, назви організацій), однак його часто вживають для ширшого кола понять [59, с. 162]. Сюди можуть також входити номери телефонів, дати, грошові значення тощо — тобто слова та сполучення, що містять конкретну інформацію. Розпізнавання іменованих сутностей (англ. *named entity recognition*) полягає в приписуванні словам та сполученням спеціальних тегів на зразок PERSON або LOCATION [59, с. 162].

На цьому ж етапі встановлюються семантичні зв’язки між іменованими сутностями — завдання, відоме як екстракція відношень [59, с. 415]. Такі відношення можуть бути кількох типів (див. *рис. 1.1*). Багато прикладів пов’язані розташуванням у просторі, належністю, ієрархічною організацією (гіпер- і гіпонімія), впливом. Наприклад, у медійних текстах дуже часто зустрічається відношення “частина-ціле” [ORG — ORG] [59, с. 416–417].

Іншим прикладом семантичних зв'язків є так звані семантичні трійки (англ. *semantic triples*), що складаються з трьох компонентів: суб'єкта, предиката та об'єкта. [100]. У літературі зустрічаються також терміни SPO (*Subject — Predicate — Object*) RDF triple (*Resource Description Framework*) [59, с. 417]. Наприклад, з речення "Steve Jobs is an entrepreneur" можна витягнути таку семантичну трійку: *Steve Jobs* (суб'єкт) *type* (предикат) *entrepreneur* (підприємець) [85]. Прикладом семантичної трійки з відношенням розташування в просторі є *Golden Gate Park* (суб'єкт) *location* (предикат) *San Francisco* (об'єкт) [59, с. 417]. Звісно, семантичні трійки, які репрезентують твердження з новин, зазвичай утворені словами, що належать до суспільно-політичного дискурсу. Наприклад, *DonaldTrump profession President* [103, с. 10].

Семантичні трійки тісно пов'язані не так з формальним синтаксисом (підмет — присудок — другорядні члени), як із семантичним. Звідси й береться позначення трійки як семантичного зв'язку, а не граматичного. В українській мові просте речення з погляду семантично-синтаксичної організації формується одним предикатом, а суб'єкт та об'єкт визначаються його валентностями; отже, предикат є головним компонентом речення [3, с. 57].

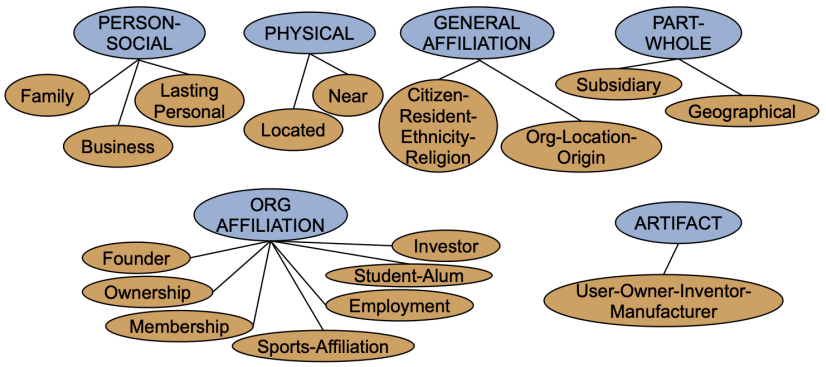


Рисунок 1.1. Типи відношень в завданні ACE [59, с. 416]

II. Збір доказів. На цьому етапі збираються факти з Інтернету, котрі потребують подальшої обробки та розміщення їх у базі знань або у графі знань [103, с. 11]. База знань — це сукупність відомостей “про можливості та способи використання мовних об’єктів у різних ситуаціях спілкування, у різних продуктах мовної діяльності, судження про такі об’єкти тощо” [6, с. 205]. Граф знань — це різновид бази знань, і він має вигляд семантичної мережі. Семантичні мережі складаються з дуг та вузлів: вузли — це точки, а дуги позначають відношення між ними [6, с. 229]. У графі знань вузлами є сутності (як-от імена людей, назви організацій, події, місця), а дугами слугують предикати [103, с. 11].

Звертаємо увагу на те, що цей крок не обов’язково виконується послідовно, після етапу виявлення тверджень. Ще перед першим етапом можна знайти готову базу або граф знань, як-от DBpedia [35], KBpedia [51], Wikidata [101], YAGO [102] тощо.

Для створення графів знань українською мовою для завдання перевірки фактів потенційні дослідники можуть використати новини з перевірених джерел. Згідно з Інститутом масової інформації, такими ресурсами є “Суспільне”, “Радіо Свобода”, “Українська правда”, “Бабель” та ін. [1].

Також потенціал для створення бази знань українською мовою має ресурс “Вікіновини” [4]. Хоча це джерело не належить до офіційних медіа і створювати статті на ньому може кожен, матеріали, як правило, перевіряються на достовірність редакторами-волонтерами перед публікацією.

III. Перевірка тверджень. Для оцінки достовірності твердження необхідно зіставити інформацію, отриману з новинних матеріалів, з доведеними фактами [103, с. 12]. Якщо семантична трійка існує у графі або вона семантично близька до наявних в ньому тверджень, то її можна вважати фактом. Менш простолінійним є сценарій, коли твердження відсутнє у графі чи базі. Щодо таких випадків С. Чжоу та ін. підсумовують три можливі погляди на достовірність [103, с. 12]:

- твердження вважається хибним;

- достовірність твердження є невідомою;
- якщо для суб'єкта та предиката у графі знань є приклади трійок, однак оригінальна трійка відсутня, то таке твердження є хибним. Утім, якщо у графі взагалі немає трійок для суб'єкта та предиката, то достовірність є невідомою.

Серед англійськомовних досліджень цікавий підхід до перевірки фактів за допомогою баз знань пропонують М. Маянк та ін., розробивши систему DEAP-FAKED [65]. У графі знань представлені дані з різних джерел у вигляді трійок *head*, *relation*, *tail* (*h, r, t*), де *h i t* — вершини, що представляють сутності, а *r* — відношення між ними [65, с. 47]. У запропонованій системі використано заголовки новин, а не повні тексти статей [65, с. 48]. На наше переконання, такий підхід є особливо цікавим з того погляду, що він відповідає реальним ситуаціям, коли рішення про правдивість новин часто потрібно приймати швидко й на основі обмеженої інформації.

Перевірка фактів також може здійснюватися на основі семантичної схожості вхідної новини з пов'язаними текстами з надійних джерел; такий метод у літературі називається перехресною перевіркою (англ. *cross-checking*) [22, с. 1].

Одним з методів вимірювання семантичної близькості новини з текстом з надійних джерел є використання векторної семантики та техніки вбудовування слів (англ. *word embeddings*). векторна семантика передбачає репрезентацію слів як точок у багатовимірному семантичному просторі, де вбудовування слів (вектори) слугують способом представлення цих лексичних одиниць [59, с. 109]. Два слова, речення або тексти вважаються семантично близькими, якщо їхні вектори подібні. Існує кілька методів для визначення подібності текстів, серед яких найпопулярнішими є:

- скалярний добуток (англ. *dot product*) [59, с. 114]:

$$dot\ product\ (\mathbf{v}, \mathbf{w}) = \mathbf{v} \cdot \mathbf{w} = \sum_{i=1}^{N} v_i w_i + v_1 w_1 + v_2 w_2 + \ldots + v_N w_N$$

- косинус подібності (англ. *cosine similarity*) [59, с. 115]:

$$cosine(\mathbf{v}, \mathbf{w}) = \frac{\mathbf{v}\cdot\mathbf{w}}{|\mathbf{v}|\cdot|\mathbf{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

Порівнюючи ці дві метрики, зазначимо, що, згідно з формулою, скалярний добуток вищий, якщо вектор довший, з більшими значеннями в кожному вимірі. Це означає, що скалярний добуток надає перевагу векторам з більшою довжиною; косинус подібності нормалізує вектори та враховує лише кут між ними [59, с. 114]. При перехресній перевірці дослідники керуються припущенням, що семантична подібність між кластером новин з надійних джерел та новиною, яку треба буде перевірити, є низькою, якщо остання є фейковою.

Фактчекінгові моделі можна покращувати, впроваджуючи обґрунтування та докази, котрі людина використовує для визначення правдивості тверджень. Так зване "моделювання обґрунтувань" позначає залучення аргументації та доказів із фактчекінгових статей у моделі машинного навчання, щоб підвищити точність визначення правдивості тверджень: у поєднанні з такою аргументацією значно покращується розрізнення моделями правдивих і неправдивих тверджень [24, с. 86].

1.3.2 Методи на основі стилю

В основі розпізнавання фейків на основі стилю (це ще називають "лінгвістичним підходом" [79]) лежить припущення про те, що в конкретному медійному тексті є певні закономірності, що роблять текст фейкової новини більш переконливим [103, с. 5]. Згідно з цим припущенням, можна розпізнавати лише фейки у вузькому розумінні цього терміна, тобто ті, що були створені навмисно. У працях про виявлення фейків на основі стилю також часто вживається термін "стилометрія". У криміналістичній експертизі стилометрія допомагає відповісти на запитання "Хто є автором конкретного документа?" [23, с. 461].

З першого погляду ідентифікація в медійному тексті шаблонів, котрі б

вказували на неправдивість інформації, здається дуже складним завданням, особливо враховуючи те, що “фейковість” тексту намагаються приховати. Автори фейкових новин часто вдаються до імітації авторитетних новинних ресурсів, подаючи цитування начебто надійних джерел або скриншоти з цитатами відомих осіб [10, с. 30]. У таких текстах часто присутні екстралінгвістичні засоби, як-от зображення та відео, що посилює достовірність інформації [16, с. 48].

Разом з цим надмірне відсилання до фактів та реальних загальновживаних публіцистичних термінів може свідчити про “фейковість” новини. При семантичному аналізі лексики новини “Conspiracy Theorist Convinces Neil Armstrong Moon Landing Was Faked” було виявлено, що в тексті фрейм СФЕРА ПРАВДИ реалізується за допомогою вживання одиниць на зразок “press conference”, “YouTube”, “reporters”, “famed astronaut”, “moon landing”, “Apollo 11 mission”, “the Lunar Module”, “July 20, 1969” [16, с. 48].

Фейкові новини, хоч і часто мають посилання на авторитети, також типово містять інформацію сенсаційного характеру та підміну понять [12, с. 35], “вилучення певного контексту, дроблення цілісної картини інформації” [21, с. 276]. Надмірна емоційність виявляється в таких словах, як “сенсаційно”, “терміново”, “достовірне джерело” [21, с. 276]. Особливо у фейкових новинах, створених для дискредитації захисників України в контексті російсько-української війни, як вказують Н. Шульська та ін., можна зустріти вирази на кшталт “мені сказали військові”, “достовірна інформація”, “я брехати не буду” [21, с. 276]. Фейкові новини характеризуються занадто емоційним поданням, оскільки користувачі з негативним психологічним станом не будуть критично аналізувати таку інформацію [9, с. 80].

Окрім надмірної емоційності, сучасним фейковим новинам про російсько-українську війну притаманна і гіперболізація для підсилення ефекту сенсації, що виражається словесними маркерами: “великий ажіотаж”, “масово”, “дуже” [21, с. 277].

Деякі лінгвістичні маркери фейкових новин поверхнево згадуються в

дослідженнях феномену фейків у сфері комп'ютерної лінгвістики. Наприклад, Н. О'Браєн та ін. намагалися обійти недолік використання глибоких нейронних мереж для виявлення фейкових новин, що полягає у відсутності прозорості в процесі прийняття рішення моделлю; автори дослідження знайшли слова, які їхня згорткова нейронна мережа вважає найбільш "фейковими" і "справжніми", за допомогою зворотного перетворення (англ. *backpropagation*) результатів мережі. [88, с. 3]. Приклади перших п'яти слів з кожної категорії подаємо у вигляді *табл. 1.1*.

"Справжні" дієслова	"Фейкові" дієслова	"Справжні" іменники	"Фейкові" іменники	"Справжні" прикметники	"Фейкові" прикметники
adapting (адаптуючи)	breaking (розбиваючи)	adaptation (адаптація)	ambassador (посол)	aboriginal (аборигенний)	able (здатний)
aiming (націлюючись)	carrying (несучи)	aim (мета)	axis (вісь)	artistic (мистецький)	bipartisan (двопартійний)
appeared (з'явилися)	continue (продовжувати)	amazon (амазон)	bias (упередження)	chaotic (хаотичний)	covert (прихований)
backing (підтримуючи)	elect (вибирати)	apprentice (учень)	cause (причина)	disappointing (який розчаровує)	deep (глибокий)
campaigning (агітуючи)	fed (годували)	artist (художник)	combat (боротьба)	eighth (восьмий)	divine (божественний)

Таблиця 1.1

Слова, що часто вважають корисними
для класифікації справжніх і фейкових новин [88, с. 4].

Праць, котрі б описували лінгвістичні риси фейкових новин, надзвичайно мало. Однак, оскільки фейковий контент належить до ширшого поняття оманливого контенту, ми, імовірно, можемо накладати певні шаблони з такого виду даних на фейкові новини. У психології та кримінальній експертизі припущення про те, що тексти, що містять неправду, мають особливі лінгвістичні риси, називають гіпотезою Ундойча (англ. *Undeutsch Hypothesis*): люди, котрі вдаються до обману, часто демонструють поведінкові сигнали або лінгвістичні маркери, які видають цей обман, що є основою для розробки

методів оцінки достовірності тверджень [95].

У контексті фейкових новин гіпотеза Ундойча частково корелює з припущенням української дослідниці О. Грищенко про існування фейкової мовленнєвої особистості, котра володіє лінгвістичними, психологічними, психічними, етнічними та іншими характеристиками, що мають різний ступінь вербалізації [5, с. 41].

Серед лінгвістичних маркерів обману загалом в дослідженнях згадуються, зокрема, такі (на основі англійської мови):

- Використання великої кількості заперечень і більш узагальнювальних термінів (*always* — завжди*, never* — ніколи*, nobody* — ніхто) [99, с. 102, 112–14]. Реалізація цієї риси оманливих текстів присутня й у фейкових новинах українською мовою. Вислови на кшталт "за деякою інформацією" через занадто узагальнений характер вказують на те, що в тексті порушено стандарт достовірності [21, с. 276].

- Уживання меншої кількості емоційної лексики та більшої кількості дієслів руху [97, с. 608]. Варто зазначити, що це твердження дещо суперечить згадкам деяких дослідників про те, що фейкові новини мають занадто емоційний характер;

- Відсутність слів на зразок "exactly" (укр. "точно"), котрі вказують на прагматичну точність [66, с. 361–362].

