Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему *Метод виявлення фейкових новин для оцінки достовірності джерел масової інформації*

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

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

Освітня програма *Комп'ютерні науки*
Назва освітньої програми

Виконав: *студент 2 курсу, група КНм-22-1* _____ *Д.О. Боровик*
Курс, група виконавця — Підпис — Ініціали, прізвище

Керівник: *д.т.н., професор, зав. кафедри КН* _____ *О.В. Бармак*
Науковий ступінь, посада — Підпис — Ініціали, прізвище

Нормоконтроль: *к.т.н., доцент кафедри КН* _____ *Р.О. Багрій*
Науковий ступінь, посада — Підпис — Ініціали, прізвище

До захисту допускаю:
Зав. кафедри КН, д.т.н., професор _____ *О.В. Бармак*
Підпис — Ініціали, прізвище

10 грудня 2023 р.

Хмельницький 2023

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

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

ЗАТВЕРДЖУЮ
Завідувач кафедри *комп'ютерних наук*

_____
(підпис)
д.т.н., професор О.В. Бармак
«*01*» *вересня* *2023* року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

1. Тема кваліфікаційної роботи магістра: *«Метод виявлення фейкових новин для оцінки достовірності джерел масової інформації»*

2. Завдання видано студенту *Боровику Дмитру Олеговичу*
(прізвище, ім'я, по батькові)

3. Керівник роботи *завідувач кафедри КН Бармак Олександр Володимирович*
(прізвище, ім'я, по батькові)

4. Затверджені наказом університету від «*15*» *серпня* *2023* р. № *30*

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

Мета кваліфікаційної роботи магістра – розробка методу виявлення в Інтернеті фейкових новин нейромережевими засобами. Для досягнення поставленої мети визначено наступні задачі: провести аналіз існуючих моделей і підходів для виявлення в Інтернеті фейкових новин; розробити метод виявлення в Інтернеті фейкових новин нейромережевими засобами; підготувати набір даних для навчання нейронної мережі; провести навчання нейронної мережі виявляти фейкові новини; розробити інформаційну систему, що реалізує запропонований метод; оцінити отримані результати виявлення в Інтернеті фейкових новин запропонованим методом за загальними статистичними показниками.

Реферат

Кваліфікаційна робота магістра присвячена розробці методу виявлення в Інтернеті фейкових новин за допомогою нейромережевих засобів.

Актуальність теми. Інтернет на сьогодні зайняв перше місце серед джерел інформації. Суттєво зросла роль онлайн соціальних мереж (ОСМ). Стрімкий технологічний розвиток, що спостерігається в останні роки, особливо в галузі мобільних пристроїв, зробив соціальні медіа, такі як Facebook, Twitter, Instagram та Sina Weibo, доступними у повсякденному житті. Онлайн соціальні мережі стали не лише засобом комунікації між людьми, а й інструментом обміну інформацією та формування громадської думки. Соціальні медіа надали віртуальне середовище для публікацій, обговорення, обміну поглядами та глобальної взаємодії між користувачами, без обмежень за місцем, часом або обсягом контенту. Основна причина використання соціальних медіа користувачами у всьому світі – отримання новин та оперативне слідкування за поточними подіями. Структура соціальних медіа дозволяє поширювати новини в режимі реального часу, незалежно від достовірності цих новин.

Однак, в останній період ОСМ відіграють не лише позитивну, а й негативну роль. Це пов’язано із забезпеченням ними поширення фейкових новин (ФН). Дослідження, проведене нещодавно, показало, що 23% осіб у США ділилися ФН, як навмисно, так і ненавмисно. Результатом поширення ФН, як правило, є страх, паніка та фінансові втрати.

На даний час існують різні апробовані підходи до виявлення ФН. Зокрема, один з підходів базується на використанні різних алгоритмів машинного та глибокого навчання. Інший – на використанні результатів аналізу сентименту новинного контенту та аналізу емоцій у коментарях користувачів. Існує і ряд інших підходів, які заслуговують на увагу, подальший аналіз і дослідження. При цьому кожен із них характеризується певним рівнем ефективності на різних масивах даних. Здійснений аналіз різних підходів виявлення ФН дозволив зробити висновок про те, що існуючі підходи є дієвими і перспективними в частині підвищення їх ефективності для

розробки нових моделей на різних наборах даних. Однак, зважаючи на стрімкий розвиток в останній період штучного інтелекту, одним із перспективних підходів вбачається використання потенціалу нейромережевих технологій.

Тому розробка методу виявлення фейкових новин на основі використання нейромережевих технологій вбачається перспективним дослідженням, результати якого можуть сприяти мінімізації зазначених вище негативних наслідків від поширення фейкових новин в Інтернеті.

Мета і задачі роботи. Метою кваліфікаційної роботи магістра є розробка методу виявлення в Інтернеті фейкових новин нейромережевими засобами.

Для досягнення поставленої мети визначено наступні задачі:

- провести аналіз існуючих моделей і підходів для виявлення в Інтернеті фейкових новин;
- розробити метод виявлення в Інтернеті фейкових новин нейромережевими засобами;
- підготувати набір даних для навчання нейронної мережі;
- провести навчання нейронної мережі виявляти фейкові новини;
- розробити інформаційну систему, що реалізує запропонований метод;
- оцінити отримані результати виявлення в Інтернеті фейкових новин запропонованим методом за загальними статистичними показниками.

Об’єкт дослідження – процес виявлення в Інтернеті фейкових новин нейромережевими засобами.

Предмет дослідження – моделі нейронної мережі, методи виявлення в Інтернеті фейкових новин.

Методи дослідження, що застосовані для вирішення поставлених завдань: для виявлення в Інтернеті фейкових новин – метод застосування згорткової нейронної мережі; для програмної реалізації методу виявлення фейкових новин – методи сучасних інформаційних технологій; для дослідження ефективності методу виявлення фейкових новин – методи вищої математики, теорії ймовірностей і математичної статистики.

Наукова новизна одержаних результатів. У результаті проведеної роботи були отримані наступні результати:

– розроблено метод виявлення в Інтернеті фейкових новин нейромережевими засобами. Сутність новизни методу полягає в удосконаленні структури багатошарової CNN нейромережі за рахунок додавання шару випадкового відключення (Dropout layer), збільшення розміру ядра, зміни функції активації одновимірного шару максимального об'єднання з сигмоїдальної функції активації на функцію активації ReLU;

– розроблено інформаційну систему реалізації запропонованого методу виявлення в Інтернеті фейкових новин нейромережевими засобами. Для розробки backend частини додатку використано мову програмування PHP та фреймворк Laravel, для frontend частини - фреймворк Vue.js, для реалізації стилістичної складової веб-застосунку - фреймворк Tailwind CSS, а безпосередньо методу виявлення фейкових новин - мову програмування Python та бібліотеки TensorFlow, NumPy, Scikit-learn для роботи з нейромережевими засобами.

Апробація результатів кваліфікаційної роботи магістра та публікації. Основні положення та результати роботи доповідалися на наукових конференціях:

- Боровик Д. О. Актуальність задачі автоматизації виявлення фейкових новин і огляд підходів та інформаційних систем, що її реалізують // Матеріали Всеукраїнської науково-практичної Інтернет-конференції (13-19 березня 2023 року) «Автоматизація та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку» – Черкаси: ЧНУ, 2023. – С. 45-47;
- Боровик Л. В., Боровик Д. О. Підвищення інформаційної ефективності виявлення недостовірної інформації в Інтернеті // Збірник тез доповідей XIX Міжнародної науково-практичної конференції (10 листопада 2023 року) «Військова освіта і наука: сьогодення та майбутнє». - К.: ВІКНУ, 2023. – С. 23-24.

Також, результати роботи опубліковані у фаховому науковому виданні:

- Боровик Д. О., Бармак О. В. Удосконалений метод виявлення фейкових новин на основі використання CNN нейромережі // Науковий журнал «Вісник

Хмельницького національного університету. Серія: Технічні науки» Хмельницький, 2023.

