

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ
СІКОРСЬКОГО»**

**Інститут прикладного системного аналізу
Кафедра Штучного Інтелекту**

До захисту допущено:
Завідувач кафедри
_____ Олена ЧУМАЧЕНКО
«__» _____ 2023 р.

**Дипломна робота
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Системи і методи штучного
інтелекту»
спеціальності 122 «Комп'ютерні науки»
на тему: «Ідентифікація fake news за допомогою нейронних мереж»**

Виконав (-ла): _____
студент (-ка) IV курсу, групи КІ-93
Шафорост Анна Сергіївна

Керівник: _____
старший викладач Шаповал Наталія Віталіївна.

Консультант з нормконтролю: _____
фахівець першої категорії Гончарук Максим
Миколайович

Консультант з економічного розділу: _____
доцент, к.е.н., Рощина Надія Василівна

Рецензент: _____
доцент, к.ф.-м.н., доцент
Шубенкова Ірина Анатоліївна

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.
Студент (-ка) _____

Київ – 2023 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Інститут прикладного системного аналізу

Кафедра Штучного Інтелекту

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки»

Освітня програма «Системи і методи штучного інтелекту»

ЗАТВЕРДЖУЮ:

Завідувач кафедри

_____ Олена ЧУМАЧЕНКО

« ___ » _____ 2023 р.

ЗАВДАННЯ

на дипломну роботу студенту

Шафорост Анна Сергіївна

1. Тема роботи «Ідентифікація fake news за допомогою нейронних мереж», керівник роботи старший викладач, ст. викладач , Шаповал Наталія Віталіївна затверджені наказом по університету від «26» травня 2023 р. № 1344-с
2. Термін подання студентом роботи 08.06.2023
3. Вихідні дані до роботи – набір текстових даних новин
4. Зміст роботи – 1. Дослідження предметної області; 2. – Математичні основи робота; 3. Аналіз результатів роботи; 4. Функціонально-вартісний аналіз програмного продукту.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	доцент, к.е.н., Рощина Н.В.		

7. Дата видачі завдання 21.02.2023

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1.	Підготовка даних до роботи	27.02.2023	Виконано
2.	Вивчення літератури за темою роботи	25.03.2023	Виконано
3.	Підготовка першого розділу	10.04.2023	Виконано
4.	Розробка нейронних мереж	16.04.2023	Виконано
5.	Підготовка другого розділу	08.05.2023	Виконано
6.	Підготовка третього розділу	24.05.2023	Виконано
7.	Підготовка економічної частини	01.06.2023	Виконано
8.	Підготовка презентації доповіді	03.06.2023	Виконано
9.	Оформлення дипломної роботи	05.06.2023	Виконано

Студент

Анна ШАФОРСТ

Керівник

Наталія ШАПОВАЛ

АНОТАЦІЯ

Дипломна робота: 88 с., 14 рис., 9 табл., 3 додатки, 17 джерел

НЕЙРОННІ МЕРЕЖІ, РЕКУРЕНТНІ НЕЙРОННІ МЕРЕЖІ, КЛАСИФІКАЦІЯ, ДОВГА КОРОТКОЧАСНА ПАМ'ЯТЬ

Робота спрямована на вирішення зростаючого занепокоєння щодо поширення фейкових новин та їхнього потенційного впливу на громадську думку, соціальний дискурс і демократичні процеси. Використовуючи нейронні мережі, запропоновано рішення автоматизованого виявлення фейкових новин, що є внеском у ширшу сферу виявлення та запобігання дезінформації.

Об'єкт дослідження – класифікація текстових новин на предмет дезінформації.

Мета роботи – створити якісне рішення для швидкої ідентифікації fake news.

Ця робота робить цінний внесок у сферу ідентифікації фейкових новин, використовуючи можливості нейронних мереж. У ній детально описано використану методологію та представлено багатообіцяючі результати. У роботі визнаються потенційні обмеження і відкриваються шляхи для майбутніх досліджень. Однак ефективність запропонованого підходу тільки виграє від подальшої перевірки та порівняння з альтернативними методами, щоб встановити його перевагу у виявленні фейкових новин у різних контекстах і мовах.

ABSTRACT

Bachelor thesis: 88 pages, 14 figures, 9 tables, 3 appendices, 17 sources.

NEURAL NETWORKS, RECURRENT NEURAL NETWORKS, CLASSIFICATION, LONG SHORT-TERM MEMORY

The study aims to address the growing concern about the spread of fake news and its potential impact on public opinion, social discourse, and democratic processes. By utilizing neural networks, a solution for automated detection of fake news is proposed, contributing to the broader field of misinformation detection and prevention.

The research object is the classification of textual news for misinformation.

The objective of the study is to create a high-quality solution for rapid identification of fake news.

This work makes a valuable contribution to the field of fake news identification by leveraging the capabilities of neural networks. The methodology used is described in detail, and promising results are presented. The study acknowledges potential limitations and opens avenues for future research. However, the effectiveness of the proposed approach would benefit from further validation and comparison with alternative methods to establish its superiority in detecting fake news across different contexts and languages.

ЗМІСТ

Вступ	8
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1 Актуальність задачі з визначення fake news	10
1.2 Сучасні напрямки досліджень	11
1.3 Аналіз існуючих сервісів	13
1.4 Аналіз існуючих підходів	15
1.4.1 Обробка природної мови (Natural Language Processing, NLP) ..	15
1.4.2 Двосторонні нейронні мережі та LSTM	17
1.4.3 Ансамблеве навчання	17
1.4.4 Видобування ознак	19
1.4.5 Гібридні підходи	19
1.5 Постановка задачі	19
Висновки до розділу 1	21
РОЗДІЛ 2 ТЕОРЕТИЧНІ ТА МАТЕМАТИЧНІ ОСНОВИ РОБОТИ	22
2.1 Огляд двосторонніх нейронних мереж LSTM	22
2.2 Архітектура двосторонніх нейронних мереж LSTM(ДКЧП)	24
2.2.1 Традиційна ДКЧП	24
2.2.1 Застосування сигмоїди у комірці LSTM	26
2.3 Опис шарів двосторонньої LSTM мережі	28
2.3.1. Embedding рівень	28
2.3.2. Bidirectional LSTM рівень	29
2.3.3. Dense рівні	30
Висновки до розділу 2	31
РОЗДІЛ 3. ПОБУДОВА МОДЕЛІ ІДЕНТИФІКАЦІЇ FAKE NEWS НА ОСНОВІ МЕРЕЖІ LSTM	32
3.1 Архітектура моделі та вибір мови програмування	32
3.1.1 Мова програмування	32
3.1.2 Фреймворк	32

	7
3.2 Опис датасетів	33
3.3 Обробка даних	36
3.4 Навчання моделі	37
3.5 Результати роботи моделі	38
3.5.1 Тренування моделі	38
3.5.2 Перевірка моделі на тестових даних	39
3.5.3 Перевірка моделі на новині згенерованій Chat-GPT	41
Висновки до розділу 3	42
РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО	
ПРОДУКТУ	43
4.1 Постановка задачі проектування	43
4.2 Обґрунтування функцій програмного продукту	44
4.3 Обґрунтування системи параметрів програмного продукту	47
4.4 Аналіз експертного оцінювання параметрів	50
4.5 Аналіз рівня якості варіантів реалізації функцій	54
4.6 Економічний аналіз варіантів розробки ПП	55
4.7 Вибір кращого варіанту ПП техніко-економічного рівня	61
4.8 Висновки до четвертого розділу	63
ВИСНОВКИ	64
Список використаних джерел	65
ДОДАТОК А ПРОГРАМНИЙ КОД	67
ДОДАТОК Б НОВИНА З СНАТ-GPT	77
ДОДАТОК В ПРЕЗЕНТАЦІЯ	79

Вступ

У сучасному інформаційному суспільстві, де доступ до новин та інформації здійснюється швидко та легко, питання достовірності цих джерел стає надзвичайно важливим. Зростання кількості фальшивих новин (fake news) на сьогоднішній день стало однією з головних проблем, яка породжує недовіру до інформаційних ресурсів та може має негативний вплив на суспільство.

Метою цієї дипломної роботи є розробка системи, що базується на використанні нейронних мереж, для ідентифікації та розпізнавання fake news. Нейронні мережі володіють потужними можливостями в аналізі тексту та виявленні складних закономірностей, що дозволяє їм ефективно працювати з великими обсягами даних.

У рамках дослідження проведено аналіз існуючих методів та підходів до виявлення fake news, а також дослідження сучасних алгоритмів нейронних мереж, зокрема методів глибокого навчання, для створення ефективної моделі ідентифікації.

Після створення моделі, проведено експерименти на реальних наборах даних, що містять як достовірну, так і фальшиву інформацію, для оцінки ефективності та точності розробленої системи. Порівняно результати з іншими підходами до виявлення fake news з метою обґрунтування ефективності використання нейронних мереж.

Важливим аспектом роботи є розробка набору ознак та параметрів, які допоможуть відрізнити фальшиві новини від достовірних. Це можуть бути ознаки, пов'язані з використанням специфічних слів, стилістичних особливостей або незвичайних джерел новин. Також враховані соціальні та контекстуальні фактори, які можуть вказувати на можливість присутності fake news.

В результаті дослідження та розробки системи ідентифікації fake news з використанням нейронних мереж, розроблено модель, яка здатна виявляти та класифікувати фальшиві новини з високою точністю та швидкістю. Це може

стати важливим внеском у боротьбу з поширенням дезінформації та зміцненням довіри до інформаційних джерел.

РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність задачі з визначення fake news

Fake news - це неправдива або оманлива інформація що презентується як новина. Неправдиві новини часто мають на меті заплямувати репутацію об'єкту новини. Термін не має фіксованого визначення і широко застосовується до будь-якого типу неправдивої інформації. Крім того, дезінформація включає поширення неправдивої інформації зі шкідливими намірами, а іноді генерується та поширюється ворожими іноземними суб'єктами, особливо під час виборів. Поширеність фейкових новин зростає зі зростанням соціальних мереж, особливо стрічки новин Facebook. Поширенню фейкових новин сприяють декілька факторів, такі як політична поляризація, політика постправди, мотивовані міркування, підтверджувальне упередження (*confirmation bias*) та алгоритми соціальних мереж.

Дезінформація впливає на сприйняття правдивих новин конкуруючи з ними за увагу та прихильність читача. До прикладу, дослідження новинних ресурсів під час президентських виборів в США 2016 року показали, що найпопулярніші фейкові новини про вибори отримали більше уваги у Facebook, ніж головні новини великих ЗМІ.

Fake news є важливим елементом інформаційної війни - вони допомагають маніпулювати думками спільнот, дискредитувати державні органи управління, розпалювати ворожнечу - усе аби створити соціальне напруження та дестабілізувати політичні, економічні стосунки всередині країни та поза її межами.

За підрахунками EU vs Disinformation campaign за 2018 рік [1], Україна стала найбільшою ціллю дезінформації з боку росії з 461 новинними посиланнями із загальної кількості у 1000 випадків. Ця дезінформація була націлена на українського і російського споживач, та на міжнародну спільноту. У той час, дослідження КМІСу за 2018 рік показали, що третина українців не здатна відрізнити правдиву інформацію від fake news. [2]

Хоча fake news намагаються імітувати справжні новини, вони мають чіткі ознаки та характеристики, що відрізняють їх від правдивої інформації:

- Емоційність - апелювання до емоційного, особливо негативного, ускладнює неупереджений аналіз інформації. “Ці фальшиві, що здаються достовірними, новини «прив’язливі», оскільки вони захоплюють увагу людей, апелюючи до стереотипів і соціальних забобонів і експлуатуючи миттєві емоції, такі як занепокоєння, презирство, гнів і розчарування.” [3]
- Спрощення - неправдиві новини таргетують аудиторію, що не буде перевіряти джерела та якість інформації
- Короткочасність - фейкові новини не мають оновлень та продовження. Вони мають на меті лише миттєву маніпуляцію думкою спільноти.
- Швидкість поширення - поширення файкових новин значно вища за поширення правдивих новин

1.2 Сучасні напрямки досліджень

Сучасні дослідницькі підходи до виявлення фейкових новин можна класифікувати на різні методи, зокрема лінгвістичні, тематично-незалежні, підходи машинного навчання та гібридні підходи [4]. Ці методи спрямовані на виявлення фейкових новин за допомогою різних прийомів, таких як аналіз лінгвістичних особливостей, використання алгоритмів машинного навчання та застосування систем, заснованих на знаннях.

Лінгвістичний підхід: Цей підхід зосереджується на лінгвістичних особливостях новинних статей, таких як стилістичні особливості (наприклад, сегментація речень, токенізація та маркування частин мови) та лексичні особливості (наприклад, словниковий запас, частота вживання слів, відмінкові схеми). [5]. Аналізуючи ці ознаки, дослідники можуть виявити оманливі підказки або стилі написання, які часто асоціюються з фейковими новинами.

Тематично-агностичний підхід: Цей підхід спрямований на виявлення фейкових новин, не покладаючись на конкретну тему або тематику новинної

статті. Він зосереджується на загальних характеристиках фейкових новин, таких як структура, стиль і подача контенту.

Методи машинного навчання: Алгоритми машинного навчання широко використовуються в дослідженнях для виявлення фейкових новин. Ці алгоритми можуть навчатися на маркованих даних (новинних статтях, позначених як справжні або фейкові) для класифікації нових немаркованих новинних статей. Наприклад, для виявлення фейкових новин використовують методи глибокого навчання, такі як згорткові нейронні мережі (CNN) і довга короткочасна пам'ять (LSTM) [5].

Підхід, заснований на знаннях: Цей підхід використовує зовнішні джерела знань, такі як бази даних або графіки знань, для перевірки фактів, представлених у новинних статтях. За допомогою перехресних посилань на інформацію в статтях з надійними джерелами дослідники можуть визначити точність змісту новин.

Гібридний підхід: Гібридний підхід поєднує в собі два або більше з вищезгаданих методів, щоб покращити загальну ефективність виявлення фейкових новин. Наприклад, модель може використовувати як ознаки на основі контенту, так і інформацію про соціальний контекст, щоб підвищити точність виявлення фейкових новин [5].

Основною складністю у виявленні фейкових новин є раннє виявлення фальшивих новин. Більшість існуючих моделей виявляють фейкові новини після певного часу поширення, що в деяких випадках може бути занадто пізно [5]. Іншою проблемою є брак маркованих даних для навчання моделей виявлення. Для вирішення цієї проблеми дослідники запропонували використовувати слабкі методи моніторингу [5], які використовують зашумлені, обмежені або неточні джерела для моніторингу міток на великих обсягах навчальних даних.

Крім того, проблема зміни понять, коли базовий взаємозв'язок між вхідними даними та цільовими змінними змінюється з часом, може вплинути на ефективність моделей виявлення фейкових новин [5]. Дослідники вивчають

способи пом'якшення наслідків зміни понять та покращення адаптивності моделей до нових подій і розробок.

Таким чином, сучасні підходи до дослідження виявлення фейкових новин включають різні методи, такі як Аналіз мови, алгоритми машинного навчання та системи, засновані на знаннях. Дослідники також вирішують такі проблеми, як раннє виявлення, відсутність маркованих даних і зміна понять, щоб підвищити ефективність і адаптивність моделей виявлення фейкових новин.

1.3 Аналіз існуючих сервісів

За останнє десятиліття спостерігається швидкий приріст поширення фейкових новин. Таке поширення статей, які не відповідають фактам, викликає багато проблем, що не обмежуються лише політикою, але охоплюють різні інші галузі, такі як спорт, здоров'я і навіть наука. Однією з таких галузей, що постраждали від фейкових новин, є фінансові ринки, де чутки можуть мати катастрофічні наслідки і призвести до зупинки ринку.

Людська здатність приймати рішення в основному залежить від типу інформації, яку особи споживають; світогляд формується на основі інформації, яку людина усвідомлює. Є все більше доказів того, що споживачі дивно реагують на новини, які пізніше виявляються фейковими. Один з недавніх випадків - поширення нового коронавірусу, коли по Інтернеті поширювалися фейкові звіти про походження, характер і поведінку вірусу. Ситуація погіршувалась, коли все більше людей читали фейковий контент онлайн. Виявлення таких новин в Інтернеті - непросте завдання.

Існує кілька обчислювальних технік, які можуть використовуватися для позначення певних статей як фейкових на основі їхнього текстового змісту. Більшість цих технік використовують веб-сайти перевірки фактів, такі як "PolitiFact" та "Snopes". Є кілька репозиторіїв, підтримуваних дослідниками, що містять списки веб-сайтів, визнаних сумнівними та фейковими. Однак проблема з цими ресурсами полягає в тому, що для виявлення фейкових

статей/веб-сайтів потрібна людська експертиза. Ще важливіше, веб-сайти перевірки фактів містять статті з певних галузей, таких як політика, і не узагальнюються для виявлення фейкових новин з різних галузей, таких як розваги, спорт і технології.

На сьогодні існує декілька способів користувачу перевірити інформацію на правдивість: ШІ-рішення, плагіни для браузерів та фактчекінгові сайти.

- ШІ-рішення: технологічні компанії та соціальні мережі працюють над автоматичним виявленням неправдивої інформації за допомогою обробки природної мови, машинного навчання та мережевого аналізу. Проте, на даному етапі, виявлення неправдивої інформації за допомогою ШІ має свої обмеження: недостатнє розуміння контексту у тексті та складність у визначенні характеру контенту.

- Плагіни для браузера: Деякі організації розробили плагіни для браузера, щоб позначати фейкові новини та джерела. Два приклади цього включають:

- B.S. Detector: Утримується The Self Agency, LLC та працює на основі OpenSources - професійно підбраного списку ненадійних або сумнівних джерел.
- This is Fake: Створений Slate.com, фейкові новини виявляються за допомогою колективного і ручного модерування співробітниками та співавторами Slate.