Вищезгадані шаблони важко відстежувати вручну, однак у цьому можуть допомогти методи машинного навчання. Такі методи полягають у використанні складних математичних алгоритмів для вивчення шаблонів у текстових даних. Звертаємо увагу на те, що ідентифікацію фейкових новин можна також розуміти як віднесення тексту до однієї з двох категорій: фейкова новина або нефейкова. Отже, найпростіша версія такого завдання є, фактично, бінарною класифікацією тексту й належить до машинного навчання з вчителем. У навчанні з вчителем набір даних складається з вхідних спостережень, кожне з яких пов'язане з відповідним правильним виходом (класом); завдання такого навчання полягає в тому, щоб прийняти вхідні спостереження x та фіксований набір вихідних

класів $Y = \{y_1, y_2, \ldots, y_M\}$ та повернути $y \in Y$. [59, 61].

У контексті фейкових новин таке навчання передбачає використання анотованого набору даних, що містить як справжні, так і фейкові новини, та застосування алгоритмів для визначення шаблонів серед цих типів контенту. Вивчені закономірності потім формують модель, котра має на меті передбачити, чи вхідні новини є фейковими чи справжніми [58, с. 1–2].

Оскільки виявлення фейкових новин є бінарною класифікацією тексту, для розв'язання цього завдання застосовуються ті ж самі методи, що й для інших подібних проблем (класифікація тексту як спам або не спам, визначення позитивного або негативного сентименту, виявлення токсичних коментарів тощо).

1.3.2.1 Статистичні методи

Наївний Байєс — це ймовірнісний класифікатор, що спирається на теорему Байєса з так званими наївними припущеннями щодо незалежності між ознаками; для документа d з усіх класів $c \in C$ класифікатор повертає той клас, що має максимальну апостеріорну ймовірність для даного документа, тобто класифікатор обчислює ймовірність кожного класу на основі вхідних ознак [59, с. 62].

Випадковий ліс — це "метаоцінка, що застосовує низку класифікаторів на основі дерев рішень на різних підвибірках набору даних та усереднює значення для підвищення точності прогнозування та контролю над перенавчанням" [81]. У цьому алгоритмі кожне окреме дерево рішень схоже на блок-схему або граф, у якому вузли — це тести на значення атрибута, гілки — результати тесту, а листя — передбачені класи [57, с. 18]. Візуалізацію дерева рішень представлено на *рис. 1.2*.

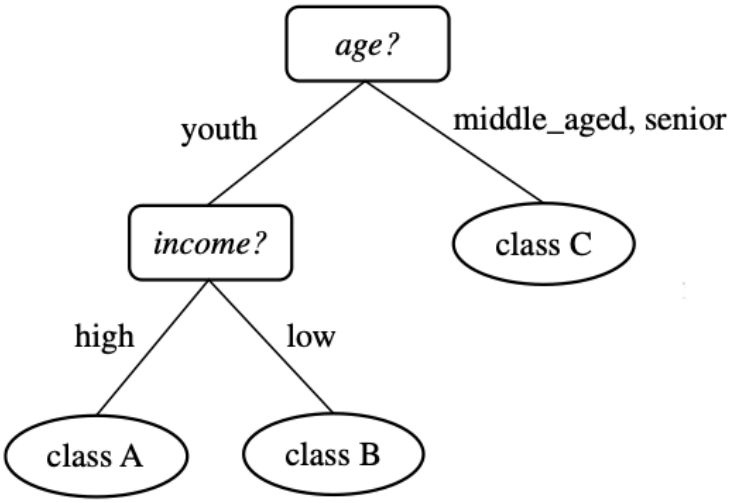


Рисунок 1.2. Дерево рішень [57, с. 18]

Логістична регресія — це розрізнювальна модель, котра передбачає ймовірність того, що задані вхідні спостереження належать до певного класу, і використовує сигмоїдну (логістичну) функцію (див. *рис. 1.3*). Сигмоїдна функція дає можливість зіставити будь-яке вихідне число зі значенням між 0 і 1 [59, с. 82].

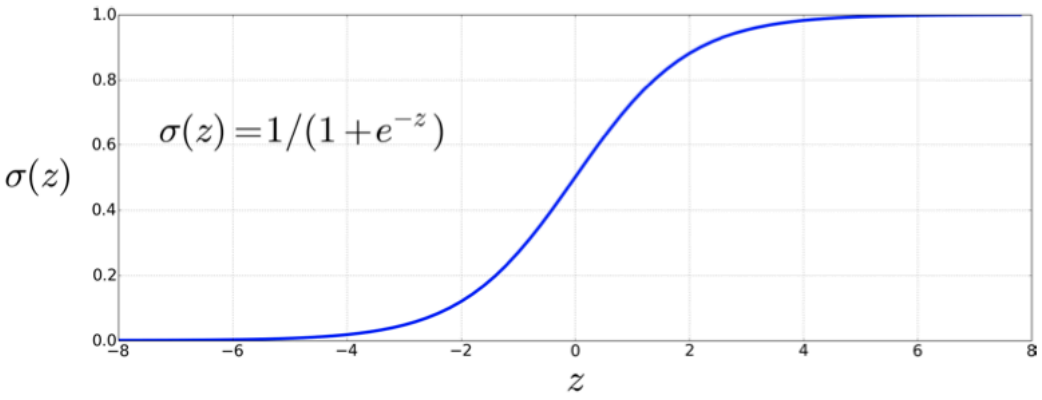


Рисунок 1.3. Сигмоїдна функція [59, с. 83]

У логістичній регресії модель вивчає набір зсувів (англ. *biases*) і ваг (англ. *weights*) навчальних даних, де кожна вага вказує на релевантність відповідної вхідної ознаки до результату класифікації, а зсув коригує поріг прийняття рішення [59, с. 83].

1.3.2.2 Методи глибокого навчання

Нейронні мережі — це мережі, котрі складаються з “невеликих обчислювальних блоків, кожен з яких отримує вектор вхідних значень і видає одне вихідне значення” [59, с. 136]. Ці обчислювальні блоки сполучені між собою зваженими зв’язками (англ. *weighted units*) [57, с. 19], що зображено на *рис. 1.4*. Сучасні нейронні мережі є глибокими (мають багато прихованих шарів) [59, с. 136]. В експериментах останніх років з класифікації фейкових новин, як і загалом у багатьох застосуваннях АОПМ, особливо часто використовують такі типи нейронних мереж: згорткові, рекурентні, мережі на основі трансформерів та довгої короткочасної пам’яті.

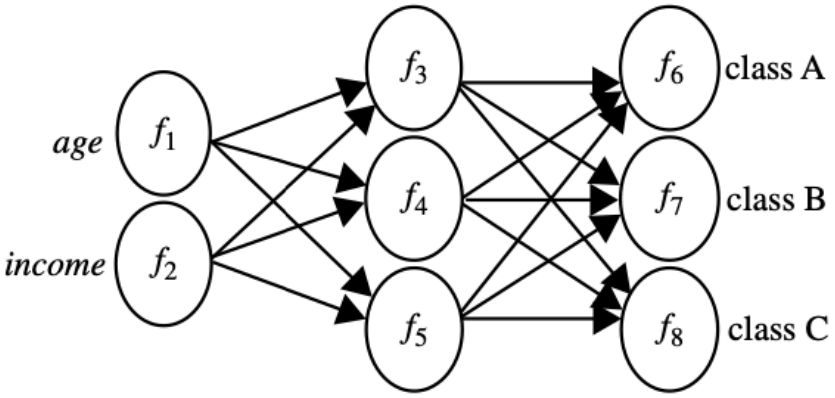


Рисунок 1.4. Спрощена будова нейронних мереж [57, с. 18]

Рекурентні нейронні мережі (англ. *recurrent neural network*) “містять цикл у своїх мережевих зв'язках” [59, с. 187], тобто вхідне значення кожної одиниці залежить від свого власного попереднього вихідного значення. Якщо говорити більш точно, прихований шар з попередньої ітерації грає роль динамічної пам'яті, що кодує минулу інформацію без обмежень щодо фіксованої довжини [59, с. 188].

На практиці рекурентні нейронні мережі погано справляються із завданнями, що вимагають використання інформації, віддаленої від поточної

точки обробки [59, с. 200]. Для того, що оминути цю ваду, на основі рекурентних нейронних мереж було створено архітектуру **довгої короткочасної пам'яті** (англ. *LSTM, long short-term memory*), котра вилучає з контексту непотрібну інформацію та додає ту, що може знадобитися в подальших прийняттях рішень [59, с. 200]. LSTM покращують традиційні архітектури рекурентних нейронних мереж через контекстний шар поряд з рекурентним прихованим шаром і використання спеціалізованих блоків (англ. *gates*), керованих додатковими вагами, для послідовного регулювання потоку інформації через вхід, попередній прихований шар і попередні контекстні шари [59, с. 200].

Згорткові нейронні мережі (англ. *convolutional neural network*) використовують згортку, за якої багатовимірні масиви даних обробляються за допомогою ядер (англ. *kernels*), котрі вибірково підкреслюють певні особливості в даних, що робить їх особливо ефективними для завдань класифікації [53, с. 327–238], що спрощено показано на *рис. 1.5*.

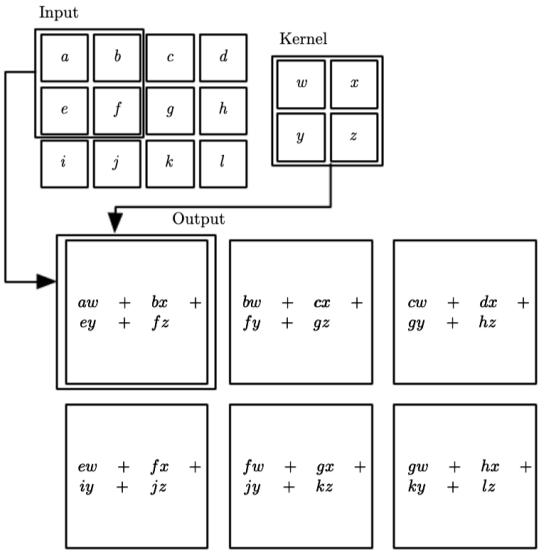


Рисунок 1.5. Спрощений алгоритм роботи згорткових нейронних мереж [53, с. 330]

Згорткові нейронні мережі традиційно використовувалися для обробки зображень. Утім, вони також можуть бути ефективними у завданнях з

класифікації тексту, що грає роль одновимірного зображення. Наприклад, П. Ґелорже та ін. описують гібридний підхід до класифікації новин: використовуючи алгоритм VAGO, який аналізує тексти на предмет нечіткості та суб'єктивності за допомогою семантичних правил та обробки природної мови, та класифікатор FAKE-CLF, що має в основі згорткову нейронну мережу для класифікації текстів як упереджених чи неупереджених [34]. Дослідження виявило позитивну кореляцію між нечіткістю/суб'єктивністю, визначеною VAGO, та класифікацією текстів щодо упередженості за допомогою FAKE-CLF, що свідчить про те, що суб’єктивні тексти з більшою імовірністю можуть бути класифіковані як упереджені [34, с. 5].

Навчання **моделей-трансформерів** відбувається за допомогою методу, котрий англійською називається *self-attention* і полягає в побудові контекстних репрезентацій значення слова, які витягують інформацію з навколишніх слів (контексту), допомагаючи моделі зрозуміти, як слова пов'язані одне з одним [59, с. 214]. Моделі на основі трансформерів перевершують ефективність інших типів нейронних мереж для багатьох мовних завдань і використовуються, зокрема, в таких інструментах, як ChatGPT.

Один з оглядів статей показав, що простих розв’язань на основі трансформерів достатньо для виявлення фейкових новин у порівнянні зі складнішими найсучаснішими технологіями [70]. *Табл. 1.2* показує F1-міру для трансформерних класифікаторів для різних новинних наборів даних англійською мовою.

	PolitiFact	GossipCop	Twitter15	Twitter16	Twitter15 T/F	Twitter16 T/F	WNUT-2020	Average Rank
SOTA	**92.8** [77]	85.0 [28]	**91.0** [30]	**92.4** [30]	82.5 [50]	75.9 [50]	**91.0** [43, 58]	4.4
CT-BERT	86.0 ± 3.2	90.6 ± 0.2	83.5 ± 2.8	83.9 ± 0.9	93.8 ± 1.6	94.0 ± 3.5	90.6	2.8
Funnel	86.4 ± 3.2	–[a]	66.9 ± 3.0	69.6 ± 2.9	83.2 ± 3.8	90.8 ± 2.2	88.5	–
RoBERTa	86.7 ± 1.2	**92.8 ± 0.5**	81.8 ± 1.5	84.8 ± 1.9	**94.4 ± 0.8**	**95.7 ± 2.8**	90.5	**2.3**
BERT	81.8 ± 3.0	89.8 ± 0.4	77.5 ± 3.3	78.2 ± 4.1	89.7 ± 1.6	91.6 ± 4.5	88.5	5.3
BERTweet	88.5 ± 1.2	92.6 ± 0.6	76.7 ± 2.9	77.7 ± 2.7	86.7 ± 1.8	92.0 ± 3.7	88.8	4.4
DeCLUTR	36.6 ± 1.4[b]	43.3 ± 0.4[b]	80.4 ± 2.6	80.5 ± 1.7	91.7 ± 1.5	94.5 ± 2.5	89.1	5.1
ELMo	83.1 ± 1.6	92.0 ± 0.5	53.7 ± 2.7	55.5 ± 4.9	74.4 ± 3.5	83.3 ± 5.0	82.4	8.0
ALBERT	80.1 ± 2.9	88.2 ± 0.9	63.4 ± 4.0	68.0 ± 3.5	83.3 ± 1.8	88.9 ± 4.3	86.8	7.7
BERT-tiny	85.3 ± 2.8	86.5 ± 0.6	54.6 ± 3.4	48.8 ± 3.9	77.8 ± 4.8	77.8 ± 4.9	79.9	8.9

Таблиця 1.2

F1-міра для моделей-трансформерів [70, с. 3436]

1.3.2.3 Попередня обробка вхідних даних

Моделі машинного навчання на вході мають отримати текст, представлений у числовому вигляді. На цьому етапі мають місце два процеси: обирання ознак (англ. *feature selection*) та вилучення (екстракція) ознак (англ. *feature extraction*). Обирання ознак передбачає вибір певних елементів з тексту, як-от біграми, розмічені частини мови, частотності або іменовані сутності; вилучення ознак створює нові дані з первинних мовних одиниць [55, с. 10].