Структура та обсяг роботи. Кваліфікаційна робота магістра складається з завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань із 79 найменувань та 3 додатків. Загальний обсяг кваліфікаційної роботи магістра становить 140 сторінок, з них 87 сторінок основного тексту та 36 сторінок додатків. У роботі наведено 20 рисунків.

Ключові слова: Інтернет, інформаційні технології, виявлення, фейкові новини, нейромережі, згорткова нейронна мережа, CNN, метод.

Зміст

Перелік скорочень

Скорочення, термін, позначення	Пояснення
ОСМ	Онлайн соціальні мережі
ML	Алгоритми машинного навчання
DL	Алгоритми глибокого навчання
ФН	Фейкові новини
ІС	Інформаційна система

Вступ

Актуальність теми. Інтернет на сьогодні зайняв перше місце серед джерел інформації. Суттєво зросла роль онлайн соціальних мереж (ОСМ). Стрімкий технологічний розвиток, що спостерігається в останні роки, особливо в галузі мобільних пристроїв, зробив соціальні медіа, такі як Facebook, Twitter/X, Instagram та Sina Weibo, невід'ємною частиною нашого повсякденного життя. ОСМ стали не лише засобом комунікації між людьми, а й інструментом обміну інформацією та формування громадської думки. Соціальні медіа надали віртуальне середовище для публікацій, обговорення, обміну поглядами та глобальної взаємодії між користувачами, без обмежень за місцем, часом або обсягом контенту. Основна причина використання соціальних медіа користувачами у всьому світі – отримання новин та оперативне слідкування за поточними подіями. Структура соціальних медіа дозволяє поширювати новини в режимі реального часу та швидко, незалежно від достовірності цих новин.

Однак, в останній період ОСМ відіграють не лише позитивну, а й негативну роль. Це пов’язано із забезпеченням ними поширення фейкових новин (ФН). Дослідження, проведене нещодавно, показало, що 23% осіб у США ділилися ФН, як навмисно, так і ненавмисно. Результатом поширення ФН, як правило, є страх, паніка та фінансові втрати [12].

На даний час існують різні апробовані підходи до виявлення ФН. Зокрема, один з підходів базується на використанні різних алгоритмів машинного та глибокого навчання. Інший – на використанні результатів аналізу сентименту новинного контенту та аналізу емоцій у коментарях користувачів. Існує і ряд інших підходів, які заслуговують на увагу, подальший аналіз і дослідження. При цьому кожен із них характеризується певним рівнем ефективності на різних масивах даних. Здійснений аналіз різних підходів виявлення ФН дозволив зробити висновок про те, що існуючі підходи є дієвими і перспективними в частині підвищення їх ефективності для розробки нових моделей на різних наборах даних. Однак, зважаючи на стрімкий

розвиток в останній період штучного інтелекту, одним із перспективних підходів вбачається використання потенціалу нейромережевих технологій.

Отже, розробка методу виявлення фейкових новин на основі використання нейромережевих технологій вбачається перспективним дослідженням, результати якого можуть сприяти мінімізації зазначених вище негативних наслідків від поширення фейкових новин в Інтернеті.

Мета і задачі роботи. Метою кваліфікаційної роботи магістра є розробка методу виявлення в Інтернеті фейкових новин нейромережевими засобами.

Для досягнення поставленої мети визначено наступні задачі:

- провести аналіз існуючих моделей і підходів для виявлення в Інтернеті фейкових новин;
- розробити метод виявлення в Інтернеті фейкових новин нейромережевими засобами;
- підготувати набір даних для навчання нейронної мережі;
- провести навчання нейронної мережі виявляти фейкові новини;
- розробити інформаційну систему, що реалізує запропонований метод;
- оцінити отримані результати виявлення в Інтернеті фейкових новин запропонованим методом за загальними статистичними показниками.

Об’єкт дослідження – процес виявлення в Інтернеті фейкових новин нейромережевими засобами.

Предмет дослідження – моделі нейронної мережі, методи виявлення в Інтернеті фейкових новин.

Методи дослідження, що застосовані для вирішення поставлених завдань: для виявлення в Інтернеті фейкових новин – метод застосування згорткової нейронної мережі; для програмної реалізації методу виявлення фейкових новин – методи сучасних інформаційних технологій; для дослідження ефективності методу виявлення фейкових новин – методи вищої математики, теорії ймовірностей і математичної статистики.

Наукова новизна одержаних результатів. У результаті проведеної роботи були отримані наступні результати:

– розроблено метод виявлення в Інтернеті фейкових новин нейромережевими засобами. Сутність новизни методу полягає в удосконаленні структури багатошарової CNN нейромережі за рахунок додавання шару випадкового відключення (Dropout layer), збільшення розміру ядра, зміни функції активації одновимірного шару максимального об'єднання з сигмоїдальної функції активації на функцію активації ReLU;

– розроблено інформаційну систему реалізації запропонованого методу виявлення в Інтернеті фейкових новин нейромережевими засобами. Для розробки backend частини додатку використано мову програмування PHP та фреймворк Laravel, для frontend частини - фреймворк Vue.js, для реалізації стилістичної складової веб-застосунку - фреймворк Tailwind CSS, а безпосередньо методу виявлення фейкових новин - мову програмування Python та бібліотеки TensorFlow, NumPy, Scikit-learn для роботи з нейромережевими засобами.

Апробація результатів кваліфікаційної роботи магістра та публікації. Основні положення та результати роботи доповідалися на наукових конференціях:

- Боровик Д. О. Актуальність задачі автоматизації виявлення фейкових новин і огляд підходів та інформаційних систем, що її реалізують // Матеріали Всеукраїнської науково-практичної Інтернет-конференції (13-19 березня 2023 року) «Автоматизація та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку» – Черкаси: ЧНУ, 2023. – С. 45-47;
- Боровик Л. В., Боровик Д. О. Підвищення інформаційної ефективності виявлення недостовірної інформації в Інтернеті // Збірник тез доповідей XIX Міжнародної науково-практичної конференції (10 листопада 2023 року) «Військова освіта і наука: сьогодення та майбутнє». - К.: ВІКНУ, 2023. – С. 23-24.

Також, результати роботи опубліковані у фаховому науковому виданні:

- Боровик Д. О., Бармак О. В. Удосконалений метод виявлення фейкових новин на основі використання CNN нейромережі // Науковий журнал «Вісник

Хмельницького національного університету. Серія: Технічні науки» Хмельницький, 2023.

Структура та обсяг роботи. Кваліфікаційна робота магістра складається з завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань із 79 найменувань та 3 додатків. Загальний обсяг кваліфікаційної роботи магістра становить 140 сторінок, з них 87 сторінок основного тексту та 36 сторінок додатків. У роботі наведено 20 рисунків.

Ключові слова: Інтернет, інформаційні технології, виявлення, фейкові новини, нейромережі, згорткова нейронна мережа, CNN, метод.

РОЗДІЛ 1
Аналіз сучасного стану використання інформаційних технологій для виявлення в Інтернеті фейкових новин

1.1. Аналіз предметної області

До недавнього часу люди отримували новини та інформацію переважно з газет і телевізійних каналів. Проте з появою Інтернету ситуація змінилася кардинально. Інтернет зайняв перше місце серед джерел інформації [1]. Суттєво зросла роль ОСМ [2-4]. Стрімкий технологічний розвиток, що спостерігається в останні роки, особливо в галузі мобільних пристроїв, зробив соціальні медіа, такі як Facebook, Twitter/X, Instagram та Sina Weibo, невід'ємною частиною нашого повсякденного життя [2, 5]. У результаті дослідження, проведеного в останній період, встановлено, що 67% осіб у США отримують новини в основному з соціальних мереж [6]. Згідно з Global Overview Report (https://datareportal.com/reports/digital-2021-global-overview-report, доступно 11 листопада 2021 року) кількість людей у світі з обліковими записами в ОСМ у 2021 році досягла рівня приблизно 4,20 мільярди, що складає понад 53% загального населення світу [7]. ОСМ стали не лише засобом комунікації між людьми, а й інструментом обміну інформацією та формування громадської думки [8]. Соціальні медіа надали віртуальне середовище для публікацій [9], обговорення, обміну поглядами та глобальної взаємодії між користувачами [10], без обмежень за місцем, часом або обсягом контенту [11].