- Сайти факт-чекінгу: Зростання популярності фейкових новин призвело до збільшення кількості сайтів факт-чекінгу, які мають на меті перевіряти заяви, що робляться ЗМІ та політиками, та розкривати фейкові новини. Деякі рекомендовані сайти включають:

FactCheck.org, Snopes, Politifact, Washington Post Fact Checker, Слово і діло, Stop Fake, VoxCheck

При використанні цих ресурсів важливо критично оцінювати їхню достовірність та упередженість, щоб забезпечити точну інформацію.

Важливо зазначити, що рішення на основі штучного інтелекту та плагіни для браузера можуть бути не ідеальними, і може все ще бути необхідна людська

інтервенція для перевірки автентичності інформації. Тому важливо розвивати навички критичного мислення та вміння впізнавати якісну журналістику.

1.4 Аналіз існуючих підходів

1.4.1 Обробка природної мови (Natural Language Processing, NLP)

Обробка природної мови (Natural Language Processing, NLP) - це підгалузь штучного інтелекту, яка займається взаємодією між комп'ютерами та людьми за допомогою природної мови. Він дозволяє комп'ютерам розуміти, інтерпретувати та генерувати людську мову таким чином, щоб вона була цінною та змістовною. У контексті виявлення фейкових новин NLP відіграє вирішальну роль в аналізі та класифікації текстових даних для визначення достовірності інформації. NLP може використовуватися для виявлення аномалій у текстових статтях, які можуть вказувати на обман. Ця техніка включає аналіз текстової структури та змісту статей для визначення, чи є вони фактами або є обманом.

Підхід НЛП до виявлення фейкових новин зазвичай включає наступні кроки:

1. Збір даних: Збір масиву даних новинних статей, включаючи як справжні, так і фейкові новини.
2. Попередня обробка: Очищення та підготовка текстових даних до аналізу.

Цей процес може включати

- a. Видалення спеціальних символів, цифр і розділових знаків
 - b. Перетворення всього тексту в малі літери
 - c. Розбиття тексту на слова або фрази
 - d. Видалення стоп-слів, таких як "and", "the" і "in"
 - e. Перетворення слів у їхні базові форми для використання.
3. Вилучення ознак: Перетворення попередньо оброблених текстових даних у формат, який можна використовувати як вхідні дані для алгоритмів машинного навчання. До поширених методів вилучення ознак належать:

- a. Пакет слів (Bag of Words, BoW): Представлення тексту у вигляді набору слів із зазначенням їхньої частоти вживання
 - b. Метод частоти, оберненої до частоти документа (TF-IDF): Зважування важливості слів на основі їхньої частоти в документі та їхньої рідкості в усьому наборі даних
 - c. Вбудовування слів, такі як Word2Vec або GloVe: Представлення слів у вигляді векторів у багатовимірному просторі, що фіксує семантичні зв'язки між словами в напрямку.
4. Навчання моделі: Використання витягнутих ознак для навчання моделі машинного або глибокого навчання для класифікації новинних статей як справжніх або фейкових. Популярні алгоритми для цього завдання включають
- a. Логістична регресія
 - b. Наївний Байєс
 - c. Машини опорних векторів (SVM)
 - d. Дерева рішень
 - e. Випадкові ліси
 - f. Моделі глибокого навчання, такі як згорткові нейронні мережі (CNN) та рекурентні нейронні мережі (RNN).
5. Оцінювання: Оцінка продуктивності навченої моделі за допомогою різних оціночних метрик, таких як точність, точність, відгук і F1-score.
6. Розгортання: Впровадження навченої моделі в реальний додаток, такий як розширення для браузера, агрегатор новин або платформа соціальних мереж, для автоматичного виявлення та позначення фейкових новинних статей.

Отже, підхід до виявлення фейкових новин на основі НЛП передбачає попередню обробку текстових даних, виокремлення ознак, навчання моделі класифікації, оцінку її продуктивності та розгортання моделі для реальних

додатків. Цей процес допомагає виявляти і відфільтровувати фейкові новини, забезпечуючи користувачам доступ до надійної і точної інформації.

1.4.2 Двосторонні нейронні мережі та LSTM

Мережі з довгою короткочасною пам'яттю (LSTM) - це тип рекурентних нейронних мереж (RNN), розроблений для вирішення проблеми короткочасної пам'яті в RNN. Вони можуть навчатися і підтримувати довгострокові залежності, використовуючи внутрішні механізми, які називаються шарами або гейтами, що регулюють потік інформації. Ці шари можуть вирішувати, які дані в послідовності необхідно зберегти, а які відкинути, що дозволяє LSTM передавати відповідну інформацію через довгий ланцюжок послідовностей для здійснення прогнозів.

LSTM має чотири стани: забути, вхід, вихід і кандидат. Стан забуття вирішує, яку інформацію з попереднього стану комірки слід забути. Вхідні стани визначають, яку нову інформацію з поточного входу слід додати до стану комірки. Стани-кандидати зберігають можливі значення, які можна додати до стану комірки. Вихідний стан обчислює вихідні дані на основі оновленого стану комірки.

LSTM та їхні різновиди, такі як Gated Recurrent Units (GRU), використовуються в різних сучасних додатках, таких як розпізнавання мови, синтез мови та розуміння природної мови.

Ці моделі глибокого навчання можуть бути навчені виявляти фейкові новини у заданому корпусі новин. Наприклад, двостороння нейронна мережа може використовуватися для аналізу тексту в обох напрямках - вперед та назад, що допомагає моделі краще розуміти контекст та семантику тексту.

1.4.3 Ансамблеве навчання

Ансамблеве навчання - це підхід у машинному навчанні, коли кілька моделей об'єднуються для покращення загальної ефективності вирішення

проблеми. У контексті виявлення неправдивих новин ансамблеве навчання можна використовувати для створення більш точної та надійної моделі, поєднуючи сильні сторони різних класифікаторів. Основна ідея полягає в тому, щоб навчити кілька моделей на наборі даних, а потім об'єднати їхні прогнози для отримання остаточного рішення.

Алгоритм застосування ансамблевого навчання для виявлення фейкових новин наступний:

1. Підготовка набору даних: Набір даних повинен складатися з маркованих прикладів справжніх і фейкових новин. Кожна стаття буде представлена текстом і цільовою міткою, яка вказує, чи є вона справжньою або фейковою.
2. Попередня обробка тексту: Попередня обробка текстових даних шляхом перетворення їх у малі літери, видалення розділових знаків і стоп-слів. Це допомагає зменшити розмірність і шум у даних.
3. Виділення ознак: використовується метод `CountVectorizer` для перетворення текстових даних у числові характеристики. `CountVectorizer` токенизує текст і підраховує частоту кожного слова.
4. Поділ набору даних: поділ набору даних на навчальний і тестовий набори. Це допоможе оцінити продуктивність ансамблевої моделі.
5. Навчання окремих моделей: навчання різних класифікаторів на попередньо оброблених даних. Наприклад, можна використовувати класифікатори логістичної регресії, дерева рішень та випадкового лісу.
6. Об'єднання моделей: Після навчання окремих моделей потрібно об'єднати їхні прогнози, використовуючи такі методи, як голосування більшістю голосів, зважене голосування або підсумовування. Це дозволить створити ансамблеву модель, яка використовує сильні сторони окремих моделей.
7. Оцінка ансамблевої моделі: Оцінка продуктивності ансамблевої моделі на тестовому наборі. Це допоможе зрозуміти ефективність підходу ансамблевого навчання для виявлення фейкових новин.

Техніки ансамблевого навчання, такі як "беггінг" та "бустінг", можуть використовуватися для навчання кількох алгоритмів машинного навчання та покращення загальної продуктивності виявлення фейкових новин. Шляхом поєднання різних алгоритмів та використання різних методів ансамблю можна створити більш надійну та точну модель.

1.4.4 Видобування ознак

Видобування лінгвістичних ознак, таких як n-грами, стиль письма та соціальний контекст (наприклад, позиція та поширення), з текстових статей може допомогти навчати моделі машинного навчання для виявлення фейкових новин. Моделі, такі як k-найближчий сусід (KNN), метод опорних векторів (SVM), логістична регресія (LR).

1.4.5 Гібридні підходи

Поєднання текстових ознак з додатковою інформацією, такою як соціальна взаємодія користувачів у соціальних медіа, може призвести до кращої точності виявлення фейкових новин. Цей підхід включає аналіз соціальної відповіді на статтю разом із текстовими ознаками.

1.5 Постановка задачі

Завдання полягає в розробці системи виявлення фейкових новин з використанням нейронних мереж LSTM (Long Short-Term Memory - довга короткочасна пам'ять). Метою є створення моделі, яка може ефективно ідентифікувати та класифікувати новинні статті як справжні чи фейкові на основі їхнього текстового змісту. Система покликана надавати точні та надійні прогнози, щоб допомогти користувачам відрізнити надійні джерела новин від потенційно оманливої або сфабрикованої інформації.

Маючи набір даних, що складається з новинних статей, позначених як справжні або фейкові, задача полягає в навчанні моделі нейронної мережі LSTM, щоб вивчити основні закономірності та ознаки, які відрізняють справжні новини від неправдивих. Архітектуру LSTM обрано через її здатність фіксувати довгострокові залежності в послідовних даних, що робить її придатною для обробки текстової природи новинних статей.

Вхідні дані для моделі складаються з текстового вмісту новинних статей, який може включати заголовок статті, основний текст і будь-які додаткові метадані, такі як тема або джерело. Модель обробляє цей текстовий вхід за допомогою шарів LSTM, які дозволяють мережі зберігати та оновлювати інформацію з часом. Навчаючись на великому наборі даних маркованих прикладів, модель має набути здатності ідентифікувати лінгвістичні підказки, контекстну інформацію та стилістичні патерни, які відрізняють справжні новини від фейкових.

Завданням моделі є виведення бінарного класифікаційного прогнозу, який вказує, чи є дана новина справжньою або фейковою.

Після навчання система виявлення фейкових новин на основі LSTM може бути використана для оцінки автентичності нових, ще не бачених новинних статей. Отримавши вхідну статтю, модель обробляє її текстовий вміст і надає прогнозну оцінку або ймовірність, що вказує на ймовірність того, що вона є фейковою. Для визначення остаточного класифікаційного рішення може бути застосований певний поріг, коли оцінка, вища за поріг, вказує на фейкові новини, а нижча - на справжні новини.

Ефективність системи виявлення фейкових новин на основі LSTM оцінюється за різними показниками, такими як accuracy, precision, recall, та F1-score. Ці показники вимірюють здатність системи правильно класифікувати новинні статті та уникати хибнопозитивних або хибнонегативних результатів. Ефективність узагальнення моделі оцінюється за допомогою окремих тестових даних, які не використовувалися під час навчання, що гарантує її здатність точно виявляти фейкові новини в реальних умовах.

Висновки до розділу 1

У цьому розділі розглянуті питання актуальності задачі з визначення fake news, сучасного вигляду предметної області. Проаналізовані існуючі підходи до вирішення поставленої задачі та актуальні сервіси з виявлення неправдивих новин.

Згідно з дослідженнями ефективності різних методів машинного навчання для детекції фейкових новин, згорткові нейронні мережі (Convolutional Neural Networks, CNN) краще справляються з видобуванням локальних та позиційно-інваріантних ознак, тоді як LSTM-RNN (Long Short-Term Memory Recurrent Neural Networks) підходить для класифікації на основі довгострокових семантичних залежностей. RNN працюють краще для завдань, де послідовне моделювання має більше значення. Результати показують, що модель з двонапрямковим LSTM-RNN значно ефективніша за однонапрямкові моделі. Експерименти також показують, що вибір алгоритму адаптивної швидкості навчання відіграє важливу роль у вирішенні проблеми зниклих градієнтів у RNN. [6]

РОЗДІЛ 2 ТЕОРЕТИЧНІ ТА МАТЕМАТИЧНІ ОСНОВИ РОБОТИ

2.1 Огляд двосторонніх нейронних мереж LSTM

Long short-term memory (LSTM, або довга короткочасна пам'ять ДКПЧ) - це штучна нейронна мережа, яка використовується у галузях штучного інтелекту та глибинного навчання. На відміну від стандартних нейронних мереж з прямим поширенням, у LSTM є зворотні зв'язки. Така рекурентна нейронна мережа (RNN) може обробляти не лише окремі дані (наприклад, зображення), але й цілі послідовності даних (наприклад, мовлення чи відео). [7] Ця характеристика робить мережі LSTM ідеальними для обробки та прогнозування даних. Наприклад, LSTM застосовується для завдань, таких як розпізнавання зв'язного рукопису без сегментації, розпізнавання мови, машинний переклад, виявлення мовної активності, керування роботами, відеоігри та охорона здоров'я.

Назва LSTM посилається на аналогію з тим, що стандартна RNN має як "довготривалу пам'ять", так і "короткотривалу пам'ять". Ваги з'єднань та зміщення в мережі змінюються один раз під час кожного епізоду навчання, аналогічно тому, як фізіологічні зміни синаптичної сили зберігають довгострокову пам'ять; патерни активації в мережі змінюються один раз на кожному кроці часу, аналогічно тому, як моментальні зміни в електричних залпах мозку зберігають короткострокову пам'ять. Архітектура LSTM має на меті забезпечити короткострокову пам'ять для RNN, яка може тривати тисячі кроків часу, отже "довготривала короткострокова пам'ять". [7]

Звичайна одиниця LSTM складається з комірки, входних гейтів, вихідних шарів та шарів забуття. Клітина запам'ятовує значення протягом довільних інтервалів часу, а три ворота регулюють потік інформації в клітину та з неї. На рис. 2.1 зображено описану архітектуру LSTM. Гейти забуття вирішують, яку інформацію відкинути з попереднього стану, призначаючи попередній стан, порівняний з поточним вводом, значення від 0 до 1. Показник 1 означає збереження інформації, а значення 0 - її відкидання. Вхідні шари

вирішують, які частини нової інформації зберегти в поточному стані, використовуючи ту саму систему, що й шари забуття. Вихідні шари контролюють, яку частину інформації в поточному стані виводити, призначаючи значення від 0 до 1 для інформації, з урахуванням попереднього та поточного станів. Вибіркове виведення відповідної інформації з поточного стану дозволяє мережі LSTM зберігати корисні довгострокові залежності для прогнозування, як в поточних, так і в майбутніх кроках часу.[9][10]

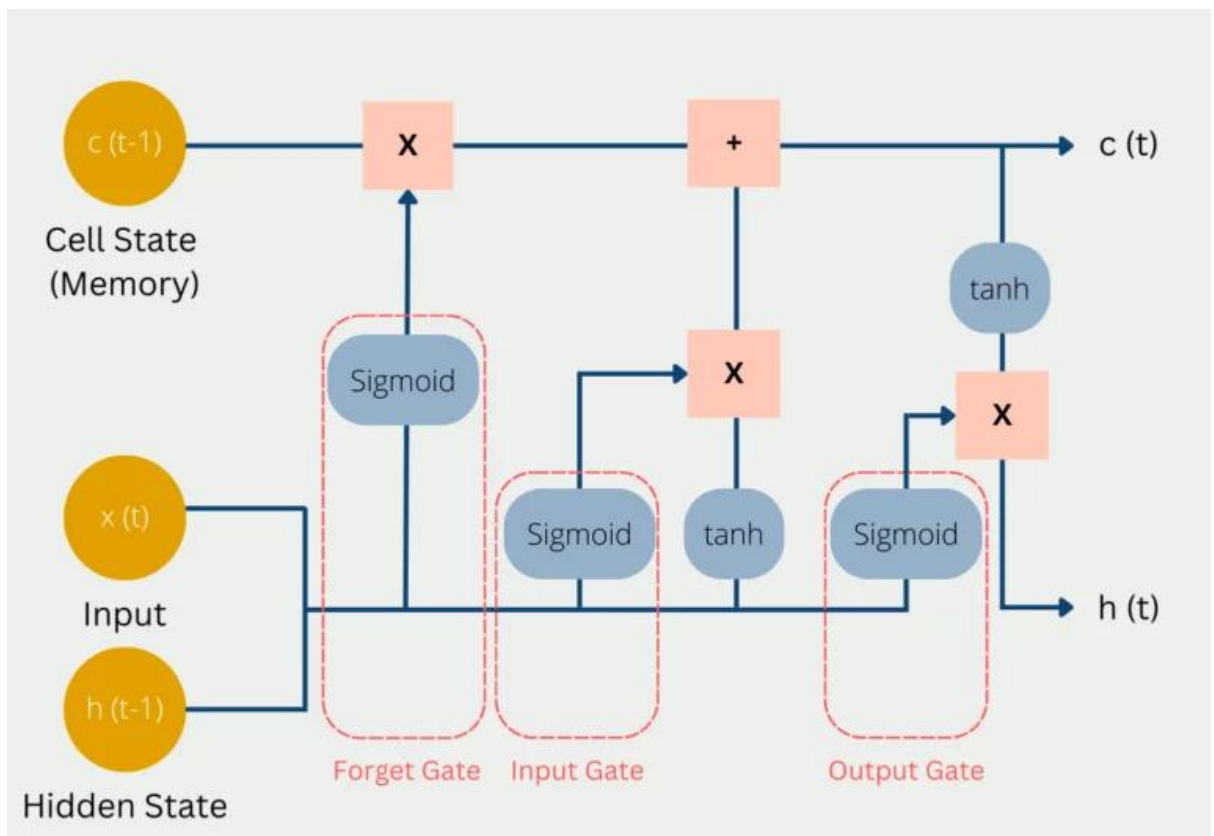


Рисунок 2.1 - Архітектура LSTM [13]

Мережі LSTM добре підходять для класифікації, обробки та прогнозування на основі часових рядів даних, оскільки між важливими подіями в часовому ряді можуть бути затримки невідомої тривалості. LSTM були розроблені для вирішення проблеми зникнення градієнту, яка може виникати при навчанні традиційних RNN. Відносна нечутливість до довжини прогалин - це перевага LSTM порівняно з RNN, моделями прихованих марковських процесів та іншими методами вивчення послідовностей в багатьох застосуваннях.[11]

2.2 Архітектура двосторонніх нейронних мереж LSTM(ДКЧП)

Розширені згортково-рекурентні нейронні мережі Чебишева - це штучні нейронні мережі, які включають вузли ДКЧП замість або на додаток до інших вузлів мережі. Вузол ДКЧП - це рекурентний вузол нейронної мережі, який відрізняється тим, що зберігає значення протягом тривалого або короткого періоду часу. Ключ до цієї здатності полягає в тому, що він не використовує функції активації у своїх рекурентних компонентах. В результаті, коли для навчання використовується рекурентне поширення в часі, ні збережене значення, ні градієнт, ні член вини не мають тенденції до розмивання з часом.