Вхідні дані для навчання мовних моделей часто, а особливо в роботах, на які ми посилаємося в цьому дослідженні, представлені у вигляді векторних репрезентацій (вкладань слів). Один зі способів це зробити — використання TF-IDF, статистичного показника того, наскільки важливим є слово для документа в колекції документів. Оцінки TF та IDF перемножуються для кожного слова, у результаті отримуємо оцінку TF-IDF. Цей показник обчислюється для всіх слів у тексті, і він є вищим для тих одиниць, які є унікальними для документа. [59, с. 116-117]. Відомими також є моделі fastText, котра розглядає кожне слово як набір символів у вигляді n-грамм [41], та GloVe, що вбудовує слова, використовуючи процес, що називається матричною факторизацією [71].

Векторні репрезентації також можуть бути утворені за допомогою спеціальних нейронних мереж, як-от word2vec [40] та BERT (*Bidirectional Encoder Representations*) [30]. Обидві моделі були створені в компанії Google.

Перед векторним перетворенням тексту відбувається його обробка. Деякі системи виключають стоп-слова — надзвичайно часті одиниці, як-от займенники, прислівники, прийменники, в англійській мові — артиклі "the" та "a" тощо. Це роблять або за частотою в навчальній базі (визначивши перші 10-100 найчастіших токенів як стоп-слова), або використовуючи один із численних готових списків стоп-слів, доступних в Інтернеті [59, с. 65].

1.3.2.4 Приклади досліджень на основі стилю

За останнє десятиліття в англомовному науковому просторі було здійснено чимало спроб класифікації новин як фейкових або справжніх за допомогою методів машинного навчання.

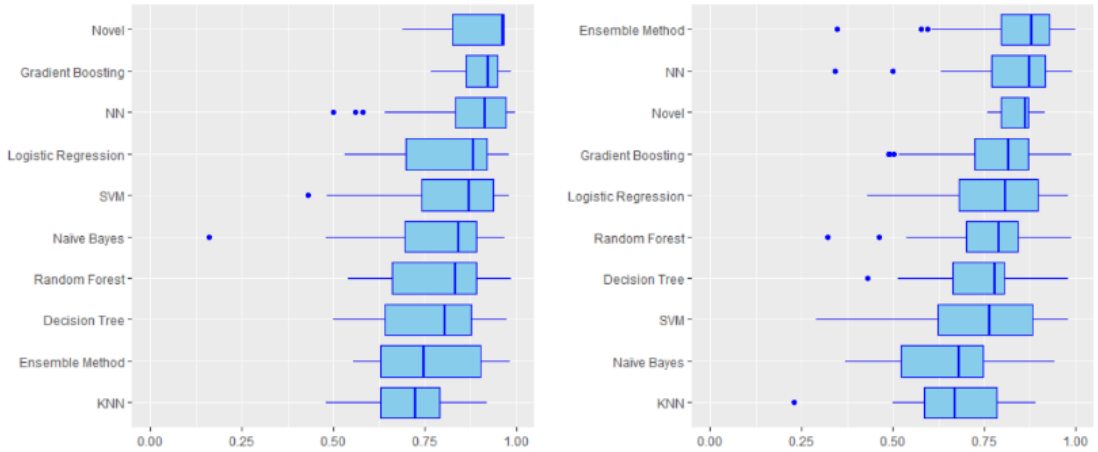
С. Афроз та ін. у 2012 [23] та М. Сірінг та ін. у 2016 [80] досягають хороших результатів у виявленні обману в тексті за допомогою методу опорних векторів. М. А. Спаленца та ін. (2021) у своєму дослідженні попередньо репрезентують текстові дані у вигляді іменованих сутностей та частин мови, а потім порівнюють кілька моделей-класифікаторів: метод опорних векторів, випадковий ліс, градієнтне підсилення, градієнтну мережу WiSARD [60].

С. Волкова та ін. (2017) представляють лінгвістичну модель на основі таких типів нейронних мереж, як рекурентні та згорткові [76]. Н. О'Браєн та ін. (2018) використовують згорткову нейронну мережу, у якій першим шаром є попередньо навчене вкладення word2vec [88]. Л. Борхес та ін. запропонували метод на основі двосторонньої рекурентної мережі для виявлення політичних позицій [32]. К. Шу та ін. (2020) використовують модель на основі LSTM для виявлення фейкових новин та шкідливих облікових записів [49]. Ч.–М. Цай (2023) пропонує гібридний метод на основі поєднання класифікації тексту з розпізнаванням іменованих сутностей [92].

Деякі з досліджень беруть до уваги не лише текст новинної статті, а й заголовок. О. Нолан та ін. класифікують зв'язок між заголовком і текстом новини (*agree* — узгодженість, *disagree* — неузгодженість, *discuss* — обговорення, *unrelated* — не пов'язані), що допомагає в автоматизованому виявленні фейкових новин [69, с. 1].

У згаданих дослідженнях різняться метрики та бенчмарки оцінки ефективності рішень. Однак відомо також про спроби уніфікувати спосіб оцінки готових розв'язань. Наприклад, Н. Гой та ін. роблять порівняння робіт, опублікованих з 2016 по 2020 рік; цей період обраний через підвищений інтерес до вивчення фейкових новин після президентських виборів 2016 року [58, с. 4]. Підсумок порівняння показано на *граф. 1.1*, де бачимо високі показники для

нейронних мереж та ансамблевих розв’язань.



Графік 1.1. Точність та F-міра моделей машинного навчання для виявлення фейкових новин [58, с. 17]

Як бачимо, багато дослідників по всьому світу працюють над розробкою рішень для виявлення фейкових новин, але все ще тривають значні дискусії щодо найбільш ефективного архітектурного підходу до цього завдання. Окрім того, науковці намагаються розв’язати такі проблеми, як адаптація моделей до нових трендів у медіа, відсутність вичерпної кількості даних різними мовами та спільного бенчмарку для порівняння готових розв’язань.

У своєму дослідженні В. О. Джуніор та ін. висвітлюють виклики перевірки правдивості інформації в контексті пандемії COVID-19: розробивши нейромережу, що використовує техніку збільшення даних (англ. *data augmentation*), автори прагнуть підвищити точність виявлення фейкових новин португальською мовою [90, с. 3]. Цю ж саму техніку застосовують В. Маслей-Крешнякова та ін. [64].

Публікація Ч. Й. Пак та ін. описує складнощі та нюанси інформаційних маніпуляцій у контексті російсько-української війни [33]. У ній представлено новий датасет VoynaSlov російською мовою, котрий містить понад 38 мільйонів дописів російських ЗМІ у мережах Twitter та ВКонтакті [33, с. 5210]. Ці дані використовуються для аналізу того, як різні медіастратегії, як-от формування порядку денного (англ. *agenda setting*), фреймінг (англ. *framing*) і праймінг

(англ. *priming*), застосовуються для маніпулювання громадською думкою [33, с. 5209].

А. Рані та ін. (2023) представляють модель, котра використовує тонко налаштовані попередньо навчені мовні моделі, як-от T5 і RoBERTa, спеціально пристосовані для завдань ідентифікації обману [77]. Водночас дослідники презентують датасет SEPSIS, котрий створено на основі новин з Twitter-сторінки газети Times of India; ці дані анотувалися відповідно до типу упущення інформації, "кольору" брехні (чорний, білий, сірий, червоний), мети брехні та її тематики [77, с. 4–5]. У тому ж самому 2023 році виходить публікація [61], присвячена розробці передових методів автоматичного виявлення та класифікації оманливого контенту, який присутній у фейкових новинах. У ній представлено нову таксономію оманливого контенту, схему анотації для завдань з обробки природної мови та набір даних MAFALDA [61, с. 6], який можна використовувати для порівняння здатності різних мовних моделей виявляти обман.

Висновки до першого розділу

Сьогодні існує гостра потреба в ефективних стратегіях виявлення фейкових новин. Окрім очевидних соціально-політичних причин такої потреби, надалі існує неузгодженість серед дослідників щодо визначення поняття "фейк". У нашому дослідженні ми будемо використовувати термін "фейкові новини" у значенні несправжніх новин, незалежно від наміру. Цей підхід дасть змогу ширше застосовувати методи виявлення такого контенту й охоплювати всі його форми. Таке визначення особливо корисне у випадках, коли дезінформація може поширюватися ненавмисно через соціум, без початкового злого наміру.

Основні методи ідентифікації фейкових новин засновані на перевірці фактів, аналізі стилю й методів поширення та аналізі джерел. Особливо популярними та широко досліджуваними є методи на основі стилю тексту, що поєднують статистичні методи та глибоке навчання. Статистичні методи, як-от наївний Байєс або логістична регресія, все ще залишаються популярними через

простоту застосування, однак існування щоразу більшої кількості неструктурованих даних призводить до ширшого застосування методів глибокого навчання, що представлені нейронними мережами. Такі мережі здатні виявляти найскладніші шаблони, котрі містяться у текстах фейкових новин.

В англійськомовному науковому світі автоматизація розпізнавання фейкових новин за допомогою методів обробки природної мови є уже досить дослідженим явищем. Що стосується вітчизняних дослідників, можна виокремити таких авторів, як Т.П. Дяк та ін. (2022), А. І. Санжаровський та В. Я. Юрчишин (2023), однак загалом проблема розпізнавання українськомовних фейкових новин є малодослідженою.

РОЗДІЛ 2. СТВОРЕННЯ КОРПУСУ УКРАЇНСЬКОМОВНИХ НОВИННИХ ТЕКСТІВ

Для створення мовної моделі для класифікації тексту новини як “фейкового” чи “справжнього” потребуємо великої кількості анотованих медійних текстів, що належать до одного з двох класів. На основі цих даних зможемо навчити модель, котра у процесі тренування буде визначати риси, що відрізняють фейкові новини від справжніх. У цьому розділі описуємо створення такого мовного корпусу та труднощі, котрі виникли в процесі. Першу частину цього корпусу становлять уже готові дані, котрі знаходяться у вільному доступі (див. п. 2.1), другу ж створюємо самостійно (див. п. 2.2).

2.1 Використання готових наборів даних

Використання доступних текстів допоможе заощадити час та ресурси, котрі були б витрачені на збір та анотацію таких даних з нуля. На сьогодні створено вже достатньо велику кількість мовних корпусів та наборів даних для досліджень фейкових новин англійською мовою, як от BaIT [69], CREDBANK [68], FakeNewsNet [49], MAFALDA [61], SEPSIS [77] та ін.

Для української мови наборів даних з фейковими новинами все ще дуже мало, і це зазвичай аматорські або студентські проєкти. Одним з таких прикладів є датасет “Ukrainian News” на ресурсі для машинного навчання Kaggle, що містить 10700 новинних текстів, з яких 2498 — фейкові новини [94]. Недоліком цього набору даних є те, що новини, позначені як справжні (*True*), не завжди є такими, а віднесені до цієї категорії на основі надійності джерела, з якого вони походять. Наприклад, автор визначає Telegram-канали “Perepichka NEWS” та “NR” як надійні джерела [94], однак насправді в них також можуть бути опубліковані фейки, котрі важко вилучити без мануальної перевірки фактів.

Для нашого експерименту якості цих даних може виявитися достатньо: по-перше, ми дуже обмежені у виборі текстів українською мовою; по-друге, більшість справжніх новин, імовірно, є такими, а тому модель все ще зможе

вивчити риси, котрі вирізнятимуть такі тексти.

Перед тим, як долучити цей датасет до наших даних, його варто збалансувати: більшість текстів належать до категорії “True” (справжні новини), що може призвести до класового дисбалансу, коли модель буде приписувати більшість елементів до класу “справжні новини”. Для цього ми написали функцію *balance_data()*, яку зображено на *рис. 2.1*.

```
def balance_data(df):
    """Балансування даних у фреймі"""
    label_col = 'Label' if 'Label' in df.columns else 'label'
    fake_label = False if 'Label' in df.columns else 'Fake'
    real_label = True if 'Label' in df.columns else 'Real'

    try:
        num_fake = (df['Label'] == False).sum()
    except: num_fake = (df['label'] == "Fake").sum()
    try:
        num_real = (df['Label'] == True).sum()
    except:
        num_real = (df['label'] == "Real").sum()


    if num_real > num_fake:
        real_indices = df[df[label_col] == real_label].index
        sampled_real_indices = pd.DataFrame(real_indices).sample(n=num_fake,
random_state=42).index
        print(sampled_real_indices)
        df_balanced = df.loc[sampled_real_indices.union(df[df[label_col] == fake_label].index)]
    else:
        df_balanced = df
    return df_balanced
```

Рис. 2.1. Збалансування кількості текстів у фреймі даних

Ще одним способом отримати дані для середньо- та малоресурсних мов є переклад датасетів з англійської. Доповнення даних за допомогою автоматичного перекладу текстів з цієї мови значно покращує сентимент-аналіз французькомовних “твітів”, розширюючи навчальний набір даних за межі того, що зазвичай доступне мовою оригіналу [27, с. 269]. Певні результати показав і переклад на урду та гінді, хоча це й призвело до втрати мовних нюансів і культурно-етнічного забарвлення та появи упереджень, що пояснюється більшою віддаленістю цих мов від англійської [87 с. 124487].

На вебсайті Kaggle є кілька великих наборів даних, що містять фейкові та справжні новини, як-от “Fake News” [48], “Fake News Detection” [46], “Getting Real about Fake News” [50]. Для нашого експерименту ми обрали датасет “Disinformation and Fake News”, котрий зосереджений на випадках дезінформації з прокремлівських ЗМІ, які поширюються в ЄС та країнах Східного партнерства [37]; такі дані корисні для нас з погляду актуальності нашого дослідження щодо превенції поширення фейків, створених росією. Ми рандомізували ці дані та обрали з них 10% текстів (736).

Наступним кроком був вибір інструменту для автоматичного перекладу текстів з англійської мови на українську. Для мови Python є кілька можливих розв’язань з відкритим доступом. Одним із них є використання бібліотеки translate [91] (*рис. 2.2*). Під час тестування ми виявили, що вона має обмеження у 500 символів, а також використовує скрейпінг, через що ми не завжди мали доступ до перекладу.

```
from translate import Translator
translator = Translator(to_lang="uk")
for t in sample_texts:
    translation = translator.translate(t[:500])
```

Рисунок 2.2. Використання бібліотеки translate

Досить ефективними для перекладу є великі моделі, як-от SeamlessM4T від компанії Meta, котрі можна знайти у відкритому доступі на платформі Hugging Face [42]. Протестувавши її, ми виявили, що понад 50% усіх перекладів становлять тексти російською, попри те, що при використанні моделі треба вказувати мову перекладу як аргумент (див. *рис. 2.3*). Наприклад, речення “Protests in Hong Kong are following the Maidan scenario…” SeamlessM4T перекладає як “Протести в Гонконге следуют сценарию Майдана…”.

```
from transformers import pipeline
translator = pipeline("translation_en_to_uk", model="facebook/seamless-m4t-v2-large")
english_text = "Protests in Hong Kong are following the Maidan scenario..."
translated_text = translator(english_text)
```

Рисунок 2.3. Використання моделі SeamlessM4T

Популярною є також бібліотека googletrans — неофіційний клієнт для комунікації з Google Translate, котрий взаємодіє з поточною вебверсією перекладача [54]. У *табл. 2.1* показуємо приклади перекладів.