Основна причина використання соціальних медіа користувачами у всьому світі – отримання новин та оперативне слідкування за поточними подіями.

Багато людей використовують соціальні медіа для публікації новин та інформації через свої облікові записи або сторінки, оскільки публікація новин на цих платформах відрізняється від публікації у традиційних ЗМІ тим, що не займає багато часу, не потребує витрат і не підлягає обмеженням аудиту [5, 12]. Структура соціальних медіа дозволяє поширювати новини в режимі реального часу та швидко,

незалежно від достовірності цих новин [3]. Наприклад, у США в 2012 році 49% користувачів ділилися новинами в соціальних мережах. Згідно з доповіддю Pew Research Center (https://www.pewresearch.org/, доступно 23 серпня 2021 року), у 2016 році понад 62% користувачів щоденно отримували новини з ОСМ [12], а у 2018 році дві третини дорослого населення США отримували новини з платформ соціальних медіа [13].

Однак, в останній період ОСМ відіграють не лише позитивну, а й негативну роль. Це пов’язано із забезпеченням ними поширення фейкових новин. Дослідження, проведене в 2016 році, показало, що 23% осіб у США ділилися фейковими новинами, як навмисно, так і ненавмисно [14]. У Китаї більше третини інформаційних повідомлень, які стають трендами на мікроблогах, є фейковою інформацією [15]. Деякі популярні джерела інформації, що вважаються надійними (наприклад, Вікіпедія), також поширюють невірну інформацію або фейкові новини [16].

Фейкові новини можна визначити як опубліковані новинні статті, що містять неправдиву інформацію з метою свідомого введення читачів в оману [2] або здійснення зловживань [17]. Зазвичай у фейкових новинах містяться високопровокаційні повідомлення, створення яких переслідує фінансову чи політичну цілі. У 2017 році «фейкові новини» за версією словника [5] були оголошені офіційним словом року.

Згідно з доповіддю GDATA, 59% користувачів ОСМ стикалися з неправдивою інформацією [18]. Понад 57% користувачів ОСМ очікували, що опубліковані новини будуть неточними [13]. Сайт Statista представив статистику (https://www.statista.com/statistics/649221/fake-news-expose-responsible-usa/, доступно 4 березня 2021 року) за 27 серпня 2019 року, що базується на опитуванні, проведеному у США в 2018 році, щодо того, як ОСМ відповідають за поширення фейкових новин. Згідно наведених даних, 29% учасників вважають, що за поширення фейкових новин в основному відповідають соціальні медіа, а 60% відмітили, що ці платформи за поширення фейкових новин відповідають частково. Фейкові новини мають на 70% більше шансів поширюватися, ніж новини реальні [12]. Згідно з

дослідженнями щодо швидкості поширення фейкових новин, твіти, які містять фальсифіковану інформацію в Twitter, досягають користувачів у шість разів швидше, ніж надійні твіти [19]. Швидкість поширення фейкових новин в Інтернеті більша, ніж реальних [12], оскільки люди цікавляться новою інформацією або новинами [20] та схильні ділитися останньою інформацією [21], особливо розповідати про важливі новини, не перевіряючи їх достовірність [22]. Повторний перегляд фейкових новин робить їх одержувачам знайомими, підвищує їх вірогідність і призводить до їх поширення як реальних новин [23].

Фейкові новини впливають на повсякденне життя людей [1, 8], маніпулюють їхніми думками та почуттями [24, 25], змінюють їхні переконання [26] і можуть призвести до прийняття неправильних рішень [1]. Розповсюдження фейкових новин у соціальних мережах негативно впливає на суспільство [27, 28] у багатьох сферах, зокрема таких, як політична, економічна, соціальна, медична, технологічна та спортивна [15, 26]. Основними намірами поширення фейкових новин є фінансовий зиск, поширення ненависті на підставі екстремістських мотивів, маніпулювання свідомістю людей з політичних міркувань або створення упереджених думок з виборчих міркувань [29, 30] тощо. Результатом поширення фейкових новин, як правило, є страх, паніка та фінансові втрати [28].

Все це вказує на негативні наслідки поширення фейкових новин і їх негативний соціальний вплив.

Про негативну роль фейкових новин можна судити, зокрема з таких прикладів. Одна з новин на Reddit спричинила реальну стрілянину (https://www.rollingstone.com/politics/politics-news/anatomy-of-a-fake-news-scandal-125877/, доступно на 15 березня 2022 року). Під час виборчої кампанії до виборів президента США у 2016 році було виявлено понад мільйон постів, пов'язаних із фейковою новиною, відомою як PIZZAGATE (https://tinyurl.com/z38z5zh, доступно на 15 червня 2022 року). У фейковій новині про мусульманську спільноту в Індії (https://thelogicalindian.com/fact-check/muslim-spit-restaurant-covid-19 -coronavirus-20457, доступно 16 липня 2021 року) на підставі доповіді BBC

(https://www.bbc.com/news/world-asia-india-53165436, доступно 16 липня 2021 року) стверджувалось, що мусульмани намірено поширюють COVID-19. Це призвело до збільшення ворожнечі до мусульман і закликів до економічного бойкоту. Крім того, 20 найпопулярніших фейкових новин були більш обговорюваними, ніж 20 найпопулярніших реальних історій (https://tinyurl.com/y8dckwhr, доступно на 15 червня 2022 року).

У 2021 році Facebook оголосив, що було закрито близько 1,3 мільярда фейкових облікових записів та видалено понад 12 мільйонів публікацій з хибною інформацією про COVID-19 та вакцини [31].

У багатьох людей виникають труднощі у відрізненні фейкових новин від реальних, незалежно від статі, віку чи рівня освіти [16]. Відрізнити фейкові новини від реальних складно, оскільки, як свідчать наукові дослідження, людська здатність розрізняти правдиву та неправдиву інформацію відносно незначна і становить близько 54% [32].

Оскільки фейкові новини стали глобальним викликом і великою загрозою для демократії, економіки та мирного співіснування [33], різні суб'єкти (громадські організації, журналісти, політики, дослідники) працюють над зменшенням ризику [12].

Отже, проблема розповсюдження фейкових новин в ОСМ на даний час є глобальною, а формування механізмів протидії – актуальним завданням сьогодення. Її вирішення пов'язується з формуванням моделей, які виявляють фейкові новини та обмежують можливість їх поширення [34].

1.2 Аналіз існуючих підходів щодо виявлення в Інтернеті фейкових новин

На даний час існують різні апробовані підходи до виявлення фейкових новин. Зокрема, один з підходів базується на використанні різних алгоритмів машинного (ML) та глибокого навчання (DL). Інший – на використанні результатів аналізу сентименту новинного контенту та аналізу емоцій у коментарях користувачів. Існує і

ряд інших підходів, які заслуговують на увагу, подальший аналіз і дослідження. При цьому кожен із них характеризується певним рівнем ефективності на різних масивах даних.

Слід зауважити, що для перевірки ефективності підходів часто застосовуються такі набори даних, як LIAR, FakeNewsNet-PolitiFact, FakeNewsNet-GossipCop і COVID-19.