Вузли ДКЧП часто включаються в "блоки" інших вузлів ДКЧП. Такий дизайн характерний для "глибоких" багат шарових нейронних мереж і полегшує паралельну апаратну реалізацію.

Блоки ДКЧП мають три або чотири гейти, які вони можуть використовувати для регулювання руху даних в пам'ять та з пам'яті. Для створення цих шарів використовується логістична функція, яка обчислює значення від 0 до 1. Можна частково дозволити або заборонити проходження інформації до або з цієї пам'яті шляхом множення на ці величини. Наприклад, "вхідні шари" регулюють ступінь, до якого нове значення потрапляє в пам'ять. "Комірка забування" регулює, як довго значення зберігається в пам'яті. Крім того, "комірка виходу" регулює, скільки значення з пам'яті використовується для обчислення активації виходу блоку. У деяких системах вхідні вентиля та вентиля забування можуть бути об'єднані в один. Їх поєднання ґрунтується на гіпотезі, що забування відбувається щоразу, коли з'являється нове значення, яке варто запам'ятати).

2.2.1 Традиційна ДКЧП

Традиційна ДКЧП із забувальними вузлами позначає добуток Адамара (поелементний добуток).

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c)$$

$$h_t = o_t \circ \sigma_h(c_t)$$

Змінні

- x_t : входовий вектор
- h_t : вихідний вектор
- c_t : вектор стану комірки
- W, U, b : матриці та вектор параметрів (W - вага, U - уточнення, b - упередження)
- f_t, i_t, o_t : вектори вентилів
 - f_t : Вектор забувального вентиля. Вага пам'ятання старої інформації.
 - i_t : Вектор вхідного вентиля. Вага отримання нової інформації.
 - o_t : Вектор вихідного гейту. Кандидатність на вихід.

Функції активації

- σ_g : сигмоїдна функція
- σ_c : гіперболічний тангенс
- σ_h : гіперболічним тангенс

На рис. 2.2 зображено структуру комірки LSTM мережі і рівняння, що описують гейти комірки LSTM [12].

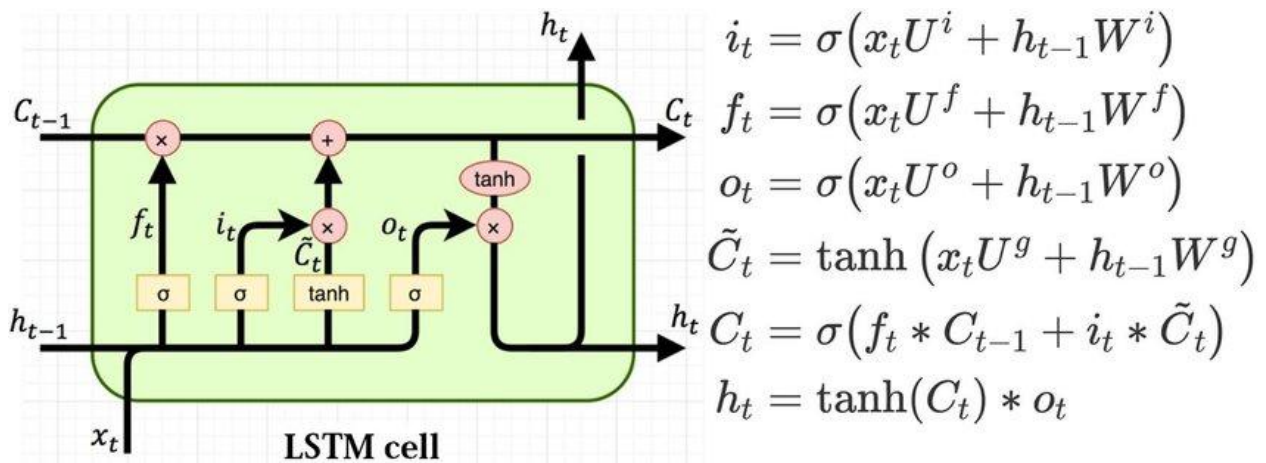


Рисунок 2.2 - Структура комірки LSTM мережі

Сигмоїдний шар - це вхідний гейт, який вирішує, яку нову інформацію буде додано до комірки. Він отримує два входи h_{t-1} та x_t . Шар \tanh створює вектор C_t нових значень-кандидатів. Разом ці два шари визначають інформацію, яка буде зберігатися у стані комірки. Їхнє поелементне множення ($i_t * C_t$) показує нам кількість інформації, яку потрібно додати до стану комірки. Результат додається до результату гейта забування, помноженого на попередній стан комірки ($f_t * C_{t-1}$), щоб отримати поточний стан комірки C_t . Далі обчислюється вихід комірки за допомогою сигмоїдного та тангенціального шарів. Сигмоїдний шар вирішує, яка частина стану комірки буде присутня у виході, тоді як тангенс шар зсуває вихід в діапазоні $[-1,1]$. Результати цих двох шарів перемножуються поелементно, щоб отримати вихід h_t комірки.

2.2.1 Застосування сигмоїди у комірці LSTM

Сигмоїдна функція - це математична функція, що має характерну "S"-подібну криву або сигмоїдну криву.

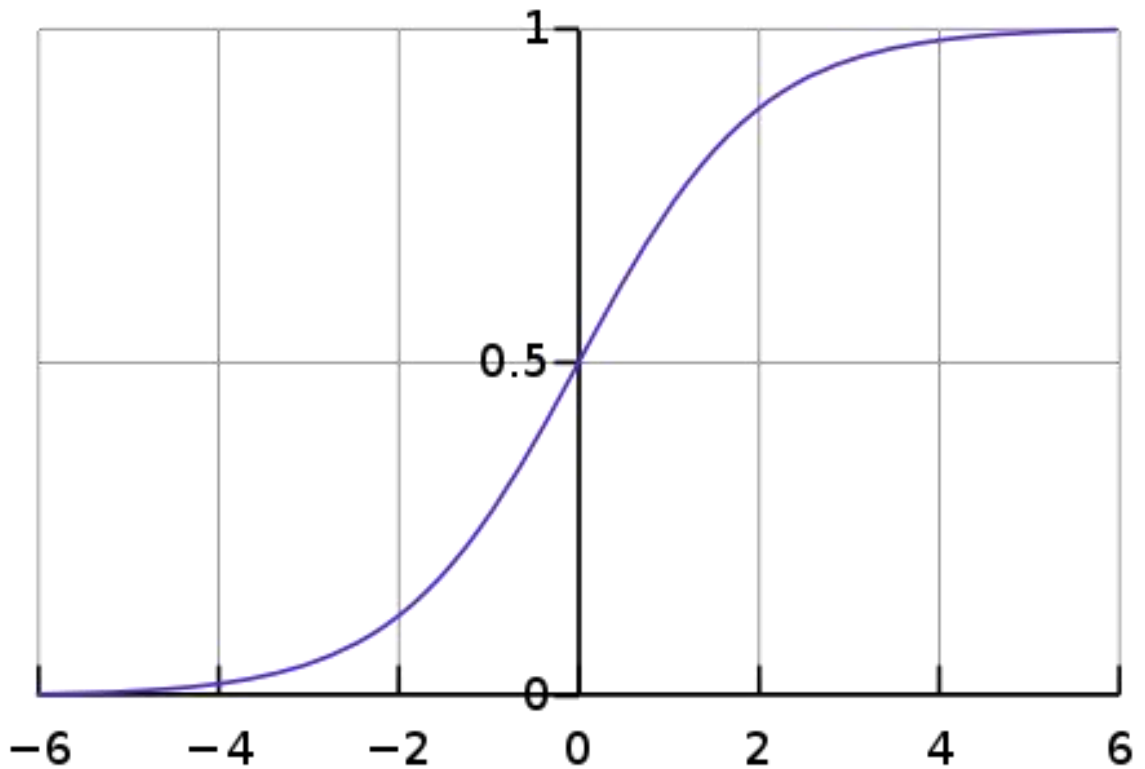


Рисунок 2.3 - Графік сигмоїди

Загалом, сигмоїдна функція є монотонною і має першу похідну, яка має дзвоноподібну форму. І навпаки, інтеграл від будь-якої неперервної, невід'ємної, дзвоноподібної функції (з одним локальним максимумом і без локального мінімуму, якщо вона не вироджена) буде сигмоїдальним. Таким чином, кумулятивні функції розподілу для багатьох поширених розподілів ймовірностей є сигмоїдальними.

Сигмоїдна функція обмежена парою горизонтальних асимптот при $x \rightarrow \pm \infty$.

Сигмоїда застосовується в нейронних мережах аби внести певну нелінійність в мережу, та, при цьому, не деформувати результати мережі.

Однією з причин використання сигмоїди в нейронних мережах є те, що її похідні легко представляються самою функцією:

$$\sigma'(x) = \sigma(x) \cdot (1 - \sigma(x))$$

Ця властивість зменшує складність обчислення методу зворотного поширення помилки, що робить її зручною у практичному використанні.

Ще одним вагомим мотивом для впровадження нелінійності є математично обґрунтована можливість досягнення точного наближення до будь-якої неперервної функції багатьох змінних за допомогою операцій додавання і множення чисел, композиції функцій, лінійних функцій та будь-якої неперервної нелінійної функції в ряді змінних, що використовується для отримання змінної.

2.3 Опис шарів двосторонньої LSTM мережі

У нейронній мережі шар - це набір нейронів, які виконують певне завдання. Нейронні мережі мають кілька шарів взаємопов'язаних нейронів, і кожен шар виконує певну функцію.

Залежно від положення в нейронній мережі розрізняють три типи шарів:

- Вхідний шар - відповідає за отримання вхідних даних і передачу їх наступному шару. Це перший шар у нейронній мережі
- Приховані шари - можна знайти майже у всіх типах нейронних мереж, окрім деяких одношарових типів, таких як персептрон. У нейронній мережі може бути декілька прихованих шарів. Кількість прихованих шарів і кількість нейронів у кожному шарі може змінюватися залежно від складності задачі, що вирішується
- Вихідний шар - останній шар у нейронній мережі, який виробляє кінцевий результат або прогноз

2.3.1. Embedding рівень

Вбудований рівень - це тип прихованого шару в нейронній мережі. Цей шар відображає вхідну інформацію з простору високої розмірності в простір низької розмірності, що дозволяє мережі дізнатися більше про взаємозв'язок між вхідними даними і обробляти їх більш ефективно. Наприклад, в обробці

природної мови (NLP) слова і фрази часто представляються як одномірні вектори, де кожен вимір відповідає окремому слову в словнику. Ці вектори є багатовимірними і розрідженими, що ускладнює роботу з ними. Замість того, щоб використовувати ці високовимірні вектори, шар вбудовування може зіставити кожне слово з низьковимірним вектором, де кожен вимір представляє певну особливість слова[14].

Шар вбудовування - це таблиця пошуку, яка вивчає значення кожного слова в контексті задачі, що розв'язується. Тут зіставляється кожне унікальне слово у словнику з вектором фіксованого розміру. Ці вектори відображають семантичне значення слів.

2.3.2. Bidirectional LSTM рівень

Двонаправлений шар LSTM використовується для обробки вхідної послідовності в прямому та зворотному напрямках. Він допомагає моделі збирати інформацію з усієї вхідної послідовності, а не тільки з останніх подій.

Як правило, у звичайній мережі LSTM отримується вихід безпосередньо, як показано на рис. 2.4, але у двонаправленій мережі LSTM вихід прямого та зворотного шару на кожному етапі передається на активаційний шар, який є нейронною мережею, і враховується вихід цього активаційного шару. Цей вихід містить інформацію або відношення між минулим і майбутнім словом.

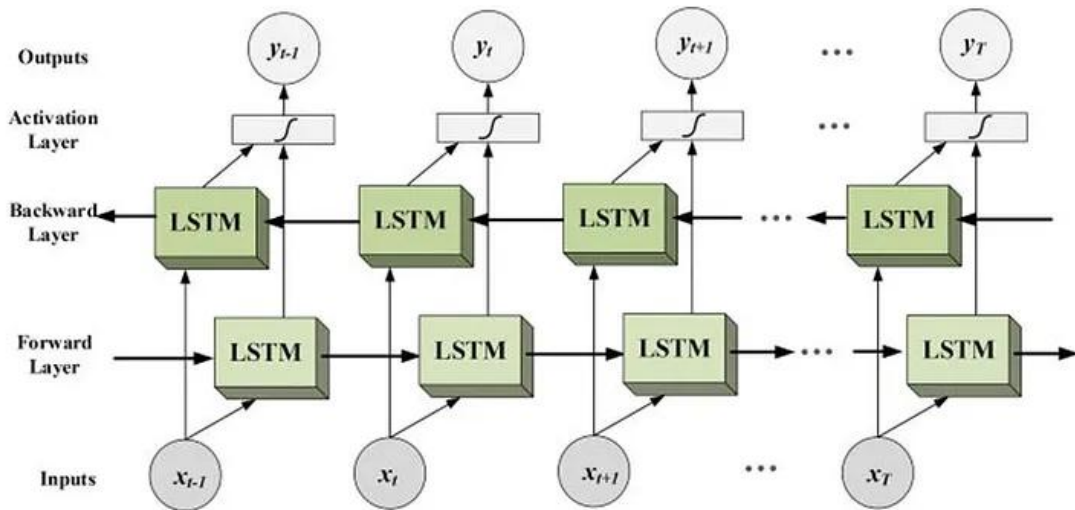


Рисунок. 2.4 - Схема двонаправленого шару [15]

2.3.3. Dense рівні

У будь-якій нейронній мережі щільний шар - це шар, який глибоко пов'язаний з попереднім шаром, тобто нейрони цього шару пов'язані з кожним нейроном попереднього шару. Цей шар є найбільш часто використовуваним шаром у штучних нейронних мережах.

Нейрон щільного шару в моделі отримує вихід від кожного нейрона попереднього шару, де нейрони щільного шару виконують матрично-векторне множення. Матрично-векторне множення - це процедура, при якій вектор-рядок виходу з попередніх шарів дорівнює вектору-стовпцю щільного шару.[16]

Для цієї моделі використано 2 шари dense:

1. Dense(128, activation='relu'): Повністю зв'язний шар з 128 прихованими елементами та функцією активації Rectified Linear Unit (ReLU).
2. Dense(1, activation='sigmoid'): Останній повністю зв'язний шар з однією вихідною одиницею та сигмоїдною функцією активації. На виході - двійкова класифікація (0 або 1), що представляє остаточний прогноз моделі.

Dense шари використовуються для перетворення вихідних даних попередніх шарів у формат, придатний для класифікації. Перший щільний шар застосовує функцію активації ReLU до виходу, тоді як другий щільний шар застосовує функцію активації Sigmoid для отримання остаточного прогнозу

Висновки до розділу 2

У цьому розділі було розглянуто теоретичну основу нейронних мереж з довгою короткочасною пам'яттю та їхнє застосування до задачі обробки і класифікації текстових даних.

Викладено архітектуру мереж ДКЧП та її окремої комірки: вхідної, вихідної та забувальної. Описано алгоритми, що використовуються у вищезгаданих комірках та їхній вплив на результат навчання моделі.

Також розглянуто рівні мережі LSTM: вбудований рівень, двонаправлений LSTM та dense рівні.

РОЗДІЛ 3. ПОБУДОВА МОДЕЛІ ІДЕНТИФІКАЦІЇ FAKE NEWS НА ОСНОВІ МЕРЕЖІ LSTM

3.1 Архітектура моделі та вибір мови програмування

У розділі 1 було проаналізовано різні підходи до розв'язку задачі класифікації текстових даних та обрано метод нейронних мереж LSTM за результатами аналізу. Варто нагадати, що мережа LSTM краще підходить для аналізу фейкових новин, ніж інші методи аналізу текстів, завдяки своїй здатності аналізувати складні патерни в текстових даних.

Тепер, розглянемо етапи реалізації задачі детальніше.

3.1.1 Мова програмування

За мову програмування було обрано Python версії 3.10, адже це остання стабільна версія. Python - це високорівнева інтерпретована мова програмування, яка широко використовується в програмах штучного інтелекту та машинного навчання. Це популярна мова для розробки нейронних мереж завдяки своїй простоті та наявності потужних бібліотек, таких як TensorFlow, PyTorch та Keras.

3.1.2 Фреймворк

В основі програмного продукту використано фреймворк TensorFlow, оскільки він є спеціально розробленим під задачі машинного навчання, а тому пришвидшує та спрощує етап створення та навчання моделі.