Текст оригіналу	**Текст перекладу**
The Ukrainian government is fascist and it collaborates with corporate pro-fascist western oligarchs and with military industry of the US.	Український уряд є фашистським, і він співпрацює з корпоративними профашистськими західними олігархами та з військовою промисловістю США.
The new president of Ukraine, Volodymyr Zelenskyi, should dissolve the Verkhovna Rada…	Новий президент України, Володимір Зеленський, повинен розчинити Верховну Раду України…
Despite all the Russophobia and anti-Russian hysteria, Russia is the saviour of mankind.	Незважаючи на всю расофобію та антиросійську істерику, Росія є Спасителем людства.

Таблиця 2.1.

Переклади, виконані за допомогою бібліотеки googletrans

Попри помилки, особливо при передачі власних назв, ми саме обрали цю бібліотеку, оскільки вона перекладає тексти правильною мовою та зберігає більшість ключових слів, а також, оскільки використовує зовнішній програмний інтерфейс (API), не вимагає великої кількості пам’яті RAM — на противагу великим моделям, котрі треба завантажувати локально.

Отже, значну частину нашого корпусу становлять тексти з датасету “Ukrainian News” (4913 текстів) та перекладені за допомогою бібліотеки googletrans на українську з англійської дані з набору “Disinformation and Fake News” (736 текстів).

2.2 Збирання текстів новин з відкритих джерел

Недоліком описаних у п. 2.1 готових рішень є їхня актуальність. У датасеті “Ukrainian News” тексти датуються від 4 лютого до 11 грудня 2022 року [94]. Набір англійською мовою містить дані до березня 2020 року [37]. Оскільки ми у цій роботі зосереджуємося в основному на виявленні фейкових новин щодо тривалої агресії росії проти України, а теми фейкових новин в пропагандистських текстах з часом змінюються, варто також зібрати тексти, котрі б охоплювали 2023 рік та початок 2024.

Для того, щоб знайти колекції фейкових новин в українськомовному медійному просторі, ми використовуємо фактченкінгові сайти. Одним з таких ресурсів є рубрика “Інфосмітники України” на сайті “Реєстр фейків України” [20]. Тут можна знайти список ненадійних джерел з українськими доменними адресами. Проаналізувавши список, ми вибрали ті ресурси, котрі ще не були видалені та до котрих можна отримати доступ, висилаючи запити з локального сервера, а також публікували новини у 2023 та 2024 році: “Puer news 24” [73], “Новини сьогодення” [15], “Україна Live” [17].

Для отримання даних з цих ресурсів ми використовуємо вебскрейпінг: “створення агента для автоматизованого завантаження, аналізу та організації даних з Інтернету” [96, с. 3]. Оскільки зазначені вище ненадійні ресурси містять новини у вигляді HTML-сторінок, ми можемо використовувати бібліотеку Beautiful Soup для мови Python [29]. Цей модуль дозволяє створювати спеціальний об’єкт BeautifulSoup та ітерувати за HTML-тегами, щоб знайти потрібний нам контент [96, с. 62]. Наприклад, для того, щоб “відкрити” окремі новинні статті на сайті “Новини сьогодення”, ми шукаємо всі заголовки “h4” з класом “entry-title” та знаходимо тег “a” (див. *рис. 2.4*).

```
def scrape_news_page(url, page_number, writer) -> bool:
    """Переходить за всіма статтями на конкретній сторінці"""

    print(f"Going to page {page_number}...")
    try:
        response = requests.get(url)
        if response.status_code == 200:
            soup = BeautifulSoup(response.text, "html.parser")

            article_tags = soup.find_all("article")
            post_ids = []
            for t in article_tags:
                post_ids.append(t.get("id").split("-")[-1])

            for i in post_ids:
                post_url = "https://ukr-live.com/news/" + i
                process_individual_news(url=post_url, writer=writer)
            return True
        else:
            print(f"Failed to fetch the page: {page_number}")
            return False
    except: return False
```

Рисунок 2.4. Використання бібліотеки Beautiful Soup

Крім цього, ми використали ще два фактчекінгові ресурси: “StopFake” [83] та “VoxCheck” [98]. Нам вдалося надсилати запити до сервісу “StopFake”, щоб отримувати посилання на оригінальні фейкові новини. Що ж до останнього, то ми не змогли витягувати з нього фейкові новини автоматично через захист від стороннього трафіку, але нам вдалося зібрати дані вручну. На жаль, більшість із них були російською мовою, опубліковані на пропагандистських сайтах, тож ми знову скористалися бібліотекою googletrans, щоб перекласти ці тексти на українську мову.

Останньою частиною нашого набору даних є “справжні” новини за останні два роки. Для відбору надійних джерел ми посилаємося на ресурси на зразок “Суспільного”, “Радіо Свободи” та “Української правди”. На жаль, офіційні сайти цих медіа часто захищені фаєрволом або не мають навігації за сторінками, тож більшість таких текстів становлять дані з їхніх Telegram-каналів, отримані за допомогою бібліотеки Telethon [86]. Завдяки їй можна під’єднатися до окремого каналу та зчитати з нього необхідні дописи (*рис. 2.5*).

```python
def fetch_posts(channel_username):
    """Зчитує кожен 5-й допис з каналу"""
    with open("real.csv", mode="a", newline="", encoding="utf-8") as file:
        writer = csv.DictWriter(file, fieldnames=["title_ukr", "url", "text", "date"])
        with client:
            channel = client.get_entity(channel_username)
            count = 0
            gathered = 0
            for message in client.iter_messages(channel):
                time.sleep(0.06)
                count += 1

                if count % 5 == 0 and len(message.text) >= 10:
                    gathered += 1
                    print(gathered)
                    writer.writerow({"title_ukr": extract_first_sentence(message.text),
                                     "url": f"https://t.me/{channel_username}/{message.id}",
                                     "text": message.text,
                                     "date": message.date,
                                     })
                    if gathered >= 700:
                        break
```

Рисунок 2.5. Використання бібліотеки Telethon для зчитування дописів з Telegram-каналів

Утім, частину текстів, опублікованих згаданими надійними джерелами, нам вдалося зібрати вручну, завантаживши статті у форматі HTML та зробивши екстракцію тексту та заголовку з локальних файлів.

Проаналізувавши тексти, ми виявили, що багато з них містять емоджі, посилання на інші сайти, синтаксис у форматі Markdown, зайві фрази, як-от “Підписуйтеся”. Оскільки ці елементи є дуже загальними та створюють шум для мовних моделей, варто прибрати їх з текстів під час попередньої обробки.

Для цього ми створили модуль постобробки мовою Python, котрий містить такі методи: *escape_markdown_links()*, *remove_emoji()*, *remove_empty_lines()*, *remove_links()*, *remove_trash_phrases()*, котрі прибирають зайві елементи з текстів.

Висновки до другого розділу

У цьому розділі ми описали процес створення корпусу з українськими новинними текстами, що відносяться до одного з двох класів: фейкові та

справжні новини. Для збору даних ми використали балансування готових датасетів, переклад текстів з англійської та російської мови, вебскрейпінг, вилучення зайвих елементів.

Хоча в Інтернеті у вільному доступі є досить багато даних для навчання моделей для класифікації новин англійською мовою, для української мови їх майже немає. Створити такі датасети допоможуть фактчекінгові сайти, як-от “StopFake”, “VoxCheck”, “Реєстр фейків України”.

Отже, ми отримали корпус текстів, розподіл котрих можна побачити у *табл. 2.2*.

Джерело	**Кількість текстів**	**Класи**
Перекладені тексти з датасету Disinformation and Fake News	736	Fake
Датасет Ukrainian News	4913	Fake, Real
Джерела з фактчекінгових сайтів	1064	Fake
Puer news 24	33	Fake
“Україна Live”	499	Fake
“Новини сьогодення”	1043	Fake
“Радіо Свобода” (Telegram)	700	Real
“Українська правда” (Telegram)	688	Real
hromadske (Telegram)	699	Real
“Новини Еспресо.TV” (Telegram)	693	Real
“NV | nv.ua | Радіо NV | Новини України | Аналітика | Відео| НВ |” (Telegram)	700	Real
Повноцінні статті з надійних сайтів (“Радіо Свобода”, “Українська правда”, hromadske та ін.)	1188	Real
Усього: 12956		

Таблиця 2.2.

Розподіл кількості текстів за джерелом

Табл. 2.3 показує середню та максимальну кількість словоформ та речень у текстах, а також значення моди.

Типи одиниць	Клас “справжні новини”			Клас “фейкові новини”			Усі класи		
	мода	макс.	сер.	мода	макс.	сер.	мода	макс.	сер.
Словоформи	8	4094	60.24	15	2276	77.41	8	4094	68.02
Речення	1	403	4.70	1	207	6.00	1	403	5.29

Таблиця 2.3

Кількісний розподіл мовних одиниць у корпусі

РОЗДІЛ 3. СТВОРЕННЯ ЗАСТОСУНКУ ДЛЯ ВИЯВЛЕННЯ ФЕЙКОВИХ НОВИН НА ОСНОВІ СТИЛЮ

У цьому розділі ми заглибимося в процес розробки та розгортання програми, спрямованої на виявлення фейкових новин. Ми починаємо з опису загальної архітектури застосунку для виявлення фейкових новин, що охоплює навчальні дані, модуль попередньої обробки, навчальний цикл, серверну частину та вебінтерфейс.

Далі в розділі ми опишемо навчання бінарних класифікаторів (наївний Байєс, логістична регресія, нейронні мережі). Ми коротко порівняємо ці методи та запропонуємо ідеї щодо покращення остаточної версії моделі.

3.1 Загальна архітектура застосунку-детектора фейкових новин

Застосунок для виявлення фейкових новин містить такі основні компоненти: навчальні дані, модуль попередньої обробки, навчальний цикл, прикладний програмний інтерфейс, вебінтерфейс.

I. Навчальні дані представлені файлами у форматі CSV, котрі містять описаний нами у Розділі 2 корпус з текстами новин. Продовжуючи роботу над класифікацією новин у майбутньому, додамо також нові дані від користувачів, котрі збиратимемо за допомогою інтерфейсу. Ці нові дані міститимуть текст новини, передбачений клас та оцінку передбачення користувачем (“True” — позитивна оцінка, “False” — негативна), і їх можна використати для донавчання моделі.

II. Модуль попередньої обробки даних містить набір функцій, котрі застосовуються перед навчанням. Є кілька методів попередньої обробки: очищення (для видалення шуму та виправлення невідповідностей), інтеграція (об’єднання даних), редукція (зменшення кількості), нормалізація (розміщення даних у вужчому діапазоні) [57, с. 83] тощо.

Наш модуль попередньої обробки містить, серед інших, такі методи:

1. Конкатенація заголовка та тексту. Для кожного елемента у фреймі даних ми об’єднуємо заголовок та текст, а не покладаємося лише на

стиль тексту, оскільки заголовок є невіддільною частиною досліджень фейкових новин. М. Маянк та ін. при створенні фреймворку DEAP-FAKED представляють заголовки новин у вигляді безперервних векторів за допомогою енкодера biLSTM, що підходить для коротких текстів [65, с. 48]. О. Нолан та ін. наполягають на тому, що заголовки мають вирішальне значення для виявлення оманливого контенту [69, с. 1]. Згідно з нашими спостереженнями, новини з ненадійних джерел часто мають нейтральний тон у самому тексті й емоційні заголовки, що містять слова та вирази “дуже цікава новина”, “нарешті”, “справжнє чудо”, “такого ніхто не чекав”, “увага”, “щойно” та ін.. Отже, заголовки є важливими для виявлення фейкових новин, тож їх варто об’єднати з основним текстом.

2. Створення словника. Для розпізнавання тексту комп’ютером необхідно представити його у числовому вигляді. Для цього спочатку створюємо словник, у якому кожна лема сполучена з відповідним унікальним цілим числом.
3. Лематизація (частина нормалізації тексту). Цей процес полягає у зведенні слова до його словникової форми, і він початково слугував для ідентифікації спільнокореневих слів [59, с. 24]. Для нашого експерименту цей етап особливо важливий: українська мова флективна, а отже, зведення різноманіття лексем до однієї леми дозволяє ідентифікувати мовні одиниці, котрі мають те саме семантичне значення, — незалежно від часу, числа чи роду.
4. Вилучення стоп-слів. Ця операція зменшує кількість “шуму”, що присутній в обох класах і не впливає на результати. Для цього ми використовуємо готовий список стоп-слів української мови, куди входять сполучники, займенники, частки, числівники тощо (усього 1983 слова) [52].
5. Токенізація тексту. На цьому етапі ми використовуємо створений

словник та замінюємо кожне слово на його числовий відповідник. Крім цього, ми визначаємо максимальну довжину тексту у 150 токенів: якщо оригінальний текст більший, ми його скорочуємо; якщо ж коротший, додаємо до нього нулі, допоки довжина не почне дорівнювати 150.

Код модуля знаходиться в Додатку А.

III. Навчальний цикл — компонент, що відповідає за навчання бінарного класифікатора. Під час початкового навчання ми обирали між кількома алгоритмами. Деталі тренування та обґрунтування вибору кінцевого методу подаємо у п. 3.3. Після закінчення циклу ми зберігаємо найкращі ваги та підвантажуємо їх до прикладного програмного інтерфейсу.

IV. Вебінтерфейс: Інтернет-сторінка, за допомогою якої користувач може вставляти текст і заголовок новини або посилання та отримувати у відповідь один з передбачених класів (фейкова новина або справжня). Цей компонент створений за допомогою бібліотеки Streamlit, що дозволяє будувати інтерфейсну частину застосунків мовою Python [84] та надає доступ до простих елементів: кнопок, текстових рядків, заголовків, віджетів тощо.

V. Прикладний програмний інтерфейс (англ. *API*, *application programming interface*), котрий може отримувати два види HTTP-запитів: GET (перевірка зв’язку із сервером) і POST (надсилання тексту для бінарної класифікації). Отримавши запит POST, сервер застосовує збережену вагову функцію для класифікації новини. Прикладний програмний інтерфейс створено за допомогою бібліотеки FastAPI та через сервіси Lambda [78] та Docker [39].