LIAR – це великий, доступний для широкого загалу набір даних фейкових новин [35]. Набір даних містить близько 12800 записів і дві основні складові: профілі користувачів і короткі політичні заяви. Характеристики профілю користувача включають ім'я спікера, роботу, партійну належність, штат, кредитну історію та контекст. Заяви (зроблені в період з 2007 по 2016 рік) були позначені редакцією Politifact.com за допомогою шести деталізованих категорій: правда, переважно правда, наполовину правда, майже неправда, неправда і шар зігрівається. Ці шість міток є відносно збалансованими за розміром. Загалом кожна заява має свою пов'язану мітку та інформацію про автора цієї заяви.

FakeNewsNet - це комплексний набір даних (https://github.com/KaiDMML/FakeNewsNet, доступно на 20 березня 2022 року), який складається з повних текстових новинних статей, зібраних з веб-сайтів politifact.com і gossipcop.com (доступно на 18 березня 2022 року). Кожен з них містить новинні статті та інформацію про соціальний контекст.

COVID-19 - це збірник соціальних медіа-повідомлень, коментарів і новин, пов'язаних із COVID-19, класифікованих як реальні або фейкові на основі їхньої правдивості. Набір даних [36] був зібраний з різних соціальних медіа-платформ, таких як Twitter і YouTube. Крім того, організатори збірника зібрали 10700 соціальних медіа-повідомлень та новинних статей про COVID-19 у формі анотованого набору даних англійською мовою.

Також слід зауважити, що критеріями оцінки, які широко використовуються в завданнях класифікації тексту, часто є точність (A), точність (P), чутливість (R) та $F1$-показник. Також при роботі з незбалансованим набором даних для оцінки

продуктивності моделей в якості міри продуктивності часто використовується площа під кривою ROC (AUC), яка є мірою для порівняння алгоритмів навчання та побудови оптимальних моделей навчання [37-39].

До переліку класичних алгоритмів ML прийнято відносити логістичну регресію (LR), метод опорних векторів (SVM), дерево рішень (DT), наївний байєсівський класифікатор (NB), випадковий ліс (RF), XGBoost (XGB), а також комбінацію цих алгоритмів. До високорівневих алгоритмів ML відносять згорткові нейронні мережі (CNN), двонаправлені рекурентні мережі з короткотривалою пам'яттю (BiLSTM), двонаправлені рекурентні мережі з вентильними блоками (BiGRU), комбінації CNN-BiLSTM і CNN-BiGRU, а також гібридний підхід на основі цих технік. Моделями на основі глибокого навчання є моделі BERTbase та RoBERTabase.

У роботі [40] представлено огляд підходів до виявлення фейкових новин на основі використання алгоритмів машинного навчання ML з двома сценаріями методів представлення слів (статистичними та контекст-незалежними). Крім того, у роботі [40] проведено порівняльну оцінку восьми передових моделей машинного навчання, а саме CNN, BiLSTM, BiGRU, CNN-BiLSTM, CNN-BiGRU, різних гібридних моделей з двома типами моделей текстового представлення (контекст-незалежними і контекст-свідомими моделями вбудови), BERTbase, RoBERTabase.

Авторами роботи [40] встановлено, що комбінація класичних методів машинного навчання з ознаками TF-IDF краща за інші методи на наборі даних LIAR, включаючи передові моделі машинного навчання. Метод BERTbase забезпечив точність на рівні з найкращими моделями. RoBERTabase показав найкращі результати на наборі даних PolitiFact з $F1$-оцінкою 93.17%. Метод SVM з ознаками TF-IDF продемонстрував результати кращі за результати моделей глибокого навчання на наборі даних GossipCop. На наборі даних COVID-19 найкращою моделлю був BERTbase. Однак експериментальні результати засвідчили, що жодна окрема техніка не змогла забезпечити найкращі показники продуктивності на всіх наборах даних.

Багато досліджень щодо виявлення фейкових новин в ОСМ залежать від однієї чи декількох ознак, таких як вміст, мережеве поширення або користувач [41-43]. Аналіз коментарів користувачів для визначення їхнього ставлення до новин може відігравати важливу роль у виявленні фейкових новин [44–46] та надавати уявлення про достовірність опублікованих новин [5, 34]. У роботі [47] стверджується, що коментарі користувачів мають велику дискримінантну цінність при виявленні фейкових новин, де вираження сентименту [38] або емоцій [48] має вирішальне значення. У праці [20] зазначено, що реакція користувачів на фейкові новини виражає емоції страху, огиди та здивування, тоді як на реальні новини - емоції очікування, суму, радості та довіри. Однак автори цієї праці не досліджували, наскільки добре емоції можуть ідентифікувати фейкові новини. Згідно [49], новизна може бути важливою складовою фейкових новин і значно підвищувати можливості їх поширення та прийняття в суспільстві. Більшість існуючих досліджень, які використовують аналіз сентименту, зосереджуються на сигналах сентименту вмісту фейкових новин [50]. Часто користувачі використовують емодзі замість текстових коментарів, щоб висловити свої думки про певні новини в ОСМ [51–53]. У цьому контексті техніки глибокого навчання (DL) суттєво сприяють класифікації, прогнозуванню та аналізу текстового контенту [54, 55]. Це пов'язано з їхньою здатністю до ефективного навчання [2, 12], виявлення ознак і складних патернів [56, 57].

У працях [3, 58] продемонстровано, що додавання ознак на основі аналізу настроїв та аналізу емоцій збільшує точність виявлення фейкових новин для більшості моделей глибокого навчання порівняно з використанням лише текстових ознак. Також авторами зазначених праць встановлено, що ознаки на основі аналізу настроїв новин та аналізу емоцій коментарів користувачів цих новин можуть бути використані соціальними медіа-платформами для боротьби з поширенням фейкових новин. Проте, застосування зазначеного підходу пов’язане з труднощами при роботі з незбалансованим набором даних.

Дослідження інших підходів виявлення фейкових новин дозволило зробити

висновок про те, що проаналізовані вище підходи є ефективними і перспективними в частині використання їх потенціалу для розробки нових моделей з високими показниками ефективності виявлення фейкових новин на різних наборах даних.

1.3 Функціональний аналіз базових алгоритмів виявлення в Інтернеті фейкових новин

Вважатимемо в подальшому в межах даної роботи, що базовими алгоритмами виявлення в Інтернеті фейкових новин є алгоритми машинного та глибокого навчання, а також алгоритми виявлення фейкових новин, що базуються на використанні результатів аналізу сентименту новинного контенту й емоцій у коментарях користувачів.

Актуальним завданням вбачається проведення функціонального аналізу зазначених алгоритмів з позиції використання їх потенціалу для опрацювання авторського алгоритму.

Авторами ряду наукових праць для виявлення фейкових новин було досліджено і випробувано велику кількість ML-алгоритмів. Зокрема:

Логістична регресія (LR): Логістична регресія - це статистична модель, яка застосовується як базова модель для широкого спектру завдань текстової класифікації.

Support Vector Machine (SVM): Класифікатор SVM - це потужний класифікатор, який є перспективним для вирішення низки завдань обробки природної мови.

Мультиноміальний наївний байєсівський (MNB): MNB - це популярний вид ймовірнісного алгоритму, який дає суттєві результати в різних завданнях обробки природної мови.

Дерево рішень (DT): DT - це алгоритм на основі дерева, в якому кінцеві вузли представляють високорівневі ознаки. Кожен вузол відображає вихід, а листок - клас

мітки. У вузлах приймається рішення на підставі навчання з нагляду, яке перетворює відображення ознак і значень у бажані результати.

Випадковий ліс (RF): RF – це алгоритм, що складається з набору дерев рішень, кожне з яких навчено на випадковому наборі ознак.

XGBoost (XGB): XGB - це комбінований алгоритм машинного навчання, в основі якого знаходиться градієнтний бустінг і який базується на деревах рішень. За допомогою бустінгу дерева будуються послідовно і кожне з них спрямоване на зменшення помилок попереднього.

Ensemble: Ensemble - це метод навчання, який об'єднує наведені вище алгоритми машинного навчання для покращення продуктивності.