TensorFlow - це фреймворк машинного навчання з відкритим вихідним кодом, розроблений компанією Google. Він є незамінним для побудови та розгортання моделей машинного навчання, зокрема глибоких нейронних мереж. TensorFlow надає повний набір інструментів, бібліотек і ресурсів, які дозволяють дослідникам і розробникам створювати і навчати різні типи моделей машинного навчання.

В основі TensorFlow лежить концепція тензорів - багатовимірних масивів. Ці тензори представляють дані, що проходять через обчислювальні графи, які будує TensorFlow. Обчислювальні графи складаються з вузлів, які виконують математичні операції над тензорами, а ребра представляють потік даних між цими вузлами. Цей графовий підхід дозволяє TensorFlow ефективно розподіляти обчислення між різними пристроями, такими як CPU, GPU і навіть спеціалізованим обладнанням, таким як Tensor Processing Units (TPU).

TensorFlow пропонує високорівневі API, які полегшують побудову моделей машинного навчання, наприклад, Keras, що надає зручний інтерфейс для визначення та навчання нейронних мереж. Він також надає API нижчого рівня, які дають користувачам більше контролю та гнучкості над архітектурою моделі та процесом навчання.

3.2 Опис датасетів

Для цієї роботи було використано ISOT Fake News датасет [17], що містить приблизно 20,000 правдивих та неправдивих новин. Розподіл кількості правдивих та неправдивих новин представлено на рис. 3.1.

Правдиві новини зібрані з видання Reuters.com, неправдиві - з декількох сайтів визнаних ненадійними факт-чекінговою організацією Politifact.

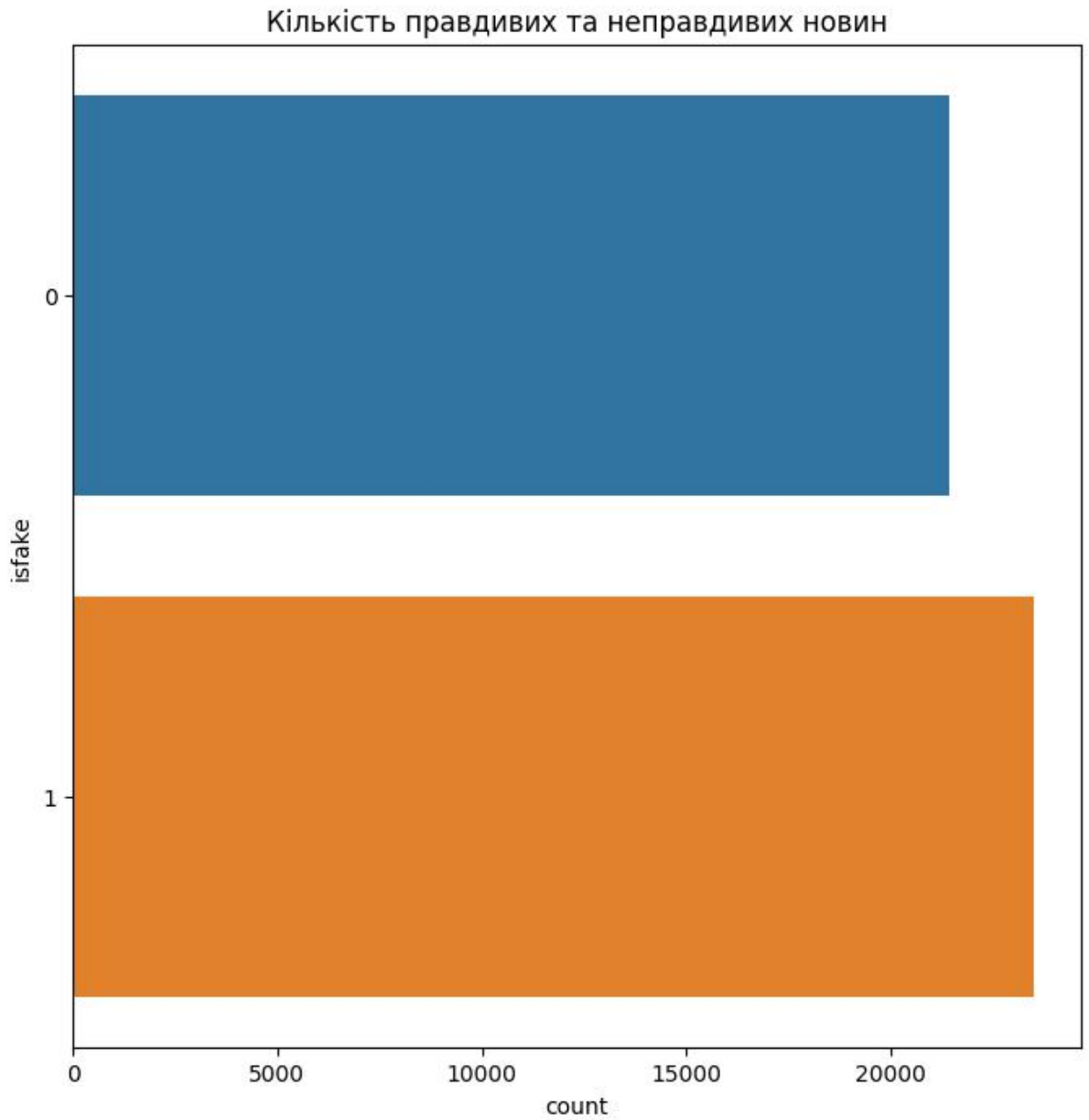


Рисунок 3.1 - Підрахунок новин за властивістю isFake

Датасети мають наступний вигляд: “назва статті, текст статті, тип новини, дата публікації”. Рис. 3.2 представляє вигляд датасету.

	title	text	subject	date
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	December 31, 2017
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	December 30, 2017
4	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNews	December 29, 2017

Рисунок 3.2 - Вигляд датасету

Датасети також мають властивість “тема новин”. Рис. 3.3 показує розподіл новин за темою.

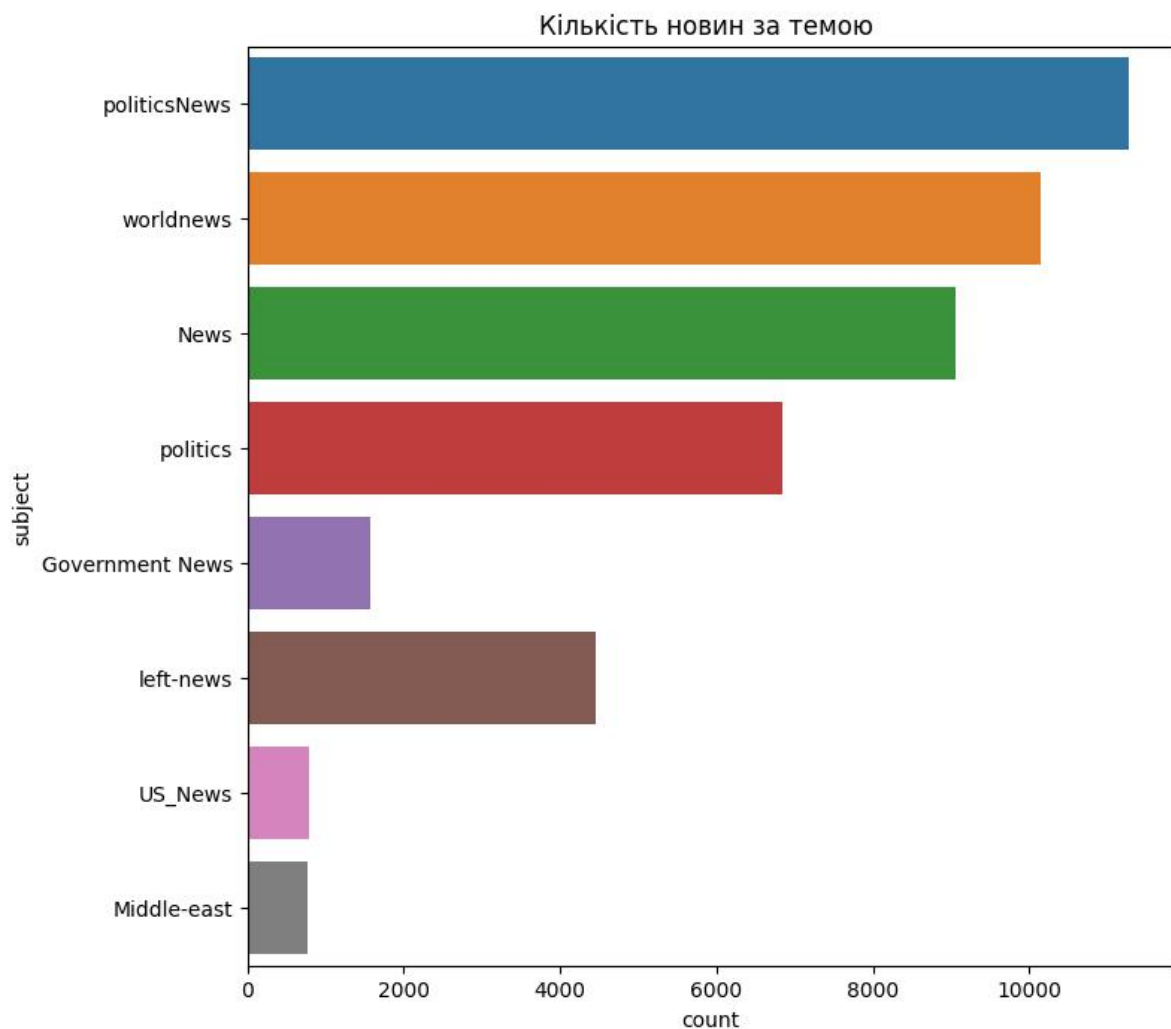


Рисунок 3.3

3.3 Обробка даних

У цій частині розглянемо важливі процедури підготовки набору даних для використання мережі LSTM для класифікації текстів. Серед необхідних кроків - вилучення стоп-слів, токенизація тексту та набивка. Ці процеси допоможуть підготувати дані для подальшої обробки за допомогою моделі глибокого навчання.

- Прибирання стоп-слів

Стоп-слова, такі як "the", "and", "is", "in" та подібні вирази, є часто вживаними словами, які не надають корисної інформації для класифікації тексту. Вони можуть вносити шум в аналіз, оскільки часто зустрічаються в тексті. Можна використовувати попередньо створені списки стоп-слів або створити унікальний список залежно від набору вхідних даних, щоб усунути стоп-слова. Список стоп-слів було взято з пакету NLTK - Natural Language Toolkit.

- Токенізація тексту

Текст розбивається на токени або слова за допомогою процесу токенизації. Це дозволяє перетворити текст на список чисел, які можна ввести в модель. Токенізація може здійснюватися кількома способами: або шляхом розбиття тексту на слова, або за допомогою більш складних підходів, таких як токенизація на рівні символів або частин слова.

- Набивка (padding)

Моделі глибокого навчання використовують вхідні дані фіксованого розміру. Нам потрібно заповнити текстові послідовності нулями або іншими значеннями, щоб переконатися, що вони мають однакову довжину, оскільки текстові документи можуть бути різної довжини. Це досягається шляхом встановлення максимальної довжини тексту, дозволеної в наборі даних, і рівномірного подовження коротших текстів шляхом додавання нулів (або інших значень) до довших текстів. Набивка є критично важливим кроком, оскільки LSTM-мережам потрібні послідовності однакової довжини.

Після виконання цих трьох кроків - видалення стоп-слів, токенизації тексту та набивки - дані є підготованими для навчання моделі LSTM для класифікації тексту. Трансформований текст може бути поданий на вхід моделі, яка виконає подальшу обробку даних та розпочне навчання за допомогою шарів LSTM для вирішення завдання класифікації тексту.

3.4 Навчання моделі

Для даної задачі було створено послідовну модель з чотирма прихованими рівнями:

1. Embedding - зіставляє кожне унікальне слово у словнику з вектором розміру 128;
2. Двонаправлена LSTM;
3. Dense з 128 елементами та функцією активації Rectified Linear Unit;
4. Dense - останній повністю зв'язний шар з однією вихідною одиницею та сигмоїдною функцією активації. На виході - двійкова класифікація (0 або 1), що представляє остаточний прогноз моделі.

У моделі використано оптимізатор adam. Опис моделі наведено на рис. 3.4

```

# Sequential Model
model = Sequential()

# Embedding layer
model.add(Embedding(total_words, output_dim = 128))

# Bi-Directional LSTM
model.add(Bidirectional(LSTM(128)))

# Dense layers
model.add(Dense(128, activation = 'relu'))
model.add(Dense(1, activation= 'sigmoid'))
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['acc'])
model.summary()

```

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, None, 128)	13914112
bidirectional (Bidirectional)	(None, 256)	263168
dense (Dense)	(None, 128)	32896
dense_1 (Dense)	(None, 1)	129

=====
Total params: 14,210,305
Trainable params: 14,210,305
Non-trainable params: 0

Рисунок 3.4 - Параметри моделі

3.5 Результати роботи моделі

3.5.1 Тренування моделі

Результати тренування моделі представлені у таблиці 3.1

Таблиця 3.1.

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
1/5	0.0407	0.9853	0.0126	0.9972
2/5	0.0025	0.9996	0.013	0.9967
3/5	0.0015	0.9997	0.0168	0.9981
4/5	0.00053095	0.9998	0.0174	0.9983
5/5	0.0014	0.9996	0.0138	0.9978

Розглянемо ці параметри детальніше.

Втрати та точність (Training Loss): Втрати при навчанні значно зменшуються від епохи до епохи, досягаючи дуже низького значення в кінці. Це свідчить про те, що модель ефективно навчається на навчальних даних. Точність навчання також постійно покращується, досягаючи майже ідеальної точності 0.9996-0.9998. Такі високі показники свідчать про те, що модель успішно навчилася класифікувати фейкові новини на навчальній вибірці.

Втрати при перевірці та точність: Втрати при перевірці спочатку зменшуються, але потім починають дещо зростати, починаючи з етапу 3. Точність валідації залишається стабільно високою (понад 0,9967), що свідчить про те, що модель все ще добре узагальнює нові, ще не бачені приклади.

Порівняння ефективності навчання та валідації: Існує невеликий розрив між значеннями втрат і точності при навчанні та валідації. Втрати та точність при навчанні трохи кращі, ніж при валідації, що очікувано. Однак різниця не є суттєвою, що свідчить про те, що модель не страждає від надмірного перенавчання.

3.5.2 Перевірка моделі на тестових даних

Перевірка моделі на 20% випадково вибраних даних показує точність у 99.610%. Показники навчання наведено у табл. 3.2.

Таблиця 3.2.

	precision	recall	f1-score	support
Клас 0	1.00	0.99	1.00	4236
Клас 1	1.00	0.99	1.00	4744
accuracy			1.00	8980
macro avg	1.00	1.00	1.00	8980
weighted avg	1.00	1.00	1.00	8980

Precision: Точність вимірює здатність моделі правильно класифікувати позитивні екземпляри (мітка 1) з усіх екземплярів, які вона прогнозує як

позитивні. Точність для класу 0 (нефейкові новини) становить 1,00, що означає, що всі випадки, передбачені як нефейкові новини, дійсно були нефейковими новинами. Точність для класу 1 (фейкові новини) також становить 0,99, що означає, що модель правильно класифікувала 99% випадків, які вона передбачила як фейкові новини.

Recall: Відгук (також відомий як чутливість або істинно позитивний показник) вимірює здатність моделі правильно ідентифікувати позитивні приклади (мітка 1) з усіх фактичних позитивних прикладів. Відгук для класу 0 становить 0,99, що означає, що модель правильно ідентифікувала 99% нефейкових новин. Відгук для класу 1 становить 1,00, що означає, що модель правильно ідентифікувала всі фейкові новини.

Оцінка F1: Показник F1 поєднує точність і відгук в єдину метрику, яка збалансовує обидва показники. Він забезпечує гармонійне середнє значення точності та пригадування. Показник F1 як для класу 0, так і для класу 1 становить 1,00, що свідчить про відмінну ефективність у правильній класифікації як нефейкових, так і фейкових новин.

Support: Під підтримкою мається на увазі кількість прикладів у кожному класі в даних тесту. До класу 0 (нефейкові новини) віднесено 4236 примірників, до класу 1 (фейкові новини) - 4744 примірники.

Accuracy: Загальна точність моделі на тестових даних становить 1,00, що означає, що модель правильно класифікувала 100% прикладів у тестовому наборі.

Macro Avg: Макросереднє обчислює середню продуктивність для обох класів, надаючи рівну вагу кожному класу. У цьому випадку середні макропоказники точності, згадуваності та F1-рахунку дорівнюють 1,00, що свідчить про відмінні показники як для класів нефейкових новин, так і для класів фейкових новин.

Weighted avg: Середньозважений показник обчислює середній показник для обох класів, беручи до уваги підтримку (кількість показів) для кожного класу. У цьому випадку середньозважений показник точності, відкриття та

F1-бали дорівнюють 1,00, що свідчить про відмінну загальну продуктивність на тестових даних.

Матриця невідповідностей моделі показана на рис. 4. Тут можна побачити, що ми отримали 32 хибно-позитивні та 3 хибно-негативні результати.