Схематично описані вище компоненти зображено в Додатку В.

3.2 Оцінювання бінарних класифікаторів

Для оцінки ефективності навчених мовних моделей використовуємо стандартні метрики: точність, влучність, повноту та F1-міру.

Точність (англ. *accuracy*) є співвідношенням правильних передбачень до загальної кількості розглянутих випадків [31] і визначається формулою:

$$\text{Точність} = (\text{ІП} + \text{ІН}) \div (\text{ІП} + \text{ІН} + \text{ХП} + \text{ХН}), \text{ де}$$

ІП — істинно позитивні значення, ІН — істинно негативні, ХП — хибно позитивні, ХН — хибно негативні.

Влучність (англ. *precision*) — це співвідношення позитивних передбачень до тих, які справді є позитивними [59, с. 71]. Для нашого випадку ми обчислюємо, скільки елементів модель передбачає як “фейкова новина” — порівняно з тим, скільки фейкових новин насправді є у вибірці. Влучність обчислюється так:

$$\text{Влучність} = \text{ІП} \div (\text{ІП} + \text{ХП})$$

Повнота (англ. *recall*) “вимірює відсоток елементів, що дійсно присутні у вхідних даних і були правильно ідентифіковані системою” [59, с. 71]. Формула для обчислення повноти виглядає так:

$$\text{Повнота} = \text{ІП} \div (\text{ІП} + \text{ХН})$$

Завдяки цій мірі можемо визначати, коли модель пропускає позитивні випадки фейкових новин.

F1-міра — це середнє гармонійне значень влучності та повноти [59, с. 71]:

$$F1 = 2\text{ВП} \div (\text{В} + \text{П}) \text{ , де}$$

В — влучність, П — повнота. У нашому випадку важливо покладатися не лише на точність, а й на F1-міру, що своєю чергою залежить від значень влучності та повноти, з двох причин:

1. Незбалансований набір даних у бік негативного класу (справжніх новин).

Модель може мати високу точність лише тому, що добре ідентифікує клас більшості, однак це не обов’язково означає, що вона добре розпізнаватиме фейки.

2. Значущість позитивного класу (фейкові новини). F1-міра не бере до уваги істинно негативні значення, а отже допомагає оцінити, наскільки добре модель ідентифікує позитивний клас загалом. У системах з виявлення фейків першим етапом може бути детекція потенційних фейкових новин на основі стилю, а вже потім мануальна оцінка таких “кандидатів” експертами. Для такого сценарію важливішим є не пропустити потенційні фейкові новини, ніж точність моделі загалом.

Оцінюємо ефективність класифікатора (застосовуємо згадані вище метрики) на двох вибірках даних: а) тестовій частині створеного нами мовного корпусу (15% або 20% від їхньої загальної кількості) — назвемо її **стандартною тестовою вибіркою**; б) 78 нових текстах, що зібрані вручну з різних видів джерел, далі — **нова тестова вибірка**. В останній, на відміну від стандартних тестових даних, до класу “фейкові новини” входять ті тексти, що дійсно містять фейкову інформацію. На противагу, зібраний нами початковий корпус охоплює, окрім фейків, дані з ненадійних медіа, котрі не обов’язково належать до оманливого контенту.

3.3 Навчання бінарних класифікаторів тексту

У процесі експериментів з різними методами та алгоритмами навчання ми протестували наївний байєсів класифікатор, логістичну регресію, нейронну мережу з архітектурою довгої короткочасної пам’яті та нейронну мережу прямого поширення з ембедингами DistilBERT. Для порівняння ефективності навчених моделей наводимо таблиці з метриками, згаданими у п. 3.2.

3.3.1 Наївний байєсів класифікатор

Для імплементації наївного байєсового класифікатора (див. п. 1.3.2.1) використовуємо бібліотеку scikit-learn, котра широко використовується для машинного навчання [82]. Ми ділимо оригінальний мовний корпус на дві частини: навчальну вибірку (80%), для якої підбираємо параметри ймовірнісної моделі, та стандартну тестову вибірку (20%).

Accuracy	**Precision**	**Recall**	**F1**
0.62	0.67	0.32	0.43

Таблиця 3.1

Наївний байєсів класифікатор на стандартній тестовій вибірці

Accuracy	**Precision**	**Recall**	**F1**
0.38	0.26	0.40	0.31

Таблиця 3.2

Наївний байєсів класифікатор на новій тестовій вибірці

Як бачимо з *табл. 3.1* та *3.2*, результати класифікації досить низькі. Загалом цей метод занадто простий для ідентифікації такого явища, як фейкові новини. Наївне припущення Байєса — це припущення про умовну незалежність: імовірності $P(f_i|c)$ є незалежними для класу c [59, с. 63]. Фейкові новини рідко мають слова, котрі є унікальними лише для фейкових новин, а контекст і їхня послідовність суттєво впливають на значення.

3.3.2 Логістична регресія

Для застосування логістичної регресії (див. п. 1.3.2.1) використовуємо бібліотеку для машинного навчання PyTorch [74]. Цього разу ми ділимо оригінальний мовний корпус не на дві, а на три частини: навчальні дані, валідаційні дані (для оцінки моделі в конкретній епосі) та стандартну тестову вибірку (для оцінки моделі після завершення навчання). Входом є тензор зі 150

вимірами (що дорівнює максимальній довжині тексту). На виході застосовуємо логістичну функцію, що визначає ймовірність належності тексту до фейкових новин. Якщо ймовірність вища, ніж 0.5, передбачення вважається позитивним, якщо менша — негативним. Після кожної ітерації обчислюємо функцію втрат для валідаційних даних [28] та оновлюємо ваги та зсуви, якщо показник цієї функції нижчий за попередній найкращий результат.

У *табл. 3.3* та *3.4* наводимо показники метрик для стандартної та нової тестової вибірки.

Спроба	**Accuracy**	**Precision**	**Recall**	**F1**
1	0.58	0.53	0.83	0.65
2	0.61	0.55	0.83	0.66
3	0.47	0.47	0.99	0.64
4	0.59	0.54	0.78	0.64
5	0.59	0.54	0.77	0.64

Таблиця 3.3

Логістична регресія на стандартній тестовій вибірці

Спроба	**Accuracy**	**Precision**	**Recall**	**F1**
1	0.53	0.55	0.71	0.62
2	0.58	0.59	0.71	0.65
3	0.53	0.53	0.98	0.69
4	0.46	0.50	0.67	0.57
5	0.50	0.53	0.64	0.58

Таблиця 3.4

Логістична регресія на новій тестовій вибірці

Як ми бачимо з результатів (нижча влучність і вища повнота), хоча модель

ефективна щодо виявлення позитивних випадків, вона робить багато помилкових позитивних передбачень. Окрім цього, точність для нової тестової вибірки майже така сама, як якби ми вибирали класи випадковим чином.

3.3.3 Довга короткочасна пам’ять

Наступним етапом у нашому експерименті є застосування складнішого розв’язання — нейронної мережі LSTM (*long short-term memory*), принцип побудови котрої ми згадали у п. 1.3.2.2. Початкова архітектура нашої мережі складається з трьох двосторонніх шарів довгої короткочасної пам’яті (biLSTM) та одного лінійного шару. Для обчислення ймовірності належності одиниці даних до фейкових новин знову використовуємо сигмоїдну функцію.

Спроба	Accuracy	Precision	Recall	F1
1	0.79	0.80	0.73	0.76
2	0.77	0.76	0.73	0.75
3	0.78	0.80	0.71	0.75
4	0.78	0.76	0.77	0.76
5	0.79	0.80	0.71	0.75

Таблиця 3.5

Довга короткочасна пам’ять на стандартній тестовій вибірці (первинна імплементація)

Спроба	Accuracy	Precision	Recall	F1
1	0.65	0.74	0.55	0.63
2	0.68	0.77	0.57	0.66
3	0.72	0.88	0.55	0.68
4	0.71	0.79	0.62	0.70
5	0.62	0.70	0.50	0.59

Таблиця 3.6

Довга короткочасна пам’ять на новій тестовій вибірці

(первинна імплементація)

Результати початкової архітектури показано в *табл. 3.5* та *3.6*. Як бачимо, вони є досить непоганими порівняно з попередніми методами. На новій вибірці (спроба 4) точність становить приблизно 71%, а F1-міра — 70%. Хоча повнота є все ще низькою (близько 62%), це значно кращий результат, ніж для наївного Байєса та логістичної регресії.

У первинній архітектурі ми мали лише один лінійний шар. З наступною імплементацією (див. Додаток Б) спробуємо використати воронкову архітектуру, тобто додати кілька лінійних шарів, де кожен наступний буде мати у два рази меншу кількість прихованих вимірів, аж допоки ми не дійдемо до вихідного шару, на якому застосовуємо сигмоїдну функцію та отримаємо лише один вимір (значення від 0 до 1). Результати другої імплементації показуємо у *табл. 3.7 та 3.8*.

Спроба	**Accuracy**	**Precision**	**Recall**	**F1**
1	0.78	0.80	0.70	0.75
2	0.76	0.76	0.72	0.74
3	0.76	0.73	0.77	0.75
4	0.77	0.79	0.70	0.74
5	0.76	0.79	0.68	0.73

Таблиця 3.7

Довга короткочасна пам’ять на стандартній тестовій вибірці
(кінцева імплементація)

Спроба	**Accuracy**	**Precision**	**Recall**	**F1**
1	0.69	0.80	0.57	0.67
2	0.72	0.83	0.60	0.69
3	0.71	0.76	0.67	0.71

Таблиця 3.8

Довга короткочасна пам’ять на новій тестовій вибірці
(кінцева імплементація)

Продовження табл. 3.8

4	0.67	0.79	0.52	0.63
5	0.67	0.81	0.50	0.62

Під час третьої спроби маємо найкращі показники для нової тестової вибірки: точність — приблизно 71%, влучність — 76%, повнота — 67%, міра F1 — 71%. Кінцеву архітектуру мережі подано у Додатку Б.

3.3.4 Нейронна мережа прямого поширення з ембедингами DistilBERT

Останньою частиною нашого експерименту є створення нейронної мережі, де вхідними даними є контекстні векторні репрезентації, побудовані за допомогою моделі DistilBERT [38]. Ця модель генерує ембединги з 768 вимірами, що означає, що такі дані потребують значно більшої пам’яті RAM, ніж дані зі 150 вимірами. Через це ми використовуємо лише частину з нашого корпусу текстів (30%), а також застосовуємо простішу архітектуру мережі, впроваджуючи лише один прихований лінійний шар.

Спроба	**Accuracy**	**Precision**	**Recall**	**F1**
1	0.82	0.81	0.75	0.78
2	0.82	0.81	0.75	0.78
3	0.82	0.82	0.75	0.78
4	0.82	0.82	0.72	0.77
5	0.83	0.81	0.76	0.79

Таблиця 3.9

Нейронна мережа прямого поширення з ембедингами DistilBERT
на стандартній тестовій вибірці

Спроба	Accuracy	Precision	Recall	F1
1	0.55	0.63	0.41	0.49
2	0.51	0.58	0.33	0.42
3	0.54	0.62	0.38	0.47
4	0.54	0.65	0.31	0.42
5	0.54	0.62	0.38	0.47

Таблиця 3.10

Нейронна мережа прямого поширення з ембедингами DistilBERT
на новій тестовій вибірці

Як бачимо з *табл. 3.9* та *3.10*, така нейронна мережа показує хороші результати для стандартної тестової вибірки (найвища точність — близько 83%, міра F1 — 79%), що означає, що модель добре навчилася розрізняти дані, які входять до оригінального датасету. Однак для нової тестової вибірки показники набагато гірші (максимальна точність — близько 55%, F1-міра — 49%).

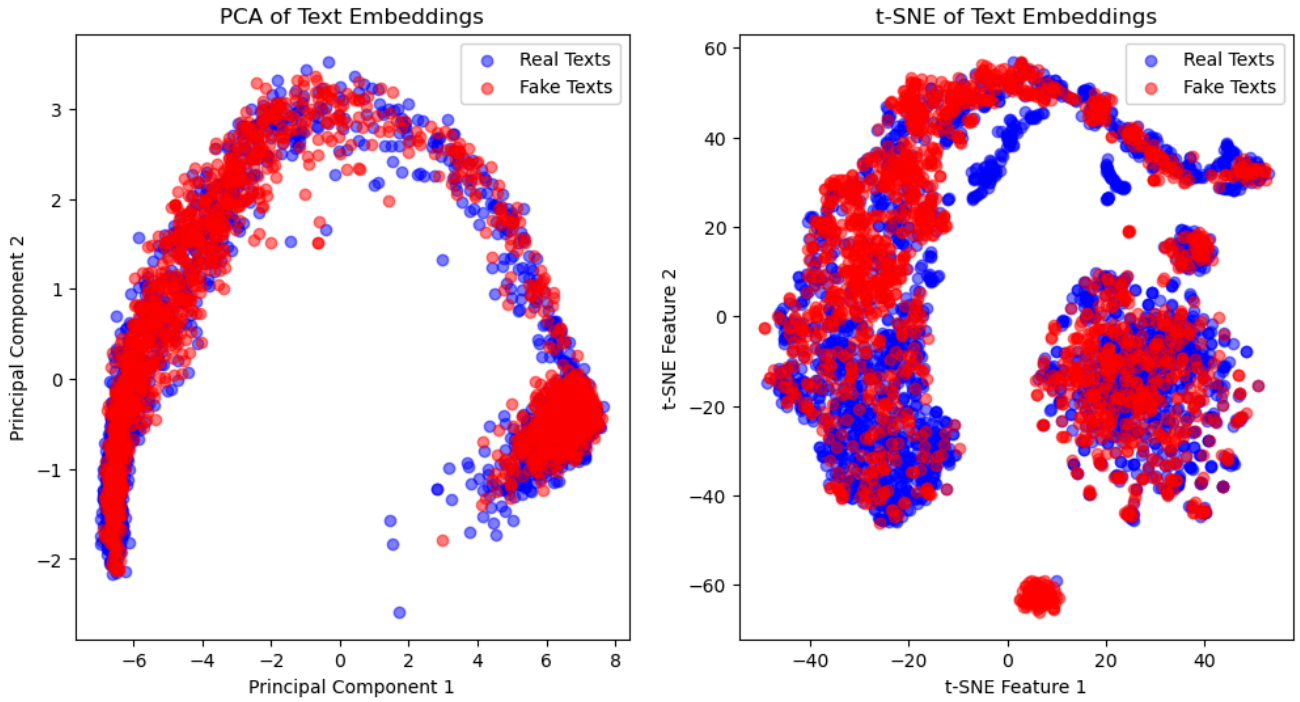
3.3.5 Аналіз результатів класифікації

Найкращий результат за узагальненою метрикою для нової тестової вибірки (F1 = 71%, модель — довга короткочасна пам’ять) отримано для таких гіперпараметрів: розмірність простору векторного представлення слів — 150, швидкість навчання — 0.0005, початкова кількість епох - 2000, поріг рішень - 0,5.