Також було досліджено такі алгоритми:

CNN: CNN – це одновимірна згорткова нейронна мережа, що є потужним методом машинного навчання для автоматичного виділення ознак з текстових вхідних даних. CNN може автоматично видобувати локальні ознаки, вона менш обчислювально витратна, ніж інші алгоритми машинного навчання. Її архітектура включає один шар CNN з 128 фільтрами розміром ядра 5, які активуються функцією активації ReLU. Згенеровану карту ознак потім вдосконалюють і зменшують за допомогою шару максимального зведення, що призводить до отримання найбільш відповідної інформації. Після цього вихід розгортають і передають на вихідний шар з одиницею, яка активується сигмоїдною функцією активації.

LSTM: Модель LSTM [59] виправляє недоліки RNN, додаючи адитивну та мультиплікативну взаємодії до формули рекурентності та окремий стан пам'яті. Складність моделі можна збільшити, стекаючи LSTM-шари. За допомогою трьох гейтів - входу, забуття і виходу - моделі LSTM усувають проблеми ванішення та вибуху градієнта, які властиві RNN. Важливою характеристикою моделей LSTM є їхня здатність захоплювати віддалені залежності. У них використовується один шар BiLSTM з 128 одиницями для кодування вхідного тексту.

GRU: У варіанті GRU існують лише два гейти: гейт оновлення та гейт скидання. Гейт оновлення об'єднує гейти забуття та входу і вирішує, яка інформація

буде передана поточному стану. Гейти скидання визначають, коли ігнорувати попередній прихований стан [60].

Проста RNN розглядає контекст минулого, але не може враховувати контекст майбутнього. Тому для врахування майбутнього та попереднього контексту використовуються бідирекційні LSTM (BiLSTM) і бідирекційні GRU (BiGRU) завдяки їхнім конструкціям. Для забезпечення їх переваги об'єднуються шари прихованих обчислень вперед і назад, що контролює потік інформації в обох напрямках і призводить до кращого навчання.

Незважаючи на те, що BiLSTM та BiGRU виявляють переваги при вирішенні ряду проблем NLP, вони не позбавлені двох недоліків: зі збільшенням високорозмірного вхідного простору зростає їх складність, що призводить до ще більшої складності при їх оптимізації; ці моделі можуть захоплювати інформацію про послідовний і наступний контекст (концепція бідирекційності), вони не можуть фокусуватися на найважливіших частинах контекстуальної інформації тексту. Щоб усунути перший недолік, можна використовувати CNN для зменшення розмірності простору ознак, зберігаючи при цьому інформативні ознаки з тексту. Крім того, CNN може захоплювати та виділяти локальні патерни.

CNN-BiLSTM: Гібридизація моделей на основі рекурентних мереж із CNN допомагає виділяти суттєві ознаки, захоплювати локальні контекстуалізовані патерни та покращувати точність моделі. Спочатку використовується один шар CNN з 128 фільтрами та розміром ядра 5 для обробки вхідних векторів і виділення локальних ознак. Результати карт ознак шару CNN подаються на вхід одному шару BiLSTM з 128 одиницями, щоб вивчити довгострокові залежності локальних ознак новинних статей. Далі йде вихідний шар з однією одиницею, активованою сигмоїдною функцією. За допомогою RNN можна вивчити і захопити часові та контекстуальні ознаки та довгострокові залежності вхідного тексту, а важливі локальні ознаки можна виявити, використовуючи потужність CNN у роботі з просторовими відношеннями [61, 62].

CNN-BiGRU: Схожа до моделі CNN-BiLSTM. Відмінність полягає в тому, що архітектуру із шаром BiLSTM замінено шаром BiGRU.

Гібрид (Hybrid): Це гібридна модель, яка об'єднує три моделі: один шар CNN із 128 нейронами та розміром ядра 5, за яким слідує максимальне згортання, потім шар BiLSTM з 128 одиницями, а далі шар BiGRU з 128 одиницями.

Ще один підхід з виявлення фейкових новин базується на застосуванні трансформерів. Однак їх використання на даний час поки що обмежене.

Для вирішення досліджуваної проблеми дослідники пропонували різні методи інтерпретації значення слова за допомогою векторів вбудовування. Методи на основі нейронних мереж, такі як Word2Vec і GloVe, загалом використовувалися для навчання векторів вбудовування слова з великих корпусів слів. Однак ці моделі вбудовування мають недолік у тому, що вони не враховують контекст і статичні вбудовування слів генеруються незалежно від їх контексту. Тому для досягнення більш точних результатів модель повинна бути здатною вловлювати семантичні та контекстуальні патерни. Більше того, вдосконалені моделі машинного навчання можуть автоматично видобувати семантичну інформацію з вхідних даних для виявлення фейкового контенту, але вони не можуть точно розпізнавати фейковий контент без глибокого розуміння тексту. Таким чином, в останні роки зросло зацікавлення до парадигми уваги. У галузі обробки природної мови спостерігається загальний зсув парадигми, спрямований на розробку набору моделей, які не лише підвищують точність, але й вирішують проблему відсутності маркованих даних.

Крім того, існує нагальна потреба в автоматичному виявленні фейкових новин. Це складна задача, оскільки існуючі моделі машинного навчання (до появи моделей на основі трансформерів) не забезпечували глибокого семантичного розуміння текстових вхідних даних. Це призвело до впровадження попередньо навчених мовних моделей на основі трансформерів. Використання таких моделей, навчених на великих немаркованих даних, для завдань класифікації тексту стає все більш популярним. Для адаптації до завдання наступного рівня над новими шарами нейронних мереж розташовуються попередньо навчені шари у мовних моделях [63]. У цьому випадку

на верхньому рівні попередньо навчених мовних моделей (PLMs) додається повністю з'єднаний (FC) шар для цілей класифікації.

Ключовими моделями з потужним мовним представленням, навченим на величезних обсягах текстового корпусу, є моделі BERT [64] і RoBERTa [65].

BERT: BERT – модель, що використовує багаторівневий бідирекціональний трансформерний кодер, який одночасно моделює лівий і правий контексти [64]. У результаті BERT генерує вектори, які враховують контекст. BERT у подальшому надає можливість подолати однобічне обмеження, проводячи передпочаткове навчання з використанням невдосконаленого завдання передбачення, яке включає в себе маскову мовну модель (Masked Language Model, MLM), яка розуміє контекст і передбачає слова. Таким чином, модель може генерувати векторне представлення, яке захоплює загальну інформацію вихідного тексту. Семантичне представлення кожного слова у вхідному тексті можна покращити за допомогою механізму уваги, підсилюючи семантичне представлення на основі контексту слова. Механізм уваги відіграє важливу роль в архітектурі трансформера, оскільки надає різні ваги різним частинам тексту в залежності від їхнього внеску в результат. Функція уваги відображає запити та слідує за парами ключ-значення та векторами-результатами. Механізм уваги використовує функцію активації Softmax, яка нормалізує вхідні дані до значення між 0 і 1.

Незважаючи на те, що BERT містить мільйони параметрів (наприклад, BERTbase має 110 мільйонів параметрів, а BERTlarge - 340 мільйонів параметрів) [64], BERT є відносно ефективним для застосування у завданнях наступного рівня з використанням спільно налаштованих параметрів на основі попередньо навченої моделі.

RoBERTa [65]: RoBERTa - оптимізована версія підходу BERT. У цьому методі BERT перенавчається за допомогою: вдосконаленої методології навчання, включаючи видалення завдання передбачення наступного речення з попереднього навчання; використання у 10 разів більшої кількості даних, ніж BERT; введення динамічного маскування з більшими пакетами даних, так що токени, які маскуються,

змінюються під час навчання, на відміну від статичного маскування, використовуваного в BERT. Таким чином, RoBERTa відрізняється від BERT в способі підходу до попереднього навчання.