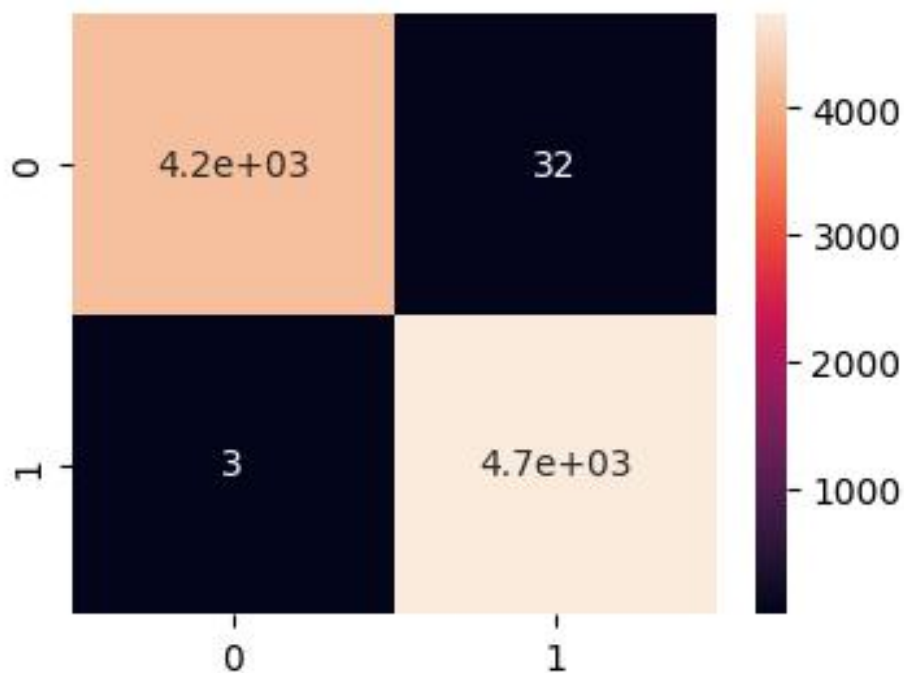


Рисунок 3.4 Матриця невідповідностей

Загалом модель демонструє чудові результати на тестових даних, досягаючи майже ідеальної точності, recall та F1-оцінки як для класів нефейкових новин, так і для класів фейкових новин. Висока точність свідчить про те, що модель добре узагальнює невидимі дані і здатна ефективно розрізняти справжні та фейкові новини.

3.5.3 Перевірка моделі на новині згенерованій Chat-GPT

Для новини, що було згенеровано Chat-GPT, модель видала показник 99,99% неправдивості. Текст новини наведено у додатку Б.

Деякі ключові слова заданої новини: stunning turn events bombshell revelation surfaced shattering core elections unveiling deceit corruption

unprecedented scandal threatens undermine integrity democratic process raise deep shadowy corridors power hidden public.

Імовірність що новина неправдива: [[0.99996036]]

Висновки до розділу 3

У цьому розділі було описано складові прикладної моделі, створеної для вирішення задачі класифікації текстових даних.

Модель досягає високої точності як на навчальній, так і на валідаційній вибірках.

Втрати при навчанні зменшуються з 0,0407 до 0,0014 протягом п'яти епох, що свідчить про те, що модель навчається і покращує свої прогнози.

Точність моделі залишається стабільно високою протягом усього процесу навчання, що свідчить про хорошу здатність до узагальнення.

Точність валідації залишається вищою за 0,9967, що свідчить про здатність моделі добре працювати на непередбачуваних даних.

РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

В заданому розділі буде проведено оцінювання основних характеристик для майбутнього програмного продукту, що спеціалізується на ідентифікації fake news за допомогою нейронних мереж.

Дана реалізація буде сприяти проведенню усіх необхідних досліджень, що дасть змогу якісно дослідити питання не лише в Україні, проте у всьому світі.

Також в даному дослідженні показано різні варіанти реалізації для забезпечення найбільш коректної та оптимальної стратегії вибору, що має вплив на економічні фактори та сумісність з майбутнім програмним продуктом. Для цього застосовувався апарат функціонально-вартісного аналізу.

Функціонально-вартісний аналіз (ФВА) передбачає собою технологію, що дозволяє оцінити реальну вартість продукту або послуги незалежно від організаційної структури компанії. ФВА проводиться з метою виявлення резервів зниження витрат за рахунок ефективніших варіантів виробництва, кращого співвідношення між споживчою вартістю виробу та витратами на його виготовлення. Для проведення аналізу використовується економічна, технічна та конструкторська інформація.

Алгоритм функціонально-вартісного аналізу включає в себе визначення послідовності етапів розробки продукту, визначення повних витрат (річних) та кількості робочих часів, визначення джерел витрат та кінцевий розрахунок вартості програмного продукту.

4.1 Постановка задачі проектування

У роботі застосовується метод ФВА для проведення техніко-економічного аналізу розробки системи ідентифікації fake news за допомогою нейронних мереж. Оскільки рішення стосовно проектування та реалізації

компонентів, що розробляється, впливають на всю систему, кожна окрема підсистема має її задовольняти. Тому фактичний аналіз представляє собою аналіз функцій програмного продукту, призначеного для збору, обробки та проведення аналізу даних по класифікації.

Технічні вимоги до програмного продукту є наступні:

- функціонування на персональних комп'ютерах із стандартним набором компонентів;
- зручність та зрозумілість для користувача;
- швидкість обробки даних та доступ до інформації в реальному часі;
- можливість зручного масштабування та обслуговування;
- мінімальні витрати на впровадження програмного продукту.

4.2 Обґрунтування функцій програмного продукту

Головна функція F_0 – розробка програмного продукту, який вирішує задачу розпізнавання та класифікації лейкоцитів на зображеннях крові. Беручи за основу цю функцію, можна виділити наступні:

F_1 – вибір мови програмування.

F_2 – вибір фреймворку машинного навчання.

F_3 – вибір середовища програмування.

Кожна з цих функцій має декілька варіантів реалізації:

Функція F_1 :

а) Python.

б) C++.

Функція F_2 .

а) TensorFlow.

б) Pytorch.

Функція F_3 :

а) Jupyter Notebook.

б) PyCharm.

Варіанти реалізації основних функцій наведені у морфологічній карті системи (рис. 4.1).

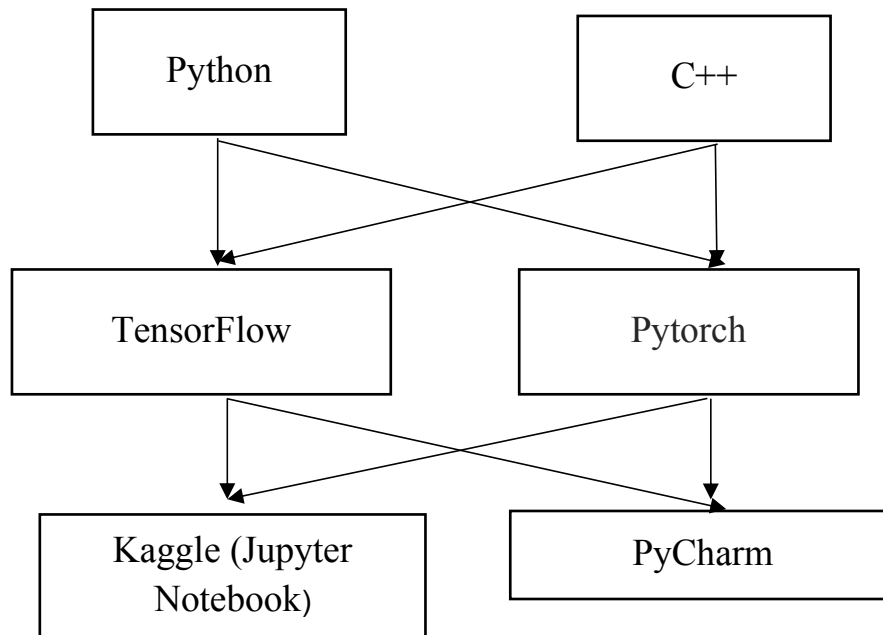


Рисунок 4.1 – Морфологічна карта

Морфологічна карта відображає множину всіх можливих варіантів основних функцій. Позитивно-негативна матриця показана в таблиці 4.1.

Таблиця 4.1.

Позитивно-негативна матриця

Функції	Варіанти реалізації	Переваги	Недоліки
F_1	<i>A</i>	Зручність, багатофункціональність, широкий вибір бібліотек	Швидкодія, обмежена підтримка апаратного прискорення
	<i>B</i>	Висока швидкодія, доступ до оптимізаційних можливостей процесора,	Складність, менша кількість бібліотек, вимога до вміння ручного керування пам'яттю
F_2	<i>A</i>	Широкий вибір	Складніший синтаксис,

		моделей, розширені можливості управління пам'яттю	більша кількість коду для досягнення тих самих результатів,.
	<i>Б</i>	Простий синтаксис, динамічні графи, легкість у розробці та налагодженні моделей,	Менша швидкодія порівняно з TensorFlow на деяких завданнях
<i>F₃</i>	<i>А</i>	Інтерактивність, можливість візуалізації даних та результатів, зручність	Потребує багато ресурсів, обмежена підтримка великих обсягів даних,
	<i>Б</i>	Підтримка автодоповнення, інструменти для оптимізації, робота з великими проектами	Висока ціна платної версії, проблеми з налаштуванням деяких бібліотек

На основі аналізу позитивно-негативної матриці робимо висновок, що при розробці програмного продукту деякі варіанти реалізації функцій варто відкинути, тому, що вони не відповідають поставленим перед програмним продуктом задачам. Ці варіанти відзначені у морфологічній карті.

Функція F_1 :

Перевагу даємо зручності, багатофункціональності, широкому вибору бібліотек. Для спрощення роботи по написанню коду варіант Б має бути відкинтий.

Функція F_2 :

Програма допускає обрання обох варіантів. Можливо використати варіанти А чи Б.

Функція F_3 :

Реалізація першого варіанту є сприйнятливою для програми. Це варіант А. Таким чином, будемо розглядати такі варіанти реалізації ПП:

$$F_{1a} - F_{2a} - F_{3a}$$

$$F_{1a} - F_{2b} - F_{3a}$$

Для оцінювання якості розглянутих функцій обрана система параметрів, описана нижче.

4.3 Обґрунтування системи параметрів програмного продукту

На основі даних, розглянутих вище, визначаються основні параметри вибору, які будуть використані для розрахунку коефіцієнта технічного рівня.

Для того, щоб охарактеризувати програмний продукт, будемо використовувати наступні параметри:

- $X1$ – швидкодія мови програмування;
- $X2$ – об'єм пам'яті для обчислень та збереження даних;
- $X3$ – час навчання даних;
- $X4$ – потенційний об'єм програмного коду.

Гірші, середні і кращі значення параметрів вибираються на основі вимог замовника й умов, що характеризують експлуатацію програмного продукту, як показано у таблиці 4.2.

Таблиця 4.2 - Основні параметри програмного продукту

Назва Параметра	Умовні позначення	Одиниці виміру	Значення параметра		
			гірші	середні	кращі
Швидкодія мови програмування	$X1$	оп/мс	9000	14000	19000
Об'єм пам'яті	$X2$	Мб	256	128	64

Час попередньої обробки даних	X3	мс	5	4	3
Потенційний об'єм програмного коду	X4	кількість рядків коду	3000	1500	500

За даними таблиці 4.3 будуються графічні характеристики параметрів – рис. 4.2 – рис. 4.5.