Як ми бачимо з результатів навчання моделей, лише довга короткочасна пам’ять показує хороші результати. Нездатність моделей робити узагальнення вказує на те, що, можливо, тексти з різних класів у навчальній вибірці занадто подібні між собою. Щоб перевірити цю гіпотезу, представимо тексти у вигляді ембедингів DistilBERT та за допомогою вже згаданої бібліотеки scikit-learn зменшимо кількість вимірів із 768 до 2. Результати показуємо на *граф. 3.1*.

Обидва класи займають майже той самий простір. З цього можна зробити

висновок, що фейкові тексти та справжні дуже схожі між собою семантично. Розв'язати цю проблему, імовірно, можна завдяки ідентифікації найбільш поширених спільних мовних одиниць (наприклад, лем) та попереднє їхнє вилучення з текстів.



Графік 3.1. Візуалізація зредукованих ембедингів

Повертаючись до метрик щодо нової тестової вибірки для довгої короткочасної пам'яті, бачимо, що для найкращої спроби (3) повнота (0.67) є дещо нижчою, ніж інші показники, що вказує на те, що модель не виявляє значну частку позитивних випадків. Сигмоїдна функція, котра перетворює результати мережі на вихідне значення, продукує число між 0 та 1. Під час навчання та тестування ми визначили поріг у 0.5: вищі значення вказують на фейкові новини, нижчі — на справжні. Однак нездатність модель виявляти позитивні випадки вказує на те, що, можливо, цей поріг занадто високий. Ми випробували кілька значень, і найкращі результати для нової тестової вибірки має поріг у 0.03: точність — 0.76, влучність — 0.77, повнота — 0.79, міра F1 — 0.78. Якщо знизимо поріг до 0, результат різко погіршиться: точність — 0.53, влучність — 0.54, повнота — 1, F1 — 0.7. З цього можна зробити висновок, що

наша модель здатна до генералізацій, однак водночас вона дуже чутлива до найменших ознак “фейковості” тексту. Навіть якщо модель лише трохи впевнена (на 3%), що текст належить до позитивного класу, вона класифікує його як такий.

Після застосування зменшеного порогу проаналізуємо збіг лексики між різними категоріями класифікованих текстів (істинно позитивні (ІП), істинно негативні (ІН), хибно позитивні (ХП), хибно негативні (ХН)). Порахувавши 1000 найпоширеніших лем у кожній групі та співставивши їхній перетин, маємо такі результати:

- ІП ∩ ХП: 268 лем;
- ІН ∩ ХН: 126 лем;
- ІП ∩ ХН: 135 лем;
- ІН ∩ ХП: 291 лема.

Перетин є вищим для істинно позитивного з хибно позитивним класом та для істинно негативного з хибно позитивним класом. Перший випадок свідчить про те, що існує, імовірно, значна кількість ключових слів, що вказують на позитивний клас, і це потребує подальшого аналізу для зменшення хибно позитивних значень. Другий випадок є можливою вказівкою на те, що багато слів, які мали б класифікуватися як негативний клас, натомість визначаються як позитивний (фейкові новини). Для вдосконалення моделі варто зосередити увагу на цих двох перетинах.

Висновки до третього розділу

У цьому розділі ми описали програмний продукт, ціль котрого — виявляти фейкові новини. Ми розглянули його архітектурні компоненти: навчальні дані, модуль попередньої обробки, навчальний цикл, веб- та прикладний програмний інтерфейс.

Ми застосували та протестували кілька моделей для бінарної класифікації новин. Для оцінки ефективності навчених моделей використовуються такі показники, як точність, точність, швидкість запам'ятовування та оцінка F1.

Класифікатори оцінювалися як на стандартному наборі даних, так і на нещодавно зібраній вибірці, що охоплює реальні приклади фейкових новин. Серед класифікаторів найвищі результати показала нейронна мережа з довгою короткостроковою пам’яттю: точність — 71%, влучність — 76%, повнота — 67%, міра F1 — 71%.

У майбутньому робота над покращенням цього програмного продукту може включати:

- збирання нових даних, серед яких — внесені користувачами тексти;
- створення системи рекомендацій справжніх новин на ту ж саму тему, що й новини, котрі були класифіковані як фейкові (на основі семантичної подібності);
- експерименти з гіперпараметрами та архітектурою нейронної мережі;
- застосування стратегій активного навчання, коли модель виявляє випадки, в яких вона не впевнена, і залучає людину для їхньої анотації;
- використання складніших методів відбору ознак — наприклад, на основі розпізнавання іменованих сутностей.

ВИСНОВКИ

У цій роботі було ґрунтовно досліджено проблему виявлення фейкових українськомовних новин. Ми розпочали з базових дефініцій (дезінформація, місінформація, фейкові новини, чутки та ін.) й окреслили соціальні й психологічні фактори, що впливають на розпізнавання фейків людьми. У дослідженні було описано різні методи виявлення такого контенту. Особливо детально ми сконцентрувалися на аналізі стилю тексту, що може відбуватися з допомогою традиційного та глибокого машинного навчання.

У результаті проведеної роботи нам вдалося виконати окреслені завдання:

1. Ми надали різні визначення поняття “фейк” та в процесі дослідження використовували цей термін у його широкому розумінні (будь-яка новина з хибним твердження — незалежно від наміру). Спорідненими поняттями є дезінформація, дипфейк, місінформація, сатиричні новини, чутки.
2. Було описано різноманітні підходи до виявлення фейкових новин, як-от методи на основі стилю, знань, джерела та способу поширення.
3. У роботі охарактеризовано методи виявлення фейкових новин за допомогою АОПМ, котрі полягають в бінарній класифікації текстів. Новітні засоби головним чином залучають глибоке навчання — тренування різних типів нейронних мереж (згорткові, рекурентні, довга короткочасна пам’ять, трансформери та ін.).
4. Ми проаналізували готові набори даних для навчання бінарного класифікатора та створили власний невеликий мовний корпус (див. Додаток Г). Він охоплює тексти з опублікованих в Інтернеті датасетів та власноруч зібрані дані.
5. Ми навчили та порівняли між собою такі класифікатори: наївний Байєс, логістична регресія, послідовна нейронна мережа з ембедингами DistilBERT та довга короткочасна пам’ять. Останнє розв’язання виявилося найефективнішим: точність становить 0.71

F1-міра — 0.67. Опустивши поріг чутливості до 0.03, маємо точність 0.76, F1-міру 0.78.

6. Ми практично реалізували розпізнавання фейкових новин на основі стилю — нам вдалося розробити програму, через яку користувач може класифікувати тексти новин. Наше розв’язання інтегрує інтерфейс користувача з системою бінарної класифікації новин, що заснована на нейронній мережі довгої короткочасної пам’яті (див. Додаток Д).

Ми можемо окреслити кілька напрямів майбутніх досліджень. По-перше, перспективним є гібридний спосіб розпізнавати фейкові новини, котрий би поєднував класифікацію тексту з, до прикладу, автоматичним фактчекінгом або аналізом метаданих. По-друге, оскільки українська мова є малоресурсною, подальші дослідження у цій сфері повинні залучати створення більших анотованих наборів даних, котрі містили б широкий спектр фейкових і справжніх новинних статей. По-третє, майбутні експерименти можуть перетинатися з іншими завданнями у сфері АОПМ. Наприклад, сентимент-аналіз буде корисним для розпізнавання оманливого контенту через наявність у ньому емоційної складової.

Хоча нам і вдалося досягти деяких результатів у виявленні фейкових новин, складність проблеми гарантує її подальшу актуальність. Крім цього, виявлення фейків є критично важливим у контексті інформаційної безпеки, а тому необхідні надійні підходи до боротьби з дезінформацією.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Білий список: 11 медіа, що стали найякіснішими. *Інститут масової інформації*. URL: https://imi.org.ua/news/bilyj-spysok-11-media-shho-staly-najyakisnishymy-i60964 (дата звернення: 27.04.2024).
2. Боротьба з фейками та цифрова гігієна — DeepState UA. *DeepState UA*. URL: https://deepstateua.com/tag/borotba-z-fieikami-ta-tsifrova-ghighiiena (дата звернення: 01.05.2024).
3. Вихованець І. Р. Граматика української мови. Синтаксис : підруч. для студ. філол. ф-тів вузів. Київ : Либідь, 1993. 365 с.
4. Вікіновини. *Вікіновини*. URL: https://uk.wikinews.org/wiki/Головна (дата звернення: 15.02.2024).
5. Грищенко О. В. Фейкова мовна особистість із погляду дискурсивної лінгвістики. *Науковий вісник Дрогобицького державного педагогічного університету імені Івана Франка. Сер. : Філологічні науки (мовознавство)*. 2016. № 6. С. 39–41. URL: http://ddpu-filolvisnyk.com.ua/uploads/arkhiv-nomerov/2016/NV_2016_6/10.pdf (дата звернення: 22.01.2024).
6. Дарчук Н. П. Комп‘ютерна лінгвістика (автоматичне опрацювання тексту) : підручник. Київ : Видавничо-полігр. центр «Київ. ун-т», 2008. 351 с.
7. Доскіч Л. С. Фейкові новини як новітній засіб маніпуляції та дезінформації. *Бібліотекознавство. Документознавство. Інформологія*. 2022. № 4. С. 72–77. URL: https://journals.uran.ua/bdi/article/view/269809 (дата звернення: 21.01.2024).
8. Дукач Ю. UPD:Труха. Як популярний телеграм-канал розганяє фейки і придумує відмазки, коли це помічають. *Texty.org.ua — статті та журналістика даних для людей — Тексти.org.ua*. URL: https://texty.org.ua/articles/107377/informacijna-telehram-smittyarka-dlya-2-miljoniv (дата звернення: 07.03.2024).
9. Дяк Т. П., Грицюк Ю. І., Горват П. П. Проблема виявлення фейкових

новин на веб-сайтах мережі Інтернет. *Науковий вісник НЛТУ України*. 2022. Т. 32(6). С. 78–94. URL: https://doi.org/10.36930/40320612 (дата звернення: 18.02.2024).

10. Кіца М. О. Особливості та методи виявлення фейкової інформації в українських ЗМІ. *Вісн. Нац. ун-ту "Львів. політехніка"*. 2017. № 883. С. 28–32. URL: https://science.lpnu.ua/sites/default/files/journal-paper/2019/apr/16109/kitsa.pdf (дата звернення: 24.01.2024).
11. Кіца М. О. Фейкова інформація в українських соціальних медіа: поняття, види, вплив на аудиторію. *Наук. зап./Укр. акад. друкарства*. 2016. № 1. С. 281–287. URL: http://nbuv.gov.ua/UJRN/Nz_2016_1_37 (дата звернення: 30.03.2024).
12. Миколаєнко А. Ю. Фейкові новини в українському медіапросторі: технології експериментальних проектів. *Наукові записки Інституту журналістики*. 2019. Т. 1(74). С. 29–38. URL: https://doi.org/10.17721/2522-1272.2019.74.3 (дата звернення: 06.03.2024).
13. Мордюк А. О. Як працювати з інтернет-контентом, щоб не стати жертвою маніпуляції: поради журналістам від вітчизняних та європейських експертів. *Наукові записки Інституту журналістики*. 2014. Т. 56. С. 240–246. URL: http://nbuv.gov.ua/UJRN/Nzizh_2014_56_48 (дата звернення: 11.03.2024).
14. Мюллер Ф., Деннер Н. Як можна протидіяти «фейковим новинам»? / пер. з англ. А. Куликов. Київ : Фонд Фрідріха Науман. за Свободу, Акад. укр. преси, 2019. 32 с. URL: https://www.aup.com.ua/wp-content/uploads/2019/11/YAK_protydiiaty_fake_news_2019.pdf (дата звернення: 04.03.2024).
15. Новини Сьогодення — Це вражає. *Новини Сьогодення — Це вражає*. URL: https://www.presentnews.biz.ua (дата звернення: 11.02.2024).
16. Омельчук Ю. О. Об'єктивація фреймів сфера правди / сфера неправди (на матеріалі сучасних англомовних псевдоновин) [Текст]. *Філологічні*

трактати. 2017. Т. 9, № 3. С. 44–50. URL: http://essuir.sumdu.edu.ua/handle/123456789/68826 (дата звернення: 17.03.2024).

17. Україна Live. *Україна Live*. URL: https://ukr-live.com (дата звернення: 16.02.2024).
18. Українські медіа, ставлення та довіра у 2023 р. 2023. 95 с. URL: https://internews.in.ua/wp-content/uploads/2023/10/Ukrainski-media-stavlennia-ta-dovira-2023r.pdf (дата звернення: 01.05.2024).
19. Фейки і наративи. *Інститут масової інформації*. URL: https://imi.org.ua/monitorings/fakes-and-narratives (дата звернення: 12.04.2024).
20. Фейкові новини. *Реєстр фейків України*. URL: https://fake.net.ua/reestr/fake-news (дата звернення: 24.03.2024).
21. Шульська Н. М., Зінчук Р. С., Башманівський В. І. Фейкоінструменарій ведення інформаційної війни в Україні: на матеріалі мови сучасних медіа. *Вчені записки Таврійського національного університету імені В. І. Вернадського*. 2023. Т. 34 (73), № 1, Ч. 2. С. 274–279. URL: http://eprints.zu.edu.ua/id/eprint/36988 (дата звернення: 10.03.2024).
22. Afrati F., Momani Z., Stasinopoulos N. Cross-Checking Multiple Data Sources Using Multiway Join in MapReduce. *Scientific Programming*. 2017. Vol. 2017. P. 1–9. URL: https://doi.org/10.1155/2017/3072813 (date of access: 29.04.2024).
23. Afroz S., Brennan M., Greenstadt R. Detecting Hoaxes, Frauds, and Deception in Writing Style Online. *2012 IEEE Symposium on Security and Privacy*, San Francisco, 22–23 May 2012. 2012. P. 461–475. URL: https://ieeexplore.ieee.org/document/6234430 (date of access: 09.03.2024).
24. Alhindi T., Petridis S., Muresan S. Where is Your Evidence: Improving Fact-checking by Justification Modeling. *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. Brussels, 2018. P. 85–90. URL: https://aclanthology.org/W18-5513/ (date of access: 15.03.2024).