Аналіз функціонального аспекту застосування наведених методів передбачає проведення аналізу попередньої обробки вхідного тексту. Попередня обробка вхідного тексту включала в себе видалення стоп-слів і пунктуації. Текст проходив процес токенізації на токени і побудови словника на основі вивченої лексики в даному текстовому корпусі. Такий словник використовувався для відображення кожного токена в єдиний цілочисельний ідентифікатор. Отримані послідовності доповнювалися або обрізалися до фіксованої кількості записів, оскільки моделям необхідно було подавати вектори однакової довжини. При цьому недоліком є те, що при обрізанні послідовностей втрачалася деяка (корисна) інформація. Ці послідовності потім перетворювалися в фіксовані вектори вбудовування слова. Потім нейронні мережі ініціалізувалися контекстно-незалежними попередньо навченими моделями вбудовування слова, такими як Word2Vec і GloVe, та контекстно-інформованими попередньо навченими мовними моделями BERTbase. У класичних алгоритмах машинного навчання для перетворення токенізованих текстів в ознаки використовувалися CountVectorizer (CV) та TF-IDF, як статистичні здобувачі ознак для моделей машинного навчання.

Також для аналізу функціонального аспекту застосування наведених методів доцільно оцінити результати застосування описаних вище моделей при виявленні фейкових новин на різних масивах даних.

Аналіз результатів на наборі даних LIAR.

Як випливає з [40], найкращий результат точності, досягнутий на наборі даних LIAR, становив 63,9% для комбінації з ознаками TF-IDF. Типовий результат при цьому становить 62%. Найгірший результат склав 51,78% для DT з ембедінгами GloVe. Класифікатор не міг виявити деякі закономірності у вхідному тексті.

При проведенні порівняльної оцінки результатів різних моделей встановлено, що більшість моделей досягли точності не більше 62% з BERTbase, а модель CNN-

BiLSTM з використанням ембедінгів BERTbase точності 63,06%. Найгірший результат було досягнуто для моделей на основі рекурентних мереж з ембедінгами Word2Vec.

Оскільки набір даних LIAR містить короткі політичні заяви, складно отримати корисні вказівки, які могли б допомогти розрізняти фейкові новини від реальних новин, зокрема при використанні складних моделей, таких як LSTM, оскільки це може збільшити ризик перенавчання.

Як зазначено в роботі [66], інформація новинної статті та розмір набору даних - це важливі фактори, що впливають на продуктивність моделей на основі рекурентних мереж. Такі моделі більш схильні долати перенавчання, коли в новинній статті надається достатньо інформації. Отримані результати показують, що класичні методи машинного навчання випереджають високо вдосконалені методи глибокого навчання на наборі даних LIAR, включаючи найновіші передові мовні моделі, такі як BERTbase та RoBERTabase.

Аналіз результатів на наборі даних FakeNewsNet.

Результати, які отримані на наборах даних PolitiFact і GossipCop, свідчать про те, що надійність має постійно високу точність, випереджаючи базові SAF та CNN навіть з комбінаціями соціальних вказівок та контенту новин на обох наборах даних. Найбільше значення точності (89,93%) на наборі даних PolitiFact було досягнуто за допомогою комбінації із CV як методу вилучення ознак. Ця модель (Ensemble+CV) виявилася другою за результатами на наборі даних GossipCop, з SVM+TF-IDF ($F1$-оцінка 91,55%) з числа інших моделей на наборі даних GossipCop. Методи глибокого навчання, зокрема найновіші передові мовні моделі, забезпечили високу продуктивність у багатьох завданнях обробки природної мови. RoBERTabase показав найкращий результат на наборі даних PolitiFact з $F1$-оцінкою 93,17%, тоді як результати точності (92%) були на рівні з передовою моделлю машинного навчання за допомогою представлень BERTbase. Для набору даних GossipCop CNN-BERTbase виявився найкращою моделлю (значення $F1$ 91,45%). Це свідчить про те, що контекстно-орієнтовані моделі на основі трансформатора допомагають виявляти

корисні закономірності для розрізнення фейкових новин від реальних. Оскільки набір даних GossipCop досить незбалансований, різні методи збалансування, такі як вибіркове збільшення та зменшення, можуть допомогти збалансувати набір даних і підвищити продуктивність виявлення.

Аналіз результатів на наборі даних COVID-19.

BERT ефективно може виявляти фейковий контент, оскільки він має можливість кодувати глибоку семантичну контекстуальну інформацію. За результатами досліджень можна зробити висновок, що модель BERTbase, до якої додано лінійний шар для класифікації, показала найкращі результати порівняно з аналогами та іншими класичними моделями машинного та глибокого навчання на наборі даних COVID-19. Другою за ефективністю на наборі даних COVID-19 була модель CNN-BERTbase.

У зв'язку з тим, що методи векторних просторів, такі як CV та TF-IDF, не враховують контекст, використання цих представлень з моделями машинного навчання ґрунтується на зовнішньому вигляді токенів при прийнятті остаточних рішень, незалежно від їхнього контексту. Результати дослідження свідчать про те, що моделі векторного простору неефективні у виявленні глибокої семантики та контекстуальних закономірностей, що містяться в користувацькому змісті, створеному на Twitter. Однією з головних переваг BERT (та його модифікацій) у випадку Twitter (де користувацький зміст часто містить орфографічні помилки, шум та скорочення), є використання підтокенів замість фіксованих токенів. Це ідеально підходить для використання з такими даними [67], оскільки воно оперує на рівні підтокенів замість стандартних контекст-незалежних векторних вбудов слів на рівні слова.

Оцінимо тепер функціональний аспект алгоритмів виявлення фейкових новин, що базуються на використанні результатів аналізу сентименту новинного контенту й емоцій у коментарях користувачів.

Насамперед відмітимо, що перевірку моделей, що відповідають зазначеним алгоритмам, як правило, здійснюють на наборі даних Fakeddit.

Набір даних Fakeddit (https://github.com/entitize/Fakeddit, доступний на 22 лютого 2022 року) є великомасштабним та багатомодальним набором даних (текст та зображення), зібраним з соціальної медіа-платформи Reddit за період з 19 березня 2008 року по 24 жовтня 2019 року [68]. Цей набір даних складається з понад мільйона повідомлень з різних галузей. До цих повідомлень додаються різні ознаки, такі як зображення, коментарі, користувачі, домени та інші метадані. У цьому наборі даних міститься багато шуму та нульових значень. Одне повідомлення може містити кілька коментарів або не мати коментарів. Для кожного повідомлення дослідники надали три мітки, класифіковані двома, трьома та шістьома способами.

У роботі [69] відзначається, що існує зв'язок між настроєм опублікованих новин та правдивістю новин. На основі цього автори роботи використовували ознаку, засновану на настрої (відношення кількості позитивних слів до кількості негативних слів), для розвитку власної моделі виявлення фейкових новин. Для покращення точності визначення фейкових новин у [70] запропоновано нову стратегію, яка включає збагачення об'єднаного набору даних настроєм як ключовою ознакою. Ефективність запропонованої стратегії перевірялася на основі трьох різних наборів даних. Результати показали ефективність запропонованого рішення. У роботі [50] підтверджено, що різницю між фейковими та справжніми новинами можна відрізнити за допомогою подвійної емоції (емоція видавця та суспільна емоція). Автори цієї роботи запропонували «Ознаки подвійної емоції», щоб виразити подвійну емоцію та взаємодію між ними. Вони також показали, що запропоновані ними ознаки легко інтегруються в існуючі моделі виявлення фейкових новин.