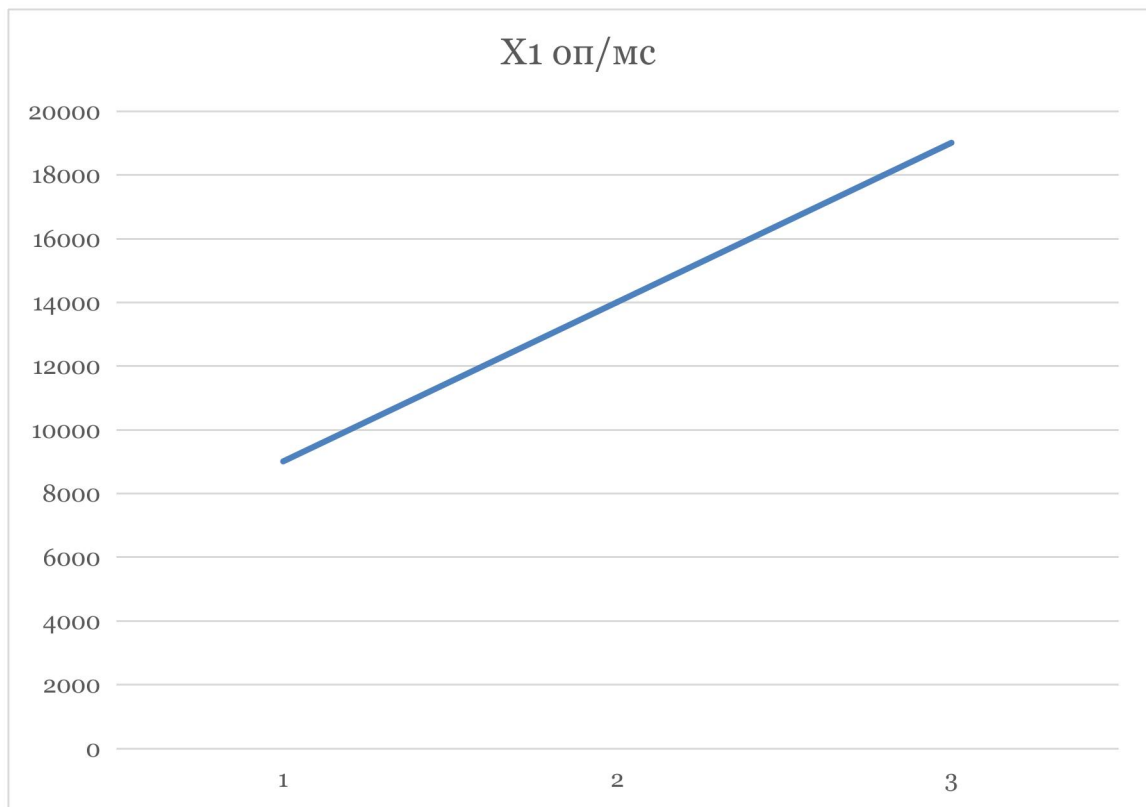


Рисунок 4.2 – X1, швидкодія мови програмування

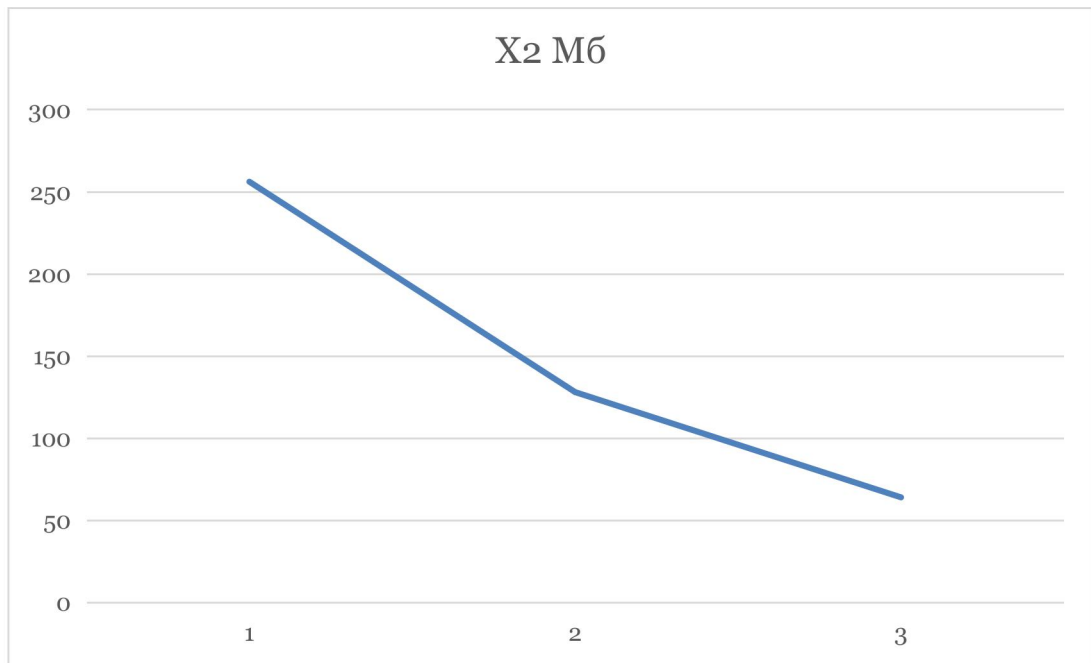


Рисунок 4.3 – X2, об'єм пам'яті

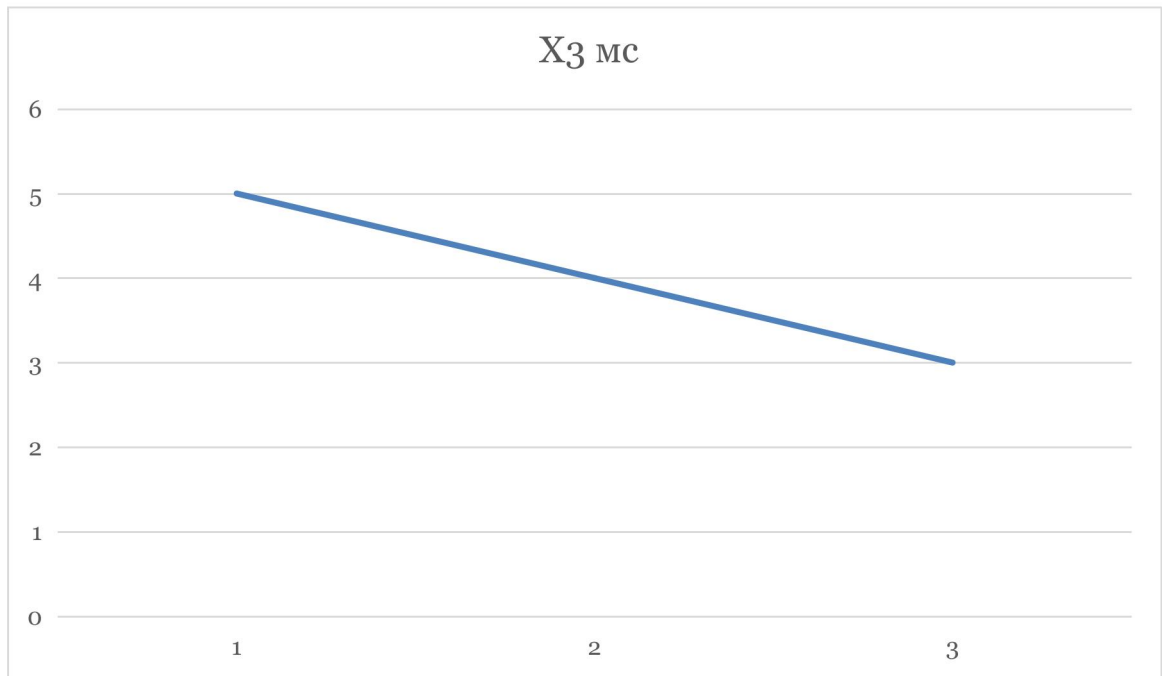


Рисунок 4.4 – X3, час попередньої обробки даних

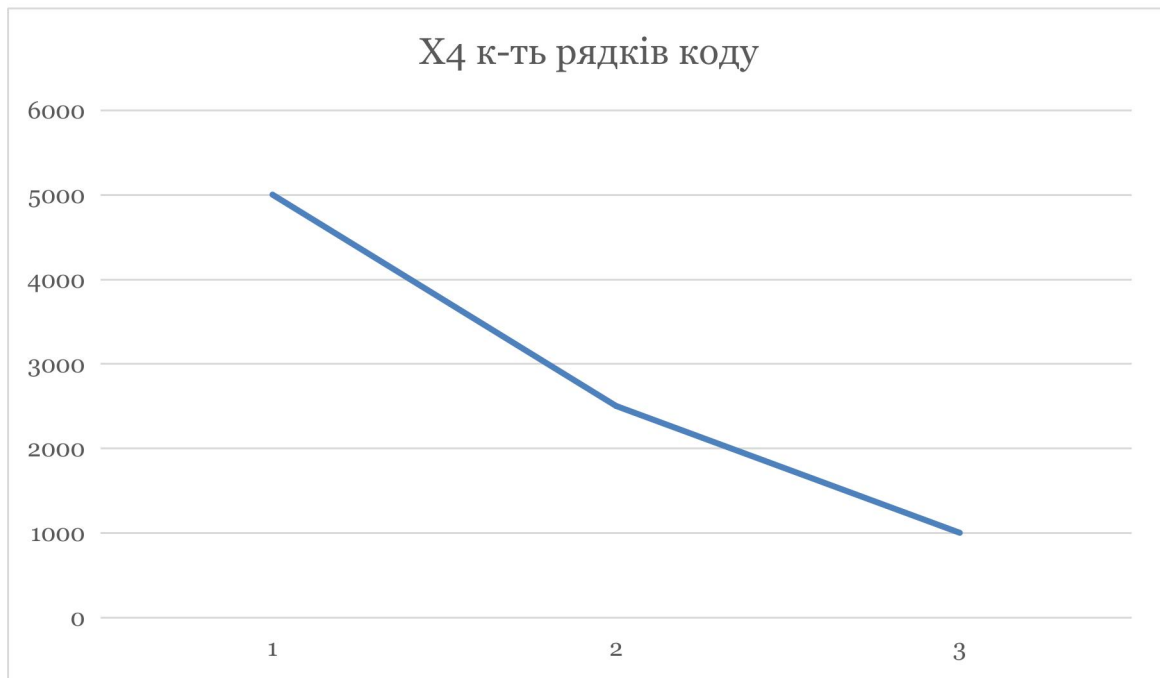


Рисунок 4.5 – X4, потенційний об'єм програмного коду

4.4 Аналіз експертного оцінювання параметрів

Після детального обговорення й аналізу кожний експерт оцінює ступінь важливості кожного параметру для конкретно поставленої цілі – розробка програмного продукту, який дає найбільш точні результати при знаходженні параметрів моделей адаптивного прогнозування і обчислення прогнозних значень.

Значимість кожного параметра визначається методом попарного порівняння. Оцінку проводить експертна комісія із 7 людей. Визначення коефіцієнтів значимості передбачає:

- визначення рівня значимості параметра шляхом присвоєння різних рангів;
- перевірку придатності експертних оцінок для подальшого використання;
- визначення оцінки попарного пріоритету параметрів;
- обробку результатів та визначення коефіцієнту значимості.

Результати експертного ранжування наведені у таблиці 4.3.

Таблиця 4.3 - Результати ранжування параметрів

Позначення параметра	Назва параметра	Одиниці виміру	Ранг параметра за оцінкою експерта							Сума рангів R_i	Відхилення Δ_i	Δ_i^2
			1	2	3	4	5	6	7			
$X1$	Швидкодія мови програмування	Оп/мс	1	2	3	1	1	5	3	16	-6.75	45.56
$X2$	Об'єм пам'яті	Мб	6	5	3	2	5	4	6	31	8.25	68.06
$X3$	Час попередньої обробки даних	мс	1	2	2	4	1	3	2	15	-7.75	60.06
$X4$	Потенційний об'єм програмного коду	Кількість рядків коду	5	4	5	6	6	1	2	29	6.25	39.06
	Разом		13	13	13	13	13	13	13	91	0	212.75

Для перевірки степені достовірності експертних оцінок, визначимо наступні параметри:

а) сума рангів кожного з параметрів і загальна сума рангів:

$$R_i = \sum_{j=1}^N r_{ij} R_{ij} = \frac{Nn(n+1)}{2} = 91, \#(4.1)$$

де N – число експертів,

n – кількість параметрів;

б) середня сума рангів:

$$T = \frac{1}{n} R_{ij} = 22.75 \quad \#(4.2)$$

в) відхилення суми рангів кожного параметра від середньої суми рангів:

$$\Delta_i = R_i - T. \quad \#(4.3)$$

Сума відхилень по всіх параметрах повинна дорівнювати 0;

г) загальна сума квадратів відхилення:

$$S = \sum_{i=1}^N \Delta_i^2 = 212.75 \quad \#(4.4)$$

Порахуємо коефіцієнт узгодженості:

$$W = \frac{12S}{N^2(n^3 - n)} = \frac{12 \cdot 182}{7^2(4^3 - 4)} = 0,868 > W_k = 0,67. \quad \#(4.5)$$

Ранжування можна вважати достовірним, тому що знайдений коефіцієнт узгодженості перевищує нормативний, котрий дорівнює 0,67.

Скориставшись результатами ранжирування, проведемо попарне порівняння всіх параметрів і результати занесемо у таблицю 4.4.

Таблиця 4.4 - Попарне порівняння параметрів.

Параметри	Експерти							Кінцева оцінка	Числове значення
	1	2	3	4	5	6	7		
X1 і X2	<	<	=	<	<	>	<	<	0,5
X1 і X3	=	=	>	<	=	<	<	=	1,0

X1 i X4	<	<	<	<	<	>	>	<	0,5
X2 i X3	>	>	>	<	>	>	>	>	1,5
X2 i X4	>	>	<	<	<	>	>	>	1,5
X3 i X4	<	<	<	<	<	>	=	<	0,5

Числове значення, що визначає ступінь переваги i -го параметра над j -тим, a_{ij} визначається по формулі:

$$a_{ij} = \begin{cases} 1.5 \text{ при } X_i > X_j \\ 1.0 \text{ при } X_i = X_j \\ 0.5 \text{ при } X_i < X_j \end{cases} \#(4.6)$$

З отриманих числових оцінок переваги складемо матрицю $A = \|a_{ij}\|$.

Для кожного параметра зробимо розрахунок вагомості K_{vi} за наступними формулами:

$$K_{vi} = \frac{b_i}{\sum_{i=1}^n b_i} \#(4.7)$$

$$b_i = \sum_{i=1}^N a_{ij} \#(4.8)$$

Відносні оцінки розраховуються декілька разів доти, поки наступні значення не будуть незначно відрізнятись від попередніх (менше 2%). На другому і наступних кроках відносні оцінки розраховуються за наступними формулами:

$$K_{vi} = \frac{b'_i}{\sum_{i=1}^n b'_i}, \#(4.9)$$

$$b_i' = \sum_{j=1}^N a_{ij} b_j \quad \#(4.10)$$

Як видно з таблиці 4.5, різниця значень коефіцієнтів вагомості не перевищує 2%, тому більшої кількості ітерацій не потрібно.

Таблиця 4.5 - Розрахунок вагомості параметрів

Параметри x_i	Параметри x_j				Перша ітер.		Друга ітер.		Третя ітер	
	X1	X2	X3	X4	b_i	K_{bi}	b_i^1	K_{bi}^1	b_i^2	K_{bi}^2
X1	1.0	0.5	1.0	0.5	3.0	0.19	9.00	0.14	27	0.190
X2	1.5	1.0	1.5	1.5	5.5	0.35	30.25	0.46	166.37	0.254
X3	0.5	0.5	1.0	0.5	2.5	0.16	6.25	0.10	15.625	0.252
X4	1.5	0.5	1.5	1.0	4.5	0.29	20.25	0.31	91.125	0.304
Всього:					15.5	1	65.75	1	300.12	1

4.5 Аналіз рівня якості варіантів реалізації функцій

Визначаємо рівень якості кожного варіанту виконання основних функцій окремо.

Абсолютні значення параметрів X2 (Об'єм пам'яті), X3 (час попередньої обробки даних) та X4 (потенційний об'єм програмного коду) відповідають технічним вимогам умов функціонування даного ПП.

Абсолютне значення параметра X1 (швидкість роботи мови програмування) обрано не найгіршим.

Коефіцієнт технічного рівня для кожного варіанта реалізації ПП розраховується так (таблиця 4.6):

$$K_K(j) = \sum_{i=1}^n K_{bi,j} B_{i,j}, \quad \#(4.11)$$

де n – кількість параметрів;

K_{vi} – коефіцієнт вагомості i -го параметра;

B_i – оцінка i -го параметра в балах.

Таблиця 4.6 - Розрахунок показників рівня якості варіантів реалізації основних функцій ПП

Основні функції	Варіант реалізації функції	Параметри	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
F1	А	X1	16000	19	0.190	3.61
F3	А	X2	64	16	0.254	4.064
	Б	X3	128	14	0.252	3.528
F4	А	X4	800	12	0.304	3.648

За даними з таблиці 4.6 за формулою:

$$K_K = K_{ty}[F_{1k}] + K_{ty}[F_{2k}] + \dots + K_{ty}[F_{zk}], \#(4.12)$$

визначаємо рівень якості кожного з варіантів:

$$K_{K1} = 3.61 + 4.06 + 3.65 = 11.32 ;$$

$$K_{K2} = 3 + 3.53 + 3.55 = 10.08 .$$

Як видно з розрахунків, кращим є 1 варіант, для якого коефіцієнт технічного рівня має найбільше значення.

4.6 Економічний аналіз варіантів розробки ПП

Для визначення вартості розробки ПП спочатку проведемо розрахунок трудомісткості.

Всі варіанти включають в себе два окремих завдання:

1. Розробка проекту програмного продукту;
2. Розробка програмної оболонки;

Завдання 1 за ступенем новизни відноситься до групи А, завдання 2 – до групи Б. За складністю алгоритми, які використовуються в завданні 1 належать до групи 1; а в завданні 2 – до групи 3.

Для реалізації завдання 1 використовується довідкова інформація, а завдання 2 використовує інформацію у вигляді даних.

Проведемо розрахунок норм часу на розробку та програмування для кожного з завдань.

Загальна трудомісткість обчислюється як:

$$T_o = T_p \cdot K_p \cdot K_{СК} \cdot K_M \cdot K_{СТ} \cdot K_{СТ.М}, \#(4.13)$$

де T_p – трудомісткість розробки ПП;

K_p – поправочний коефіцієнт;

$K_{СК}$ – коефіцієнт на складність вхідної інформації;

K_M – коефіцієнт рівня мови програмування;

$K_{СТ}$ – коефіцієнт використання стандартних модулів і прикладних програм;

$K_{СТ.М}$ – коефіцієнт стандартного математичного забезпечення

Для першого завдання, виходячи із норм часу для завдань розрахункового характеру ступеню новизни А та групи складності алгоритму 1, трудомісткість дорівнює: $T_p = 25$ людино-днів. Поправочний коефіцієнт, який враховує вид нормативно-довідкової інформації для першого завдання: $K_p = 1.5$. Поправочний коефіцієнт, який враховує складність контролю вхідної та вихідної інформації для всіх семи завдань рівний 1: $K_{СК} = 1$. Оскільки при розробці першого завдання використовуються стандартні модулі, врахуємо це

за допомогою коефіцієнта $K_{CT} = 0.7$. Тоді загальна трудомісткість програмування першого завдання дорівнює:

$$T_1 = 25 \cdot 1.5 \cdot 0.7 = 26.25 \text{ людино-днів.}$$

Проведемо аналогічні розрахунки для подальших завдань.

Для другого завдання (використовується алгоритм третьої групи складності, степінь новизни Б), тобто $T_P = 20$ людино-днів, $K_{II} = 1.3$, $K_{CK} = 1$, $K_{CT} = 0.9$:

$$T_2 = 20 \cdot 1.3 \cdot 0.9 = 23.4 \text{ людино-днів.}$$

Складаємо трудомісткість відповідних завдань для кожного з обраних варіантів реалізації програми, щоб отримати їх трудомісткість:

$$T_I = (39,68 + 27 + 4.6 + 27) \cdot 8 = 786,2 \text{ людино-годин.}$$

$$T_{II} = (39,68 + 27 + 8.22 + 27) \cdot 8 = 815,2 \text{ людино-годин.}$$

Найбільш високу трудомісткість має варіант II.

В розробці беруть участь два програмісти з окладом 25600 грн., один аналітик в області даних з окладом 36570. Визначимо середню зарплату за годину за формулою:

$$C_{ч} = \frac{M}{T_m \cdot t} \text{ грн., \#(4.14)}$$

де M – місячний оклад працівників;

T_m – кількість робочих днів тиждень;

t – кількість робочих годин в день.

$$C_{\text{ч}} = \frac{29526 + 29526 + 36908}{3 \cdot 21 \cdot 8} = 174.15 \text{ грн.} \#(4.15)$$

Тоді, розрахуємо заробітну плату за формулою:

$$C_{\text{зп}} = C_{\text{ч}} \cdot T_i \cdot K_{\text{д}}, \#(4.16)$$

де $C_{\text{ч}}$ – величина погодинної оплати праці програміста;

T_i – трудомісткість відповідного завдання;

$K_{\text{д}}$ – норматив, який враховує додаткову заробітну плату.

Зарплата розробників за варіантами становить:

$$\text{I. } C_{\text{зп}} = 174.15 \cdot 786,2 \cdot 1.2 = 164300,10 \text{ грн.}$$

$$\text{II. } C_{\text{зп}} = 174.15 \cdot 815,2 \cdot 1.2 = 170360,49 \text{ грн.}$$

Відрахування на єдиний соціальний внесок становить 22%:

$$\text{I. } C_{\text{вд}} = C_{\text{зп}} \cdot 0.22 = 181092,87 \cdot 0.22 = 36146,02 \text{ грн.}$$

$$\text{II. } C_{\text{вд}} = C_{\text{зп}} \cdot 0.22 = 184419,36 \cdot 0.22 = 37479,31 \text{ грн.}$$

Тепер визначимо витрати на оплату однієї машино-години. ($C_{\text{м}}$)

Так як одна ЕОМ обслуговує одного програміста з окладом 25600 грн., з коефіцієнтом зайнятості 0,2 то для однієї машини отримаємо:

$$C_{\text{г}} = 12 \cdot M \cdot K_3 = 12 \cdot 25600 \cdot 0,2 = 61440 \text{ грн.}$$

З урахуванням додаткової заробітної плати:

$$C_{3П} = C_{Г} \cdot (1 + K_3) = 70862,4 \cdot (1 + 0.2) = 73728 \text{ грн.}$$

Відрахування на соціальний внесок:

$$C_{ВІД} = C_{3П} \cdot 0.22 = 85034,88 \cdot 0,22 = 16220.16 \text{ грн.}$$

Амортизаційні відрахування розраховуємо при амортизації 25% та вартості ЕОМ – 27000 грн.

$$C_A = K_{ТМ} \cdot K_A \cdot Ц_{ПР} = 1.2 \cdot 0.25 \cdot 27000 = 8100 \text{ грн.,}$$

де $K_{ТМ}$ – коефіцієнт, який враховує витрати на транспортування та монтаж приладу у користувача;

K_A – річна норма амортизації;

$Ц_{ПР}$ – договірна ціна приладу.

Витрати на ремонт та профілактику розраховуємо як:

$$C_P = K_{ТМ} \cdot Ц_{ПР} \cdot K_P = 1.2 \cdot 27000 \cdot 0.05 = 1620 \text{ грн.,}$$

де K_P – відсоток витрат на поточні ремонти.

Ефективний годинний фонд часу ПК за рік розраховуємо за формулою:

$$\begin{aligned} T_{ЕФ} &= (D_K - D_B - D_C - D_P) \cdot t_3 \cdot K_B = (365 - 104 - 8 - 11) \cdot 8 \cdot 0.7 = \\ &= 1355,2 \text{ години,} \end{aligned}$$

де D_K – календарна кількість днів у році;

D_B, D_C – відповідно кількість вихідних та святкових днів;

D_p – кількість днів планових ремонтів устаткування;

t – кількість робочих годин в день;

K_B – коефіцієнт використання приладу у часі протягом зміни.