25. Allcott H., Gentzkow M. Social Media and Fake News in the 2016 Election. *Journal of Economic Perspectives*. 2017. Vol. 31(2). P. 211–236. URL: https://www.aeaweb.org/articles?id=10.1257/jep.31.2.211 (date of access: 06.03.2024).
26. AP Fact Check. *AP News*. URL: https://apnews.com/ap-fact-check (date of access: 08.05.2024).
27. Barriere V., Balahur A. Improving Sentiment Analysis over non-English Tweets using Multilingual Transformers and Automatic Translation for Data-Augmentation. *Proceedings of the 28th International Conference on Computational Linguistics*. 2020. P. 266–271. URL: https://doi.org/10.18653/v1/2020.coling-main.23 (date of access: 15.04.2024).
28. BCELoss — PyTorch 2.3 documentation. *PyTorch*. URL: https://pytorch.org/docs/stable/generated/torch.nn.BCELoss.html (date of access: 08.05.2024).
29. Beautiful Soup Documentation. *Crummy: The Site*. URL: https://www.crummy.com/software/BeautifulSoup/bs4/doc/ (date of access: 24.04.2024).
30. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / J. Devlin et al. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, 2019. P. 4171–4186. URL: https://doi.org/10.48550/arXiv.1810.04805 (date of access: 17.03.2024).
31. Bonnet A. Accuracy vs. Precision vs. Recall in Machine Learning: What is the Difference?. *The Complete Data Development Platform for AI | Encord*. URL: https://encord.com/blog/classification-metrics-accuracy-precision-recall/ (date of access: 18.04.2024).
32. Borges L., Martins B., Calado P. Combining Similarity Features and Deep Representation Learning for Stance Detection in the Context of Checking Fake News. *Journal of Data and Information Quality*. 2019. Vol. 11(3). P. 1–26.

URL: https://doi.org/10.48550/arXiv.1811.00706 (date of access: 14.03.2024).

33. Challenges and Opportunities in Information Manipulation Detection: An Examination of Wartime Russian Media / C. Y. Park et al. *Findings of the Association for Computational Linguistics: EMNLP 2022*. 2022. P. 5209–5235. URL: https://aclanthology.org/2022.findings-emnlp.382 (date of access: 06.02.2024).
34. Combining Vagueness Detection with Deep Learning to Identify Fake News / P. Guélorget et al. *2021 IEEE 24th International Conference on Information Fusion (FUSION)*. 2021. P. 1–8. URL: https://doi.org/10.48550/arXiv.2110.14780 (date of access: 17.02.2024).
35. DBpedia Association. *DBpedia Association*. URL: https://www.dbpedia.org(date of access: 09.03.2024).
36. de Beer D., Matthee M. Approaches to Identify Fake News: A Systematic Literature Review. *Integrated Science in Digital Age 2020*. Cham, 2020. P. 13–22. URL: https://doi.org/10.1007/978-3-030-49264-9_2 (date of access: 08.05.2024).
37. Disinformation and Fake News. *Kaggle: Your Machine Learning and Data Science Community*. URL: https://www.kaggle.com/datasets/corrieaar/disinformation-articles (date of access: 11.04.2024).
38. distilbert/distilbert-base-multilingual-cased. *Hugging Face — The AI community building the future.*URL: https://huggingface.co/distilbert/distilbert-base-multilingual-cased (date of access: 01.02.2024).
39. Docker: Accelerated Container Application Development. *Docker*. URL: https://www.docker.com (date of access: 22.04.2024).
40. Efficient Estimation of Word Representations in Vector Space / T. Mikolov et al. *International Conference on Learning Representations*. 2013. P. 1–12. URL: https://api.semanticscholar.org/CorpusID:5959482 (date of access: 15.04.2024).

41. Enriching Word Vectors with Subword Information / P. Bojanowski et al. *Transactions of the Association for Computational Linguistics*. 2017. Vol. 5. P. 135–146. URL: https://aclanthology.org/Q17-1010.pdf (date of access: 02.05.2024).
42. facebook/seamless-m4t-v2-large. *Hugging Face — The AI community building the future.*URL: https://huggingface.co/facebook/seamless-m4t-v2-large (date of access: 16.03.2024).
43. Fact Checker. *The Washington Post*. URL: https://www.washingtonpost.com/politics/fact-checker/ (date of access: 26.01.2024).
44. FactCheck.org. *FactCheck.org*. URL: https://www.factcheck.org/ (date of access: 08.02.2024).
45. Fact Check. *Reuters*. URL: https://www.reuters.com/fact-check/ (date of access: 03.02.2024).
46. Fake News detection. *Kaggle: Your Machine Learning and Data Science Community*. URL: https://www.kaggle.com/datasets/jruvika/fake-news-detection (date of access: 19.04.2024).
47. Fake News Detection on Social Media using Geometric Deep Learning / F. Monti et al. *ICLR*. 2019. P. 1–13. URL: https://rlgm.github.io/papers/34.pdf (date of access: 06.03.2024).
48. Fake News. *Kaggle: Your Machine Learning and Data Science Community*. URL: https://www.kaggle.com/datasets/hassanamin/textdb3 (date of access: 09.04.2024).
49. FakeNewsNet: A Data Repository with News Content, Social Context, and Spatiotemporal Information for Studying Fake News on Social Media / K. Shu et al. *Big Data*. 2018. Vol. 8(3). P. 171–188. URL: https://api.semanticscholar.org/CorpusID:85528899 (date of access: 10.02.2024).
50. Getting Real about Fake News. *Kaggle: Your Machine Learning and Data*

Science Community. URL: https://www.kaggle.com/datasets/mrisdal/fake-news (date of access: 11.04.2024).

51. Giasson M. B. F. KBpedia. *KBpedia*. URL: https://kbpedia.org (date of access: 04.03.2024).
52. GitHub — skupriienko/Ukrainian-Stopwords: the list of ~2000 ukrainian stopwords (with numbers). *GitHub*. URL: https://github.com/skupriienko/Ukrainian-Stopwords/tree/master (date of access: 13.03.2024).
53. Goodfellow I. J., Bengio Y., Courville A. Deep Learning. Cambridge : MIT Press, 2016. 785 p.
54. Googletrans: Free and Unlimited Google translate API for Python — Googletrans 3.0.0 documentation. *Googletrans: Free and Unlimited Google translate API for Python — Googletrans 3.0.0 documentation*. URL: https://py-googletrans.readthedocs.io/en/latest/ (date of access: 14.03.2024).
55. Graph-Based Modeling of Online Communities for Fake News Detection / S. Chandra et al. *ArXiv*. 2020. Abs/2008.06274. P. 1–16. URL: https://doi.org/10.48550/arXiv.2008.06274 (date of access: 16.02.2024).
56. Guo Z., Schlichtkrull M., Vlachos A. A Survey on Automated Fact-Checking. *Transactions of the Association for Computational Linguistics*. 2022. Vol. 10. P. 178–206. URL: https://doi.org/10.1162/tacl_a_00454 (date of access: 08.05.2024).
57. Han J., Kamber M., Pei J. Data Mining: Concepts and Techniques. 3rd ed. Waltham : Morgan Kaufmann Publishers, 2012. 703 p.
58. Hoy N., Koulouri T. A Systematic Review on the Detection of Fake News Articles. *ArXiv*. 2021. Abs/2110.11240. P. 1–23. URL: https://doi.org/10.48550/arXiv.2110.11240 (date of access: 28.01.2024).
59. Jurafsky D., Martin J. H. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Third Edition Draft. 2024. 569 p. URL: https://web.stanford.edu/~jurafsky/slp3/ed3bookfeb3_2024.pdf (date of access:

17.04.2024).

60. LCAD — UFES at FakeDeS 2021: Fake News Detection Using Named Entity Recognition and Part-of-Speech Sequences / M. A. Spalenza et al. 2021. P. 1–9. URL: https://api.semanticscholar.org/CorpusID:238207976 (date of access: 29.02.2024).
61. MAFALDA: A Benchmark and Comprehensive Study of Fallacy Detection and Classification / C. Helwe et al. *ArXiv*. 2023. Abs/2311.09761. P. 1–36. URL: https://doi.org/10.48550/arXiv.2311.09761 (date of access: 10.04.2024).
62. Mainstream News Articles Co-Shared with Fake News Buttress Misinformation Narratives / P. Goel et al. *ArXiv.*2023. Abs/2308.06459. P. 1–86. URL: https://doi.org/10.48550/arXiv.2308.06459 (date of access: 02.03.2024).
63. Manning C. D. Foundations of Statistical Natural Language Processing. Cambridge, Mass : MIT Press, 1999. 680 p.
64. Maslej-Krešňáková V., Sarnovský M., Jacková J. Use of Data Augmentation Techniques in Detection of Antisocial Behavior Using Deep Learning Methods. *Future Internet*. 2022. Vol. 14, no. 9. P. 260. URL: https://doi.org/10.3390/fi14090260 (date of access: 19.02.2024).
65. Mayank M., Sharma S., Sharma R. DEAP-FAKED: Knowledge Graph based Approach for Fake News Detection. *2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, Istanbul, Turkey. 2022. P. 47–51. URL: https://doi.org/10.1109/asonam55673.2022.10068653 (date of access: 23.03.2024).
66. Meibauer J. The Linguistics of Lying. *Annual Review of Linguistics*. 2018. Vol. 4, no. 1. P. 357–375. URL: https://doi.org/10.1146/annurev-linguistics-011817-045634 (date of access: 02.02.2024).
67. Misinformation. *Cambridge Dictionary | English Dictionary, Translations & Thesaurus*. URL:

https://dictionary.cambridge.org/dictionary/english/misinformation (date of access: 21.03.2024).

68. Mitra T., Gilbert E. CREDBANK: A Large-Scale Social Media Corpus With Associated Credibility Annotations. *Ninth International AAAI Conference on Web and Social Media*. 2015. Vol. 9, no. 1. P. 258–267. URL: https://doi.org/10.1609/icwsm.v9i1.14625 (date of access: 13.03.2024).
69. Nolan O., van Mourik J., Tilbury C. R. BaIT: Barometer for Information Trustworthiness. *ArXiv*. 2022. Abs/2206.07535. P. 1–9. URL: https://doi.org/10.48550/arXiv.2206.07535 (date of access: 05.02.2024).
70. Pelrine K., Danovitch J., Rabbany R. The Surprising Performance of Simple Baselines for Misinformation Detection. *WWW '21: Proceedings of the Web Conference 2021*. 2021. P. 3432–3441. URL: https://doi.org/10.1145/3442381.3450111 (date of access: 18.02.2024).
71. Pennington J., Socher R., Manning C. GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha. 2014. P. 1532–1543. URL: https://aclanthology.org/D14-1162 (date of access: 02.05.2024).
72. PolitiFact. *PolitiFact*. URL: https://www.politifact.com (date of access: 14.01.2024).
73. Puer News 24: українські новини: останні події України. *Puer News 24*. URL: https://puer-press.org.ua (дата звернення: 28.02.2024).
74. PyTorch documentation — PyTorch 2.3 documentation. *PyTorch*. URL: https://pytorch.org/docs/stable/index.html (date of access: 16.02.2024).
75. Rumour. *Cambridge Dictionary | English Dictionary, Translations & Thesaurus*. URL: https://dictionary.cambridge.org/dictionary/english/rumour (date of access: 20.03.2024).
76. Separating Facts from Fiction: Linguistic Models to Classify Suspicious and Trusted News Posts on Twitter / S. Volkova et al. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver. 2017. P. 647–653. URL: https://aclanthology.org/P17-2102/ (date of access:

14.03.2024).

77. SEPSIS: I Can Catch Your Lies — A New Paradigm for Deception Detection / A. Rani et al. *ArXiv*. 2023. Abs/2312.00292. P. 1–30. URL: https://api.semanticscholar.org/CorpusID:265551498 (date of access: 24.02.2024).
78. Serverless Function, FaaS Serverless — AWS Lambda — AWS. *Amazon Web Services, Inc.*URL: https://aws.amazon.com/lambda/ (date of access: 26.04.2024).
79. Shrestha M. Detecting Fake News with Sentiment Analysis and Network Metadata. 2018. P. 1–6. URL: https://api.semanticscholar.org/CorpusID:207926728 (date of access: 25.02.2024).
80. Siering M., Jascha-Alexander K., Deokar A. Detecting Fraudulent Behavior on Crowdfunding Platforms: The Role of Linguistic and Content-Based Cues in Static and Dynamic Contexts. *Journal of Management Information Systems*. 2016. Vol. 33. P. 421–455. URL: https://api.semanticscholar.org/CorpusID:11633004 (date of access: 26.03.2024).
81. sklearn.ensemble.RandomForestClassifier. *scikit-learn*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (date of access: 15.04.2024).
82. sklearn.naive_bayes.GaussianNB. *scikit-learn*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html (date of access: 11.04.2024).
83. StopFake. *StopFake*. URL: https://www.stopfake.org/uk (date of access: 26.01.2024).
84. Streamlit: A Faster Way to Build and Share Data Apps. *Streamlit*. URL: https://streamlit.io (date of access: 19.04.2024).
85. Subject-Predicate-Object (SPO) Triple — GM-RKB. *Gabor Melli's Home Page*. URL:

http://www.gabormelli.com/RKB/Subject-Predicate-Object_(SPO)_Triple (date of access: 02.03.2024).

86. Telethon's Documentation — Telethon 1.35.1 documentation. *Telethon's Documentation — Telethon 1.35.1 documentation*. URL: https://docs.telethon.dev/en/stable/ (date of access: 23.03.2024).
87. The Impact of Translating Resource-Rich Datasets to Low-Resource Languages Through Multi-Lingual Text Processing / A. Ghafoor et al. *IEEE Access*. 2021. Vol. 9. P. 124478–124490. URL: https://ieeexplore.ieee.org/document/9529190 (date of access: 25.04.2024).
88. The Language of Fake News: Opening the Black-Box of Deep Learning Based Detectors / N. O'Brien et al. 2018. P. 1–5. URL: https://api.semanticscholar.org/CorpusID:54072129 (date of access: 10.03.2024).
89. The science of fake news / D. M. J. Lazer et al. *Science*. 2018. Vol. 359, no. 6380. P. 1094–1096. URL: https://doi.org/10.1126/science.aao2998 (date of access: 13.03.2024).
90. The Use of Data Augmentation as a Technique for Improving Neural Network Accuracy in Detecting Fake News About COVID-19 / W. O. Júnior et al. *ArXiv*. 2022. Abs/2205.00452. P. 1–11. URL: https://doi.org/10.48550/arXiv.2205.00452 (date of access: 05.03.2024).
91. translate. *PyPI*. URL: https://pypi.org/project/translate/ (date of access: 16.03.2024).
92. Tsai C. M. Stylometric Fake News Detection Based on Natural Language Processing Using Named Entity Recognition: In-Domain and Cross-Domain Analysis. *Electronics*. 2023. Vol. 12(3676). P. 1–16. URL: https://doi.org/10.3390/electronics12173676 (date of access: 30.03.2024).
93. UaReview. *UaReview*. URL: https://uareview.com (date of access: 07.02.2024).
94. Ukrainian News. *Kaggle: Your Machine Learning and Data Science Community*. URL:

https://www.kaggle.com/datasets/zepopo/ukrainian-fake-and-true-news?rvi=1 (date of access: 08.04.2024).