У роботі [71] запропоновано стратегію EmoCred, модель на основі довгострокової пам'яті (LSTM), яка враховує емоційні сигнали для виявлення різниці між справжніми та фейковими твердженнями. Головним кроком EmoCred є вилучення емоційних сигналів з тверджень. Дослідники вивчали три різні методи визначення емоційних сигналів у твердженнях: метод, який використовує сучасні лексикони емоцій та базується на лексиконі (emoLexi); метод, який визначає інтенсивність емоцій у твердженнях (emoInt); метод нейронної мережі (NN), який

вказує на рівень інтенсивності тверджень і представляє кількість емоційних реакцій, які це твердження може викликати у читачів (emoReact). Дослідження авторів зазначеної праці, в якому використані реальні набори даних, продемонструвало цінність емоційних сигналів при оцінці вірогідності.

У роботі [72] проведено класифікацію новин на реальні та фальшиві на основі аналізу емоцій у новинах за допомогою комбінації моделей, що включають у себе згорткові нейронні мережі (CNN) і мережі BI-LSTM з механізмом уваги. Результати застосування моделі виявилися кращими, ніж результати застосування інших моделей.

Нижче наведені результати застосування описаних вище моделей при виявленні фейкових новин на різних масивах даних.

У роботі [68] наведено багатомодальний набір даних Fakeddit, застосовано модель BERT для виявлення фейкових новин на основі тексту і техніки ResNet для виявлення фейкових новин на основі зображень. Дослідники виявили, що точність результатів моделі, побудованої на основі BERT і ResNet з використанням ознак як тексту, так і зображень, була кращою, ніж у моделі, яка використовує лише текст для виявлення фейкових новин.

У роботі [73] запропоновано модель виявлення фейкових новин, яка ґрунтується на DeepNet та реальних даних наборів даних BuzzFeed і Fakeddit. З використанням факторизації тензорів, яка інтегрує контент новин та дані, що базуються на соціальному контексті, DeepNet демонстрував кращі параметри, ніж поточні моделі виявлення фейкових новин. Результати показали, що використання комбінації ознак на основі соціального контексту і ознак контенту новин призводило до більш точних результатів DeepNet.

У роботі [74] також використано набір даних Fakeddit і представлено багатомодальну мережеву архітектуру, яка дозволяє об'єднувати різні рівні та види об'єднання інформації, включаючи не тільки текст заголовків, але і метадані та інший контент, що стосується заголовків новин. Їхня методологія для виявлення неправдивої інформації залежить від чотирьох модальностей вводу: основний

текстовий контент новини або запису; додаткова інформація або реакція на основний контент (наприклад, коментарі); візуальний контент запису; будь-які інші доступні метадані. Для врахування унікальності фундаментальної структури модальностей, автори роботи вони об'єднали інформацію на різних рівнях. Результати запропонованої ними моделі були покращені завдяки додатковим модальностям, що свідчить про те, що вони надають цінну інформацію.

Унікальна мережа вилучення та мережа мислення, відома як SERN, а також механізм визначення уваги на основі речення, були наведені в роботі [75]. Автори роботи пояснили, що відповіді, надіслані різними читачами, містять як факти, так і деякі висновки. Вони використали графічну мережу мислення. Дослідники у своїй роботі використали набори даних Fakeddit і PHEME та вказали, що запропонована ними модель продемонструвала перспективні результати порівняно з іншими сучасними моделями.

У роботі [76] розроблено підхід, який використовує ознаки контенту новин та ознаки соціального контексту для виявлення фейкових новин. Підхід авторів цієї праці базується на архітектурі трансформатора, яка складається з двох блоків: блоку кодування для вилучення значущих представлень з даних фейкових новин та блоку декодування для передбачення майбутньої поведінки на основі минулих даних. Вони провели обширні випробування на даних Fakeddit та NELA-GT-2019, щоб оцінити ефективність запропонованого ними підходу. Вони використовували стратегію недорогого семплування, яка включає видалення записів з більшості класу з метою наближення більшості класу до меншості класу, щоб вирішити проблему нерівноваги даних в обох наборах даних. Автори роботи [76] використовували ознаки соціального контексту набору даних Fakeddit. За результатами застосування їхньої моделі можна зробити висновок, що для оптимальної роботи моделі потрібно враховувати як контент новин, так і соціальний контекст.

За результатами порівняльного оцінювання наведених моделей встановлено, що найефективніша модель виявлення, яка використовується на валідаційному наборі

даних, це Bi-LSTM зі значенням AUC 96,77% та 97,81% за $F1$-оцінкою. Проте, навіть для неї мають місце труднощі при роботі з незбалансованим набором даних.

Таким чином, проведений функціональний аналіз алгоритмів машинного (ML) та глибокого (DL) навчання, а також алгоритмів виявлення фейкових новин, що базуються на використанні результатів аналізу сентименту новинного контенту й емоцій у коментарях користувачів, дозволяє зробити такі висновки: на даний час жодна з проаналізованих існуючих моделей виявлення фейкових новин не є універсальною для різних наборів даних; різні моделі та їх комбінації мають як переваги, так і недоліки, які не дозволяють ефективно застосовувати моделі.

1.4 Постановка задачі

Метою кваліфікаційної роботи магістра є розробка методу виявлення в Інтернеті фейкових новин нейромережевими засобами. Тому для виконання поставленого завдання потрібно:

– провести аналіз існуючих моделей і підходів для виявлення в Інтернеті фейкових новин;

– розробити метод виявлення в Інтернеті фейкових новин нейромережевими засобами;

– підготувати набір даних для навчання нейронної мережі;

– провести навчання нейронної мережі виявляти фейкові новини;

– розробити інформаційну систему, що реалізує запропонований метод;

– визначити якість запропонованого методу виявлення в Інтернеті фейкових новин за відомими статистичними показниками.

Висновки до розділу 1

У ході опрацювання першого розділу було обгрунтовано актуальність роботи, сформовано мету роботи, визначено завдання дослідження. Проведено аналіз

предметної області та існуючих публікацій щодо виявлення в Інтернеті фейкових новин. Також здійснено функціональний аналіз базових алгоритмів виявлення в Інтернеті фейкових новин, до числа яких віднесено алгоритми машинного та глибокого навчання, а також алгоритми виявлення фейкових новин, що базуються на використанні результатів аналізу сентименту новинного контенту й емоцій у коментарях користувачів. Здійснено постановку завдання та часткових задач, які потребують вирішення в даній роботі.

РОЗДІЛ 2
Метод виявлення фейкових новин для оцінки достовірності джерел масової інформації

2.1 Фейкові новини, їх класифікація та методи виявлення на основі використання нейромережевих технологій

2.1.1 Основні положення про фейкові новини

Одним із важливих завдань при розробці методу виявлення фейкових новин на основі використання нейромережевих технологій є аналіз того, що собою являють фейкові новини, якими вони бувають і які методи та прийоми боротьби з ними на даний час є ефективними.

Слід зазначити, що поняттю фейкових новин приділена увага різних категорій дослідників. Значною мірою фейкові новини, фейк-ньюз(-с) (з англ. fake news - підроблені/імітаційні новини) інтерпретуються як підроблення або імітація новин (маніпулятивне викривлення фактів; дезінформація), які створено без урахування редакційних норм, правил, технологій, застосовуваних у ЗМІ для забезпечення відповідності й перевіреності, та яке не витримує ніяких, навіть поверхневих, перевірок на відповідність і об’єктивність, але, разом з тим, має суттєвий вплив на свідомість значної кількості людей.

Фейк-ньюс та їх поширення в ОСР мають параметри та ознаки, які деякою мірою можна вважати феноменальними, а саме:

- швидкість поширення фейк-ньюс в середньому вшестеро більша, ніж швидкість поширення реальних новин;
- ймовірність репосту фейк-ньюс на 70 % більша ніж репосту реальних новин;
- уявна «глибина поширення» – окремі фейк-ньюс поширюються в десять разів швидше, ніж реальні новини, й можуть мати довжину ланцюжка в 19 перепостів (у реальних новин відповідне значення майже ніколи не більше 10 репостів).