Витрати на оплату електроенергії розраховуємо за формулою:

$$C_{\text{ЕЛ}} = T_{\text{ЕФ}} \cdot N_{\text{С}} \cdot K_3 \cdot C_{\text{ЕН}} = 1355,2 \cdot 0,4 \cdot 0,23 \cdot 4,86 = 605,93 \text{ грн.},$$

де $N_{\text{С}}$ – середньо-споживча потужність приладу;

K_3 – коефіцієнтом зайнятості приладу;

$C_{\text{ЕН}}$ – тариф за 1 КВт-годин електроенергії.

Накладні витрати розраховуємо за формулою:

$$C_{\text{Н}} = C_{\text{ПР}} \cdot 0,67 = 27000 \cdot 0,67 = 18090 \text{ грн.}$$

Тоді, річні експлуатаційні витрати будуть:

$$C_{\text{ЕКС}} = C_{\text{ЗП}} + C_{\text{ВІД}} + C_{\text{А}} + C_{\text{Р}} + C_{\text{ЕЛ}} + C_{\text{Н}}, \#(4.17)$$

$$C_{\text{ЕКС}} = 85034,88 + 18707,67 + 8100 + 1620 + 658,62 + 18090 = 118364,09 \text{ грн.}$$

Собівартість однієї машино-години ЕОМ дорівнюватиме:

$$C_{\text{М-Г}} = C_{\text{ЕКС}} / T_{\text{ЕФ}} = 118364,09 / 1355,2 = 87,34 \text{ грн/год.}$$

Оскільки в даному випадку всі роботи, які пов'язані з розробкою програмного продукту ведуться на ЕОМ, витрати на оплату машинного часу, в залежності від обраного варіанта реалізації, складає:

$$C_M = C_{M-Г} \cdot T, \#(4.18)$$

$$I. C_M = 97,55 \cdot 792,64 = 77322,03 \text{ грн.}$$

$$II. C_M = 97,55 \cdot 807,2 = 78742,36 \text{ грн.}$$

Накладні витрати складають 67% від заробітної плати:

$$C_H = C_{зп} \cdot 0,67, \#(4.19)$$

$$I. C_H = 181092,87 \cdot 0,67 = 121332,2 \text{ грн.}$$

$$II. C_H = 184419,36 \cdot 0,67 = 123560,97 \text{ грн.}$$

Отже, вартість розробки ПП за варіантами становить:

$$C_{пп} = C_{зп} + C_{в\text{і}д} + C_M + C_H, \#(4.20)$$

$$I. C_{пп} = 181092,87 + 39840,43 + 77322,03 + 121332,2 = 419587,53 \text{ грн.}$$

$$II. C_{пп} = 184419,36 + 40572,25 + 78742,36 + 123560,97 = 427294,94 \text{ грн.}$$

4.7 Вибір кращого варіанту ПП техніко-економічного рівня

Розрахуємо коефіцієнт техніко-економічного рівня за формулою:

$$K_{\text{TEP}j} = K_{kj} / C_{\Phi j}, \# (4.21)$$

$$K_{\text{TEP}1} = 13,64 / 419587,53 = 3,2508 \cdot 10^{-5},$$

$$K_{\text{TEP}2} = 11,28 / 427294,94 = 2,6398 \cdot 10^{-5}.$$

Як бачимо, найбільш ефективним є перший варіант реалізації програми з коефіцієнтом техніко-економічного рівня $K_{\text{TEP}1} = 3,2508 \cdot 10^{-5}$.

Після виконання функціонально-вартісного аналізу програмного комплексу що розроблюється, можна зробити висновок, що з альтернатив, що залишились після першого відбору двох варіантів виконання програмного комплексу оптимальним є перший варіант реалізації програмного продукту. У нього виявився найкращий показник техніко-економічного рівня якості

$$K_{\text{TEP}} = 3,2508 \cdot 10^{-5}.$$

Цей варіант реалізації програмного продукту має такі параметри:

Мова програмування – Python;

Широкий вибір моделей, розширені можливості управління пам'яттю

Інтерактивність, можливість візуалізації даних та результатів, зручність

Даний варіант виконання програмного комплексу дає користувачу зручний інтерфейс, швидку та зручну реалізацію програми, доступний функціонал для роботи.

4.8 Висновки до четвертого розділу

У даній частині було проведено повний функціонально-вартісний аналіз програмного продукту. Також було знайдено оцінку основних функцій програмного продукту.

У результаті виконання функціонально-вартісного аналізу програмного комплексу що розроблюється, було визначено та проведено оцінку основних функцій програмного продукту, а також знайдено параметри, які його характеризують.

На основі аналізу вибрано варіант реалізації програмного продукту.

ВИСНОВКИ

У цій роботі проведено дослідження класифікації fake news з використанням рекурентних нейронних мереж з довгою короткочасною пам'яттю.

Перший розділ присвячено темам актуальності задачі та сучасного вигляду предметної області. Також, надано формулювання задачі класифікації текстових новин. Далі окреслено постановку задачі, та описано вимоги до системи.

У другому розділі увага акцентується на математичній основі роботи. Введено основні поняття рекурентних нейронних мереж з довгою короткочасною пам'яттю. Наведено приклади базових рівнів мереж рекурентних нейронних мереж з довгою короткочасною пам'яттю, описано рівні нейронної мережі.

Розділ 3 присвячений практичній частині роботи. Системні вимоги розглядаються детально, виходячи з мови програмування та фреймворку, обраного для кожного компонента системи. Описуються методи попередньої обробки вхідних даних для підвищення продуктивності мережі. Розглянуто основні метрики, що використовуються для задач класифікації тексту. Показано та описано існуючі архітектури нейронних мереж. Створено свою власну архітектуру. Нарешті описано та продемонстровано роботу створеної системи та проаналізовано результати роботи. Також, перевірено роботу моделі на даних не представлених у початкових датасетах та проаналізовано результат.

У четвертому розділі проводиться комплексний функціональний аналіз і аналіз вартості програмного продукту, а також оцінюються основні функції програмного продукту. У результаті виділяються та оцінюються основні функції програмного продукту та знаходяться параметри, що його характеризують. Вибрано варіант реалізації програмного продукту.

Список використаних джерел

1. Ukraine under information fire - euvdisinfo. *EUvsDisinfo*. [Електронний ресурс]. Режим доступу: <https://euvdisinfo.eu/ukraine-under-information-fire>
2. Кожен третій українець зізнається, що не може виявити фейк в мас-медіа, - дослідження [Електронний ресурс]. Режим доступу: <https://web.archive.org/web/20190103004806/https://intvua.com/news/society/1522160173-kozhen-tretyi-ukrayinets-ziznaetsya-shcho-ne-mozhe-viyaviti-feyk.html>
3. Фейків багато, теорій фейків іще більше. *StopFake*. [Електронний ресурс]. Режим доступу: <https://www.stopfake.org/uk/fejkiv-bagato-teorij-fejkiv-ishhe-bilshe/>.
4. Esma Aımeur, Fake news, disinformation and misinformation in social media: a review. University of Montreal, Montreal, Canada [Електронний ресурс]. Режим доступу: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9910783/>
5. Raza, S., Ding, C. Fake news detection based on news content and social contexts: a transformer-based approach. *Int J Data Sci Anal* **13**, 335–362 (2022). [Електронний ресурс]. Режим доступу: <https://doi.org/10.1007/s41060-021-00302-z>
6. Pritika Bahad, Preeti Saxena, Raj Kamal. Fake News Detection using Bi-directional LSTM-Recurrent Neural Network. *Procedia Computer Science*, Volume 165, 74-82 (2019). [Електронний ресурс]. Режим доступу: <https://doi.org/10.1016/j.procs.2020.01.072> .
7. K. Greff, R. Srivastava, J. Koutnik, B. Steunebrink, J. Schmidhuber. LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 2016. [Електронний ресурс]. Режим доступу: <https://doi.org/10.1109/TNNLS.2016.2582924> .
8. Long Short-Term Memory in Recurrent Neural Networks. [Електронний ресурс]. Режим доступу: <http://www.felixgers.de/papers/phd.pdf>.

9. Olusola Adeniyi Abidogun. Data Mining, Fraud Detection and Mobile Telecommunications: Call Pattern Analysis with Unsupervised Neural Networks (2005). [Электронный ресурс]. Режим доступа: https://web.archive.org/web/20120522234026/http://etd.uwc.ac.za/usrfiles/modules/etd/docs/etd_init_3937_1174040706.pdf
10. A generalized LSTM-like training algorithm for second-order recurrent neural networks. [Электронный ресурс]. Режим доступа: <http://www.overcomplete.net/papers/nn2012.pdf>
11. Graves, A. (2013). Generating Sequences With Recurrent Neural Networks. *ArXiv*. /abs/1308.0850
12. Structure of the LSTM cell. [Электронный ресурс]. Режим доступа: https://www.researchgate.net/figure/Structure-of-the-LSTM-cell-and-equations-that-describe-the-gates-of-an-LSTM-cell_fig5_329362532
13. Long Short-Term Memory Networks (LSTM).[Электронный ресурс]. Режим доступа: <https://databasecamp.de/en/ml/lstms>
14. What Are Embedding Layers in Neural Networks? [Электронный ресурс]. Режим доступа: <https://www.baeldung.com/cs/neural-nets-embedding-layers>
15. What does it mean by Bidirectional LSTM? [Электронный ресурс]. Режим доступа: <https://medium.com/analytics-vidhya/what-does-it-mean-by-bidirectional-lstm-63d6838e34d9>
16. A Complete Understanding of Dense Layers in Neural Networks [Электронный ресурс]. Режим доступа: <https://analyticsindiamag.com/a-complete-understanding-of-dense-layers-in-neural-networks/>
17. Fake News Detection Datasets [Электронный ресурс]. Режим доступа: <https://onlineacademiccommunity.uvic.ca/isot/2022/11/27/fake-news-detection-datasets/>

ДОДАТОК А ПРОГРАМНИЙ КОД

```
# !pip install --upgrade tensorflow
# !pip install plotly
# !pip install --upgrade numpy
# !pip install --upgrade nbformat
# !pip install nltk
# !pip install spacy
# !pip install WordCloud
# !pip install gensim # !pip install seaborn
```

```
import nltk
```

```
import nltk
```

```
import ssl
```

```
try:
```

```
    _create_unverified_https_context =
ssl._create_unverified_context
```

```
except AttributeError:
```

```
    pass
```

```
else:
```

```
    ssl._create_default_https_context =
_create_unverified_https_context
```

```
# nltk.download()
```

```
nltk.download('punkt')
```

```
import tensorflow as tf
```

```
import pandas as pd
```

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud, STOPWORDS
import nltk
import re
from nltk.stem import PorterStemmer, WordNetLemmatizer
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize
import gensim
from gensim.utils import simple_preprocess
from gensim.parsing.preprocessing import STOPWORDS
from tensorflow.keras.preprocessing.text import one_hot,
Tokenizer
from tensorflow.keras.preprocessing.sequence import
pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten, Embedding,
Input, LSTM, Conv1D, MaxPool1D, Bidirectional
from tensorflow.keras.models import Model

df_true = pd.read_csv("sample_data/True.csv")
df_fake = pd.read_csv("sample_data/Fake.csv")

df_true.head()

df_fake.head()

df_true['isfake'] = 0
```

```
df_true.head()

df_fake['isfake'] = 1
df_fake.head()

df = pd.concat([df_true, df_fake]).reset_index(drop = True)
df
df['isfake'].value_counts()
df.drop(columns = ['date'], inplace = True)

df['original'] = df['title'] + ' ' + df['text']
df.head()

nltk.download("stopwords")

from nltk.corpus import stopwords
stop_words = stopwords.words('english')
stop_words.extend(['from', 'subject', 're', 'edu', 'use'])

# Removing Stopwords And Remove Words With 2 Or Less
# Characters
def preprocess(text):
    result = []
    for token in gensim.utils.simple_preprocess(text):
        if token not in gensim.parsing.preprocessing.STOPWORDS
        and len(token) > 3 and token not in stop_words:
            result.append(token)
```

```
    return result

df['clean'] = df['original'].apply(preprocess)

df['original'][0]

print(df['clean'][0])

df

# Obtaining The Total Words Present In The Dataset
list_of_words = []
for i in df.clean:
    for j in i:
        list_of_words.append(j)

list_of_words

len(list_of_words)

total_words = len(list(set(list_of_words)))
total_words

df['clean_joined'] = df['clean'].apply(lambda x: " ".join(x))

df['clean_joined'][0]
```

```
plt.figure(figsize = (8, 8))
plt.title("Кількість новин за темою ")
sns.countplot(y = "subject", data = df)
plt.show()

plt.figure(figsize = (8, 8))
plt.title("Кількість правдивих та неправдивих новин")
sns.countplot(y = "isfake", data = df)
plt.show()

# Plotting The Word Cloud For Text That Is Real
plt.figure(figsize = (20,20))
wc = WordCloud(max_words = 2000 , width = 1600 , height =
800 , stopwords = stop_words).generate(" ".join(df[df.isfake
== 1].clean_joined))
plt.imshow(wc, interpolation = 'bilinear')
plt.show()

# Determining The Maximum Number Of Words In Any Document
Required To Create Word Embeddings
maxlen = -1
for doc in df.clean_joined:
    tokens = nltk.word_tokenize(doc)
    if(maxlen<len(tokens)):
        maxlen = len(tokens)
print("The maximum number of words in any document is =",
maxlen)

# Visualizing The Distribution Of Number Of Words In A Text
```

```
import plotly.express as px

fig = px.histogram(x = [len(nltk.word_tokenize(x)) for x in
df.clean_joined], nbins = 100)

fig.show()

# Splitting Data Into Test And Train

from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test =
train_test_split(df.clean_joined, df.isfake, test_size = 0.2)

from nltk import word_tokenize

x_train

# Creating A Tokenizer To Tokenize The Words And Create
Sequences Of Numbers

tokenizer = Tokenizer(num_words = total_words)

tokenizer.fit_on_texts(x_train)

train_sequences = tokenizer.texts_to_sequences(x_train)

test_sequences = tokenizer.texts_to_sequences(x_test)

print("The encoding for document\n",df.clean_joined[0],"\n
is : \n\n",train_sequences[0])

# Adding Padding

padded_train = pad_sequences(train_sequences,maxlen = 40,
padding = 'post', truncating = 'post')
```



```
padded_test = pad_sequences(test_sequences,maxlen = 40,
truncating = 'post')

for i,doc in enumerate(padded_train[:2]):
    print("The padded encoding for document",i+1," is : ",doc)

# Sequential Model
model = Sequential()

# Embedding layer
model.add(Embedding(total_words, output_dim = 128))

# Bi-Directional LSTM
model.add(Bidirectional(LSTM(128)))

# Dense layers
model.add(Dense(128, activation = 'relu'))
model.add(Dense(1,activation= 'sigmoid'))

model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['acc'])

model.summary()

y_train = np.asarray(y_train)

model.fit(padded_train, y_train, batch_size = 64,
validation_split = 0.1, epochs = 5)
```

```

# Making prediction
pred = model.predict(padded_test)

padded_test

# If The Predicted Value Is >0.95 (i.e., More Than 95%), It
Is Real Else It Is Fake

prediction = []

for i in range(len(pred)):
    if pred[i].item() > 0.95:
        prediction.append(1)
    else:
        prediction.append(0)

from sklearn.metrics import accuracy_score

accuracy = accuracy_score(list(y_test), prediction)

print("Model Accuracy : ", accuracy)

# Getting The Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(list(y_test), prediction)
plt.figure(figsize = (4, 3))
sns.heatmap(cm, annot = True)
plt.show()

# Getting The classification_report
```

```
from sklearn.metrics import classification_report
print(classification_report(list(y_test), prediction))

df_chat_gpt=pd.read_csv("sample_data/Fake-Chat-GPT.csv",
delimiter=";")

df_chat_gpt

def predict(df_chat_gpt):
    df_chat_gpt['clean'] = df_chat_gpt['text'].apply(preprocess)

    df_chat_gpt['clean_joined'] =
df_chat_gpt['clean'].apply(lambda x: " ".join(x))

    print("Ключові слова заданої новини: ",
df_chat_gpt['clean_joined'][0])

    list_of_words = []

    for i in df_chat_gpt['clean'][0]:

        list_of_words.append(i)

total_words = len(list(set(list_of_words)))

tokenizer = Tokenizer(num_words = total_words)
tokenizer.fit_on_texts(df_chat_gpt['clean_joined'])
```