95. Undeutsch U. Beurteilung der Glaubhaftigkeit von Aussagen. *Handbuch der Psychologie*. 1967. Vol. 11. P. 26–181.
96. vanden Broucke S., Baesens B. Practical Web Scraping for Data Science. Berkeley, CA : Apress, 2018. URL: https://doi.org/10.1007/978-1-4842-3582-9 (date of access: 11.04.2024).
97. Van Swol L. M. Linguistic Cues. *Encyclopedia of Deception* / ed. by T. R. Levine. Los Angeles, 2014. Vol. 2. P. 606–608.
98. VoxCheck. *VoxUkraine*. URL: https://voxukraine.org/category/voxcheck-uk (date of access: 17.04.2024).
99. Vrij A. Detecting Lies and Deceit: Pitfalls and Opportunities. 2nd ed. Chichester : Wiley, 2008. 504 p.
100. What is a triple?. *Oxford Semantic Technologies Knowledge Graph & AI | RDFox*. URL: https://www.oxfordsemantic.tech/faqs/what-is-a-triple (date of access: 10.02.2024).
101. Wikidata. *Wikidata*. URL: https://www.wikidata.org/wiki/Wikidata:Main_Page (date of access: 30.03.2024).
102. Yago. *Yago Project*. URL: https://yago-knowledge.org (date of access: 18.02.2024).
103. Zhou X., Zafarani R. A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities. *ACM Computing Surveys*. 2020. Vol. 53, no. 5. P. 1–40. URL: https://doi.org/10.1145/3395046 (date of access: 16.12.2023).

ДОДАТКИ

Додаток А. Модуль для попередньої обробки тексту

```
import pandas as pd
import torch
import ast
import string
import pymorphy2
from collections import Counter
from typing import Callable
from nltk import sent_tokenize

morph = pymorphy2.MorphAnalyzer(lang='uk')
with open(r'stopwords_ua_list.txt', "r", encoding="utf-8") as f:
   stopwords = ast.literal_eval(f.read())
with open("markers.txt", "r", encoding="utf-8") as f:
   markers = [w.replace("\n", "") for w in f.readlines()[:100]]


def cat_titles_and_texts(titles: list[str], texts: list[str]) -> list[str]:
   """Конкатенація заголовків та текстів"""
   output = []
   for title, text in zip(titles, texts):
       try:
           if not text.startswith(title):
              text = title + " " + text
       except: pass
       output.append(text)
   return output

def is_not_punctuation(word) -> bool:
   """Перевірка, чи слово є пунктуацією"""
   return all(char not in string.punctuation for char in word)

def get_x(df: pd.DataFrame) -> list[str]:
   """Екстракція списку текстів з датафрейму"""
   x = []
   for _, row in df.iterrows():
       try:
           text = row["ukr_text"]
       except KeyError:
           text = row["Text"]
       x.append(text)
   return x

def get_x1(df: pd.DataFrame) -> list[str]:
   """Екстракція списку заголовків з датафрейму"""
   x1 = []
   for index, row in df.iterrows():
       try:
           title = row["title_ukr"]
       except LookupError:
           text = row["Text"]
           title = sent_tokenize(text)[0]
       x1.append(title)
   return x1

def get_y(df: pd.DataFrame) -> list[int]:
```

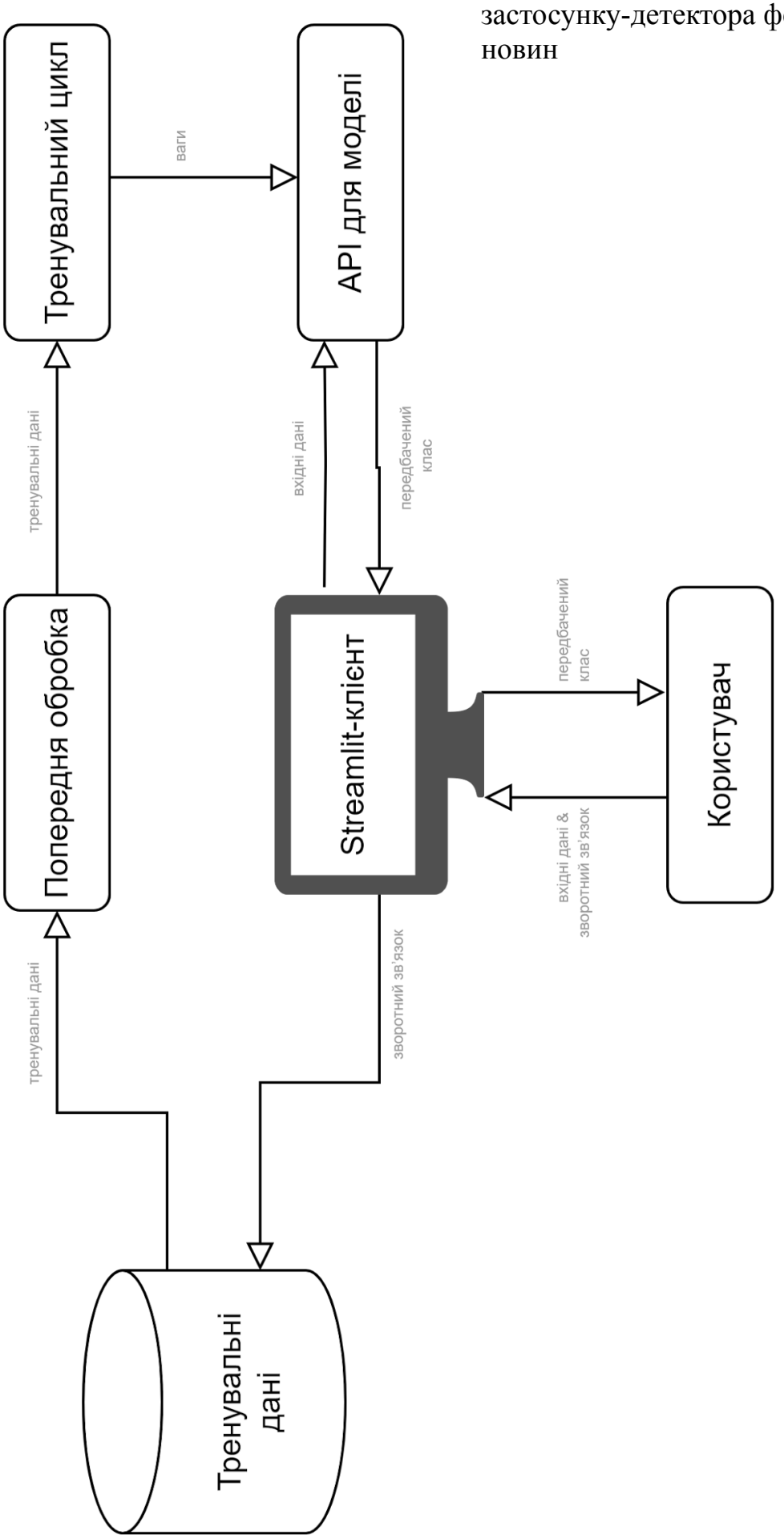
```
    """Екстракція класів з датафрейму"""
    try:
        y = df["label"]
    except KeyError:
        y = df["Label"]
    nums = []
    for element in y:
        if element == "Real" or element == True: nums.append(float(0))
        else: nums.append(float(1))
    return nums

def create_dictionary(x: list[str]) -> dict:
    """Створення словника (слово = int)"""
    dictionary = dict()
    count = 1
    for text in x:
        words = preprocess_text(text)
        for i, word in enumerate(words):
            if word != 0:
                if word not in dictionary:
                    dictionary[word] = count
                    count += 1
                else:
                    words[i] = dictionary[word]
    return dictionary

def tokenize_x(x: list[str], dictionary: dict) -> list[list[float]]:
    """Репрезентація тексту в нумеричному вигляді"""
    tokenized = []
    for text in x:
        words = preprocess_text(text)
        for i, word in enumerate(words):
            if word not in dictionary or word in markers or word in stopwords or not
is_not_punctuation(word):
                words[i] = 0
            else:
                words[i] = dictionary[word]
        tokenized.append(words)
    return tokenized

def lemmatize_word(word: str) -> str:
    """Лематизація слова"""
    return morph.parse(word)[0].normal_form

def bert_tokenize_without_masks(x: list[str], tokenizer: Callable) -> (torch.Tensor):
    """Токенізація за допомогою DistilBERT"""
    if torch.cuda.is_available():
        device = torch.device("cuda")
    else:
        device = torch.device("cpu")

    tokenized_x = []
    for text in x:
        encoded_dict = tokenizer.encode_plus(
            text,
            add_special_tokens = True,
            max_length = 150,
            truncation = True,
            padding = 'max_length',
```

```
            return_tensors = 'pt',
        )

        tokenized_x.append(encoded_dict['input_ids'].to(device))

    tokenized_x = torch.cat(tokenized_x, dim=0)
    return tokenized_x

def get_vocab_size(texts: list[str]) -> int:
    """Кількість унікальних слів"""
    all_texts = ""
    for t in texts:
        try:
            all_texts += (t + " ")
        except: pass
    vocab_size = len(set(all_texts.lower().split()))
    return vocab_size

def preprocess_text(text: str, padding: int=150) -> list[str]:
    """Застосування лематизації та падингу"""
    text = str(text)
    words = [lemmatize_word(word) for word in text.split() if word.lower() not in stopwords]
    padded = words[:padding] + [0] * (padding - len(words))
    return padded

def balance_data(df) -> pd.DataFrame:
    """Збалансування даних у датафреймі"""
    label_col = 'Label' if 'Label' in df.columns else 'label'
    fake_label = False if 'Label' in df.columns else 'Fake'
    real_label = True if 'Label' in df.columns else 'Real'

    try:
        num_fake = (df['Label'] == False).sum()
    except: num_fake = (df['label'] == "Fake").sum()
    try:
        num_real = (df['Label'] == True).sum()
    except:
        num_real = (df['label'] == "Real").sum()

    if num_real > num_fake:
        real_indices = df[df[label_col] == real_label].index
        sampled_real_indices = pd.DataFrame(real_indices).sample(n=num_fake,
random_state=42).index
        print(sampled_real_indices)
        df_balanced = df.loc[sampled_real_indices.union(df[df[label_col] == fake_label].index)]
    else:
        df_balanced = df
    return df_balanced

def get_average_length(x: list[str]) -> int:
    """Середня кількість слів у тексті"""
    lengths = []
    for text in x:
        try:
            words = [lemmatize_word(word) for word in text.split() if word.lower() not in
stopwords]
            lengths.append(len(words))
        except: lengths.append(0)
    return sum(lengths) / len(lengths)
```

Додаток Б. Визначення класу з моделлю LSTM та гіперпараметри

```
class LSTMClassifier(nn.Module):
   """Архітектура мережі"""
   def __init__(self,
                vocab_size,
                embedding_dim,
                hidden_dim,
                output_dim,
                num_layers,
                bidirectional,
                dropout):

       super().__init__()
       self.embedding = nn.Embedding(vocab_size, embedding_dim, padding_idx=0)
       self.lstm = nn.LSTM(embedding_dim, hidden_dim, num_layers=num_layers,
                           bidirectional=bidirectional, dropout=dropout,
batch_first=True)
       hidden_dim = hidden_dim * 2 if bidirectional else hidden_dim

       self.fc1 = nn.Linear(hidden_dim, hidden_dim // 2)
       self.fc2 = nn.Linear(hidden_dim // 2, hidden_dim // 4)

       self.fc3 = nn.Linear(hidden_dim // 4, output_dim)
       self.act = nn.Sigmoid()

   def forward(self, x):

       x_embedded = self.embedding(x)

       _, (hidden_x, _) = self.lstm(x_embedded)

       if self.lstm.bidirectional:
           hidden_x = torch.cat((hidden_x[-2,:,:], hidden_x[-1,:,:]), dim=1)
       else:
           hidden_x = hidden_x[-1,:,:]

       hidden_x = torch.relu(self.fc1(hidden_x))
       hidden_x = torch.relu(self.fc2(hidden_x))

       output = self.fc3(hidden_x)
       return self.act(output)


# гіперпараметри
vocab_size = get_vocab_size(x)
embedding_dim = len(X_train_tensor[0])
output_dim = 1
hidden_dim = 32
num_layers = 3
learning_rate=0.0005
dropout = 0.0
num_epochs = 2000
bidirectional = True
```

Додаток В. Архітектура застосунку-детектора фейкових новин



Додаток Г. Корпус з новинами українською мовою

Посилання: https://www.kaggle.com/datasets/sophiamatskovych/fake-news-ua

Опис: цей корпус містить колекцію новинних статей та текстів із соцмереж разом з їхніми метаданими і має такі поля:

- title_ukr: назва новини українською мовою.
- url: посилання, за яким можна знайти оригінальну новину.
- source_text: повний текст новини мовою оригіналу.
- source_lang: мова оригіналу.
- ukr_text: український переклад тексту.
- label: “Real” (справжні новини) або “Fake” (фейкові новини та тексти з ненадійних джерел).
- date: дата публікації тексту.

Додаток Д. Вебзастосунок для класифікації текстів новин

Посилання: https://fakenewsua.streamlit.app/

Опис: Цей додаток допомагає автоматично виявляти фейки в українських новинах за допомогою нейромережі з архітектурою LSTM. Користувачі можуть ввести новину, вставивши URL-адресу або безпосередній текст. При введенні URL додаток перевіряє, чи посилання дійсне, і витягує вміст новини, якщо тільки джерело не заблоковане або недоступне. Після цього відбувається класифікація новини. Якщо новина визначається як фейкова, додаток також виділяє слова, що часто зустрічаються в ненадійних джерелах.