Базові ознаки:

- примітивізм - фейкові новини, як правило, створюються для людей, які через різні причини, не перевірятимуть достовірність отриманої інформації;
- емоційне подання інформації - акцент на емоційне (як правило негативне) сприйняття інформації, яке не сприяє критичному аналізу;
- як правило, не мають продовження - фейкові новини як правило не мають продовження, вони розраховані на оперативне маніпулювання громадською думкою в короткостроковій перспективі.

Причинами масового поширення фейкових новин є таке:

- сучасні люди характеризуються кліповим сприйняттям. Вони все сприймають швидко і так само швидко до всього втрачають інтерес;
- аналізуючи матеріал соціальних мереж, людині потрібно менше однієї секунди, щоб усвідомити, чи цікавий їй цей матеріал;
- 96% людей сприймають інформацію, аналізуючи лише заголовки статей;
- згідно матеріалів однієї із статей Масачусетського технологічного інституту, яка була опублікована в журналі Science, неправдива інформація в Твіттері поширюється *вшестеро скоріше за правду*.

На даний час існують окремі правила, що дозволяють розпізнавати фейкові новини. Ці правила можуть бути сформульовані так:

- переконайтеся, що веб-сайт, інформація з якого аналізується, є надійним. Наприклад, чи представлені в ньому контактна або адресна інформація;
- здійсніть перевірку автора та його попередніх публікацій. Якщо сенсаційна новина від автора є його першою новиною, це може вас спонукати до роздумів;
- необхідно читати всю статтю, а не тільки її заголовок. Невиконання цього правила може заплутати та призвести до певних непорозумінь;
- слід переконатися, що інформаційне повідомлення не є якимось жартом;
- слід перевірити інформацію у різних джерелах;
- слід врахувати думку експертів щодо правдивості інформації;

- слід переконатися, що стаття є новою, а не, наприклад, вийшла певний проміжок часу назад. Якщо вона не свіжа, то для фіксованого моменту часу це може мати зовсім інший зміст та контекст;
- слід оцінити, чи погляди або переконання автора не впливають на сприйняття інформації і чи не сприймається вами матеріал упереджено.

Фейками можуть називатися:

- підроблені тексти, аудіо, фото чи відеозаписи;
- сторінки чи блоги, які ведуться від чийогось імені у соціальних мережах (наприклад, від імені реальних чи вигаданих людей, історичних персонажів, художніх персонажів);
- підроблені сторінки відомих або популярних сайтів (при аналізі цих сторінок людина помилково вважає, що одержує інформацію з якісного ресурсу, якому можна довіряти; такі сторінки часто створюються зловмисниками для того, щоб заволодіти увагою користувача та спонукати його поділитися на фальшивій сторінці своїми справжнім логіном або паролем з метою їх подальшого використання в своїх інтересах);
- штучно створена на чиєсь замовлення популярність особистості, чи твору, чи деякого проекту (як правило, з використанням інтернет-ботів і (чи) тих же фальшивих акаунтів, за допомогою яких виставляються «лайки» і постяться схвальні коментарі);
- повністю невірна чи частково спотворена інформація про факти, події або явища.

Проведений аналіз ряду інформаційних джерел дозволяє зробити такі висновки. Незважаючи на те, що на даний час фейкові новини є досить поширеним явищем, комплексні дослідження цього феномена поки що відсутні. Відсутні чіткі критерії визначення поняття «фейкові новини». Більше того, у різних дослідженнях аналізоване явище детермінується різними поняттями, які відрізняються за обсягом і змістом. Наприклад, фейкові новини, фальшиві новини, псевдоновини, інформаційні вкидання, медіафейк, медіамістифікація. У сучасному медіапросторі фейком може

бути не лише фальшива новина, а й підроблена фотографія, сторінки у соціальних мережах, які створені від імені іншої людини, а також все, що не відповідає реальним фактам.

Зважаючи на це, у межах даної роботи, з урахуванням її мети та завдань, під фейковою новиною будемо розуміти опубліковані новинні статті, які містять неправдиву інформацію, з метою свідомого введення людей в оману чи здійснення якихось зловживань.

2.1.2. Класифікація фейкових новин

Ще одним актуальним завданням є дослідження можливих підходів до класифікації фейкових новин. Проведений аналіз різних джерел вказує на відсутність єдиної класифікації.

На даний час використовуються такі класифікації.

І. Класифікація фейкових новин за М. Кіцою. В основу класифікації покладено критерії, що можуть слугувати таксономії інформаційних повідомлень, які аналізуються читачами.

Такими фейковими новинами можуть бути:

- за формою презентації - текст, аудіо, фото, відеозапис;
- за змістом - агітація, маніпуляція, пропаганда тощо;
- за тематикою - політичні, соціальні, світські новини тощо;
- за призначенням для певної вікової групи - для молоді, для дорослих людей, для людей пенсійного віку тощо;
- за джерелом інформації - без джерела, невідоме джерело, від першого джерела.

ІІ. Класифікація фейкових новин за Д. Лавникевичем. В основу класифікації покладено критерій, що характеризує мету створення фейкових новин.

Такими фейковими новинами можуть бути:

- фейки, що створені випадково;
- фейки, що створені для ведення інформаційної війни;

- фейки, що створені з комерційною ціллю;
- фейки, що створені для залучення додаткового трафіку;
- фейки, що створені з неявною метою.

III. Класифікація фейкових новин за К. Уордл (директор проекту First Draft News). В основу класифікації покладено критерії, що характеризують різні типи створюваного контенту, мотиви людей, які створюють контент, засоби поширення контенту.

Вказані критерії дозволяють виокремити сім видів фейкових новин:

- розважальні - пародія чи сатира;
- хибний зв'язок - коли заголовки, зображення або написи на екрані не відповідають контенту;
- оманливий контент - вживання такої інформації, яка забезпечує дискредитацію певної особи чи якоїсь справи (дії);
- неправдива ситуація – наповнення реального контенту неправдивою контекстуальною інформацією;
- контент «самозванця» - ситуація, коли дійсні, правдиві джерела інформації видають себе за фейкові джерела інформації;
- маніпуляційний контент – ситуація, коли інформація (реальна чи вигадана) використовується для маніпуляції або обману. Як правило, інформація підкріплюється фейковими фотографіями;
- сфабрикований контент – новий хибний контент, який створений для введення в оману або завдання шкоди.

IV. Класифікація фейкових новин у залежності від співвідношення часток достовірної і недостовірної інформації:

1. Інформаційне повідомлення є цілковитою брехнею.

2. Інформаційне повідомлення на тлі загалом достовірної інформації частково містить і недостовірну інформацію.

3. Інформаційне повідомлення стосується реальної події, окремі аспекти якої спотворені. Це можуть бути, зокрема, змінені у потрібному для фальсифікаторів

напрямі аудіозаписи чи відеозаписи, відредаговані фотографії, цитати, що вирвані з контексту певних інформаційних джерел або які представлені в певній послідовності тощо.

V. Класифікація фейкових новин у залежності від достовірності обставин, часу або місця події:

1. Інформаційний фейк видає за новину правдиву інформацію, яка мала місце в минулому.

2. Новина про подію, яка насправді мала місце в одному місці, подається як така, що трапилася в іншому місці.

VI. Класифікація фейкових новин у залежності від складу осіб, що згадуються в новині:

1. Новина містить посилання на висловлювання відомої особи, яке нібито мало місце, що розміщене від імені фейкового акаунта.

2. Новина виставляє як головну дійову особу другорядного учасника події.

3. Новина, що базується на неперевірених показах осіб, які нібито були свідками певних подій.

VII. Класифікація фейкових новин у залежності від мети їх створення та поширення:

1. Інформаційні фейки, що створюються та поширюються з метою розваги споживачів.

2. Новини, що створюються та поширюються для досягнення політичних переваг: дискредитації політичних конкурентів (у тому числі під час виборчої кампанії), провокації протестів чи насильницької зміни влади тощо.

3. Інформаційні фейки, що створюються