```
tokenized =
tokenizer.texts_to_sequences(df_chat_gpt['clean_joined'])

print("Токенізований текст: ", tokenized)

padded = pad_sequences(tokenized,maxlen = 40, padding =
'post', truncating = 'post')

print("Набивка:", padded)

pred = model.predict(padded)

print("Імовірність що новина неправдива:",pred)

predict(df_chat_gpt)
```

ДОДАТОК Б НОВИНА 3 СНАТ-GPT

Title: "Unprecedented Scandal Unearthed: Shocking Revelations Rock the US Elections"

Text:

In a stunning turn of events, a bombshell revelation has surfaced, shattering the very core of the US elections. Unveiling a web of deceit and corruption, this unprecedented scandal threatens to undermine the integrity of the democratic process and raise serious questions about the fairness of the recently concluded elections.

Subject: US Elections

As the dust settles after one of the most contentious elections in US history, an investigation by a team of renowned journalists has unearthed a series of shocking discoveries that challenge the legitimacy of the electoral process. Deep within the shadowy corridors of power, hidden from public scrutiny, a grand conspiracy was at play, manipulating the outcomes and subverting the will of the American people.

The evidence gathered reveals a complex network of high-level operatives operating behind the scenes, orchestrating a meticulously planned operation to sway the election results. Covert meetings, undisclosed transactions, and clandestine collaborations were the tools employed to engineer a rigged system that favored a particular candidate, casting a dark cloud over the democratic principles that the nation holds dear.

The investigation further exposes an intricate web of foreign interference aimed at influencing the election outcome. Foreign actors, acting with nefarious intent, deployed sophisticated cyber tactics and disinformation campaigns to sow seeds of doubt and manipulate public opinion. Their insidious actions strike at the heart of the democratic process, threatening the very foundation upon which the nation stands.

The implications of these revelations are staggering. They raise fundamental questions about the credibility of the electoral system and the sanctity of the democratic process. The American people, who have entrusted their voices to be heard through the ballot box, now find themselves confronting an unsettling reality where their voices may have been drowned out by deception and subversion.

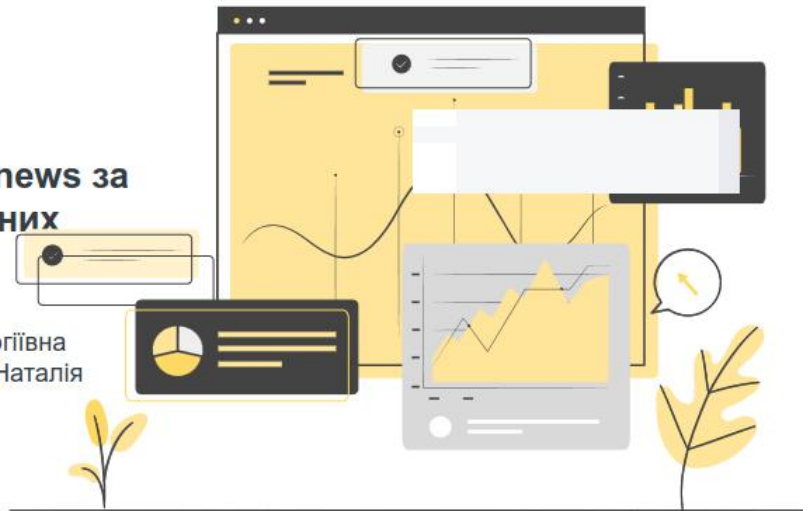
As the nation grapples with the aftermath of this unprecedented scandal, calls for an independent, impartial investigation are growing louder. Citizens demand transparency, accountability, and a thorough examination of the electoral machinery to restore faith in the democratic process. The integrity of future elections hangs in the balance as the nation seeks to confront this dark chapter head-on and pave the way for a more transparent, fair, and resilient electoral system.

ДОДАТОК В ПРЕЗЕНТАЦІЯ

Слайд 1

Ідентифікація fake news за допомогою нейронних мереж

Виконав: Шафорост Анна Сергіївна
Науковий керівник: Шаповал Наталія Віталіївна



Слайд 2

Предмет та об'єкт досліджень

Об'єкт - неправдива інформація і fake news у новинних виданнях.

Предмет дослідження - моделі нейронних двонаправлених мереж ДКПЧ

Мета роботи - створення моделі класифікації текстової інформації на основі нейронних мереж для виявлення неправдивої інформації.

Слайд 3

Що таке Fake News?

Fake news - це неправдива або оманлива інформація що презентується як новина. Неправдиві новини часто мають на меті запламувати репутацію об'єкту новини.



Слайд 4

Актуальність задачі

Fake news є важливим елементом інформаційної війни - вони допомагають маніпулювати думками спільнот, дискредитувати державні органи управління, розпалювати ворожнечу - усе аби створити соціальне напруження та дестабілізувати політичні, економічні стосунки всередині країни та поза її межами.

Слайд 5

Постановка задачі

Задачею дослідження є побудова моделі класифікації текстової інформації на основі нейронних мереж для виявлення неправдивої інформації. В ході роботи, потрібно проаналізувати ефективність моделі на реальних даних новинних видань.

Слайд 6

Підходи до класифікації текстової інформації

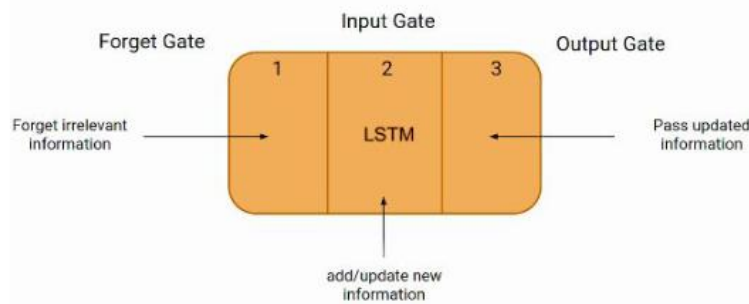
- Лінгвістичний підхід: зосереджується на лінгвістичних особливостях новинних статей, таких як стилістичні особливості (наприклад, сегментація речень, токенизація та маркування частин мови) та лексичні особливості (наприклад, словниковий запас, частота вживання слів). Аналізуючи ці ознаки, дослідники можуть виявити оманливі підказки або стилі написання, які часто асоціюються з фейковими новинами.
- Тематично-агностичний підхід: спрямований на виявлення фейкових новин, не покладаючись на конкретну тему або тематику новинної статті. Він зосереджується на загальних характеристиках фейкових новин, таких як структура, стиль і подача контенту.
- Методи машинного навчання: для виявлення фейкових новин використовують методи глибокого навчання, такі як згорткові нейронні мережі (CNN) і довга короткочасна пам'ять (LSTM)

Сучасні рішення часто можуть комбінувати вищенаведені підходи для більш точних результатів.

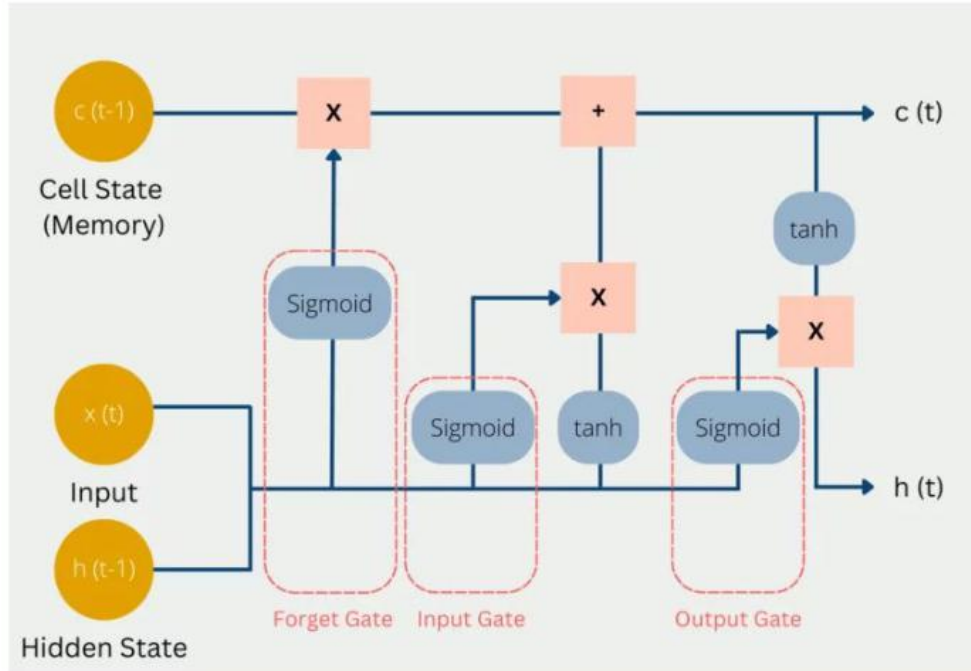
Слайд 7

Довга короткочасна пам'ять (ДКЧП)

Мережі з довгою короткочасною пам'яттю (LSTM) - це тип рекурентних нейронних мереж (RNN), розроблений для вирішення проблеми короткочасної пам'яті в RNN. Вони можуть навчатися і підтримувати довгострокові залежності, використовуючи внутрішні механізми, які називаються шарами або гейтами, що регулюють потік інформації. Ці шари можуть вирішувати, які дані в послідовності необхідно зберегти, а які відкинути, що дозволяє LSTM передавати відповідну інформацію через довгий ланцюжок послідовностей для здійснення прогнозів.

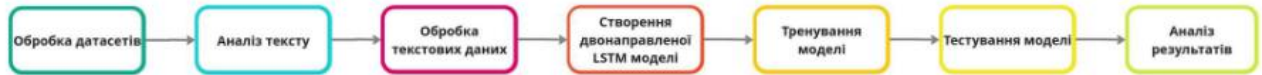


Слайд 8



Слайд 9

Етапи розв'язку задачі



Слайд 9

Початкові дані

Для нашої задачі ми використаємо 2 датасети: 20,000+ правдивих і неправдивих новин. Правдиві новини зібрані з видання Reuters.com, неправдиві - з декількох сайтів визнаних ненадійними факт-чекінговою організацією Politifact.

Датасети мають наступний вигляд: "назва статті, текст статті, тип новини, дата публікації":

	title	text	subject	date
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	December 31, 2017
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	December 30, 2017
4	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNews	December 29, 2017

Слайд 10

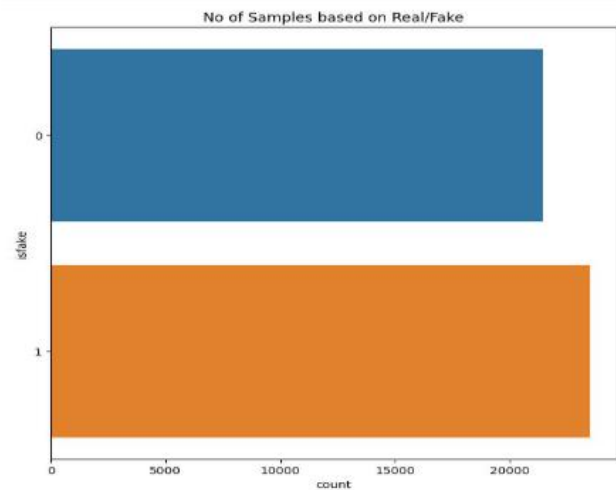
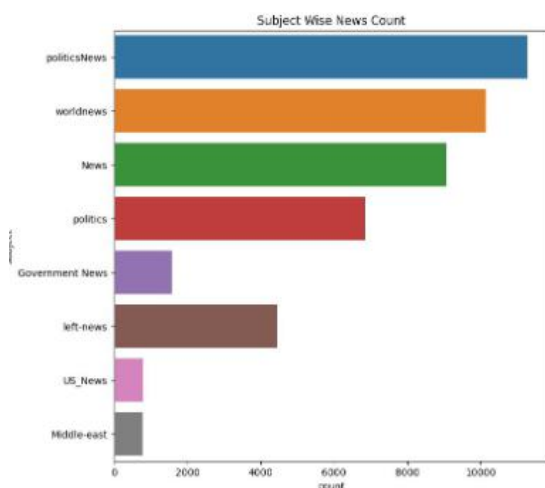
Обробка даних

Ці датасети необхідно доповнити властивістю isFake: для правдивих новин isFake=0, для неправдивих isFake=1. Далі потрібно об'єднати два датасети, отримаємо наступний результат

	title	text	subject	isfake	original
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	0	As U.S. budget fight looms, Republicans flip t...
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	0	U.S. military to accept transgender recruits o...
2	Senior U.S. Republican senator: 'Let Mr. Muefl...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	0	Senior U.S. Republican senator: 'Let Mr. Muefl...
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	0	FBI Russia probe helped by Australian diplomat...
4	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNews	0	Trump wants Postal Service to charge 'much mor...
...
44893	McPain: John McCain Furious That Iran Treated ...	21st Century Wire says As 21WIRE reported earl...	Middle-east	1	McPain: John McCain Furious That Iran Treated ...
44894	JUSTICE? Yahoo Settles E-mail Privacy Class-ac...	21st Century Wire says It s a familiar theme. ...	Middle-east	1	JUSTICE? Yahoo Settles E-mail Privacy Class-ac...
44895	Sunnistan: US and Allied 'Safe Zone' Plan to T...	Patrick Henningsen 21st Century WireRemember ...	Middle-east	1	Sunnistan: US and Allied 'Safe Zone' Plan to T...
44896	How to Blow \$700 Million: Al Jazeera America F...	21st Century Wire says Al Jazeera America will...	Middle-east	1	How to Blow \$700 Million: Al Jazeera America F...
44897	10 U.S. Navy Sailors Held by Iranian Military ...	21st Century Wire says As 21WIRE predicted in ...	Middle-east	1	10 U.S. Navy Sailors Held by Iranian Military ...

Слайд 11

Аналіз тексту



Слайд 14

Токенізація тексту

```
[24] # Splitting Data Into Test And Train
# !pip install scikit-learn
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(df.clean_joined, df.isfake, test_size = 0.2)

[25] from nltk import word_tokenize

[26] x_train

14132  erdogan says turkey iran russia agreed carry p...
27704  democrats literally scream shame republicans b...
15615  japan says time exert maximum pressure north k...
18260  france offers mediate baghdad kurds paris reut...
13306  israeli missiles military position near damasc...
...
4409   attorney bharara takes trump criticism jokes r...
6168   trump picks member lead agency bloomberg washi...
35152  hypocrite hillary gives speech evils inequalit...
2578   trump pleads republicans roll obamacare washin...
14181  kremlin says following situation ukraine luhan...
Name: clean_joined, Length: 35918, dtype: object

[30] # Creating A Tokenizer To Tokenize The Words And Create Sequences Of Numbers
tokenizer = Tokenizer(num_words = total_words)
tokenizer.fit_on_texts(x_train)
train_sequences = tokenizer.texts_to_sequences(x_train)
test_sequences = tokenizer.texts_to_sequences(x_test)

[31] print("The encoding for document\n",df.clean_joined[0],"\n is : \n\n",train_sequences[0])

The encoding for document
budget fight looms republicans flip fiscal script washington reuters head conservative republican faction congress voted month huge expansion national debt cuts called fiscal cc
is :

[1276, 47, 360, 105, 37, 537, 1200, 35, 1277, 120, 7907, 37, 9, 360, 105, 37, 537, 1200, 4402, 271, 35, 1277, 120, 943, 3, 3540, 1276, 2, 79, 392, 962, 21, 309, 70, 3, 1121, 300
```

Слайд 15

Набивка (padding)

```
✓ [32] # Adding Padding
padded_train = pad_sequences(train_sequences,maxlen = 40, padding = 'post', truncating = 'post')
padded_test = pad_sequences(test_sequences,maxlen = 40, truncating = 'post')

✓ [33] for i,doc in enumerate(padded_train[:2]):
    print("The padded encoding for document",i+1," is : ",doc)

The padded encoding for document 1 is : [1276  47  360  105  37  537 1200  35 1277  120 7907  37  9  360
 105  37  537 1200 4402  271  35 1277  120  943  3 3540 1276  2
 79 392 962  21 309  70  3 1121 300 891  3 4130]
The padded encoding for document 2 is : [ 75 1248 7410 2740  38 2574 1920  871  27  8  75 3632
 159 1205 2040 3528  8 1484 1248 9579  38 1443 287 1920
 871 2423 116 1389  38 2479 1920  88  75 558 173 524
1229  22 13347  871]
```

Слайд 16

Параметри моделі

```
# Sequential Model
model = Sequential()

# Embedding layer
model.add(Embedding(total_words, output_dim = 128))

# Bi-Directional LSTM
model.add(Bidirectional(LSTM(128)))

# Dense layers
model.add(Dense(128, activation = 'relu'))
model.add(Dense(1, activation= 'sigmoid'))
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['acc'])
model.summary()
```

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, None, 128)	13914112
bidirectional (Bidirectional)	(None, 256)	263168
dense (Dense)	(None, 128)	32896
dense_1 (Dense)	(None, 1)	129

Total params: 14,210,305
Trainable params: 14,210,305
Non-trainable params: 0

Слайд 17

Результати роботи

Модель на 99% точно може відрізнити правдиву і неправдиву інформацію та має мінімальні показники хибних результатів:

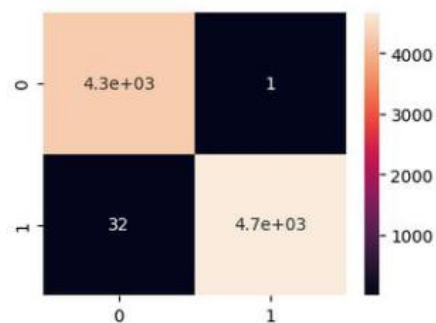
```
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(list(y_test), prediction)
print("Model Accuracy : ", accuracy)
```

Model Accuracy : 0.9963251670378619

Classification report:

	precision	recall	f1-score	support
0	0.99	1.00	1.00	4279
1	1.00	0.99	1.00	4701
accuracy			1.00	8980
macro avg	1.00	1.00	1.00	8980
weighted avg	1.00	1.00	1.00	8980

Матриця Невідповідностей (Confusion Matrix):



Слайд 18

Перевірка моделі на незалежних даних

При перевірці моделі на новині, згенерованій за допомогою Chat-GPT, отримано задовільний результат 99% - новина неправдива

```
predict(df_chat_gpt)
```

```
Ключові слова заданої новини: stunning turn events bombshell revelation surfaced shattering core ele  
Токенізований текст: [[17, 18, 19, 20, 21, 22, 23, 24, 2, 25, 26, 27, 7, 8, 28, 29, 9, 3, 1, 10, 11,  
Набивка: [[17 18 19 20 21 22 23 24 2 25 26 27 7 8 28 29 9 3 1 10 11 30 31 32  
2 33 34 35 2 36 5 37 38 39 40 41 42 43 44 45]]  
1/1 [=====] - 0s 30ms/step  
Імовірність що новина неправдива: [[0.99996036]]
```

Слайд 19

Висновки

Модель ідентифікації fake news за допомогою двонаправленої LSTM мережі показала задовільні результати з точністю понад 99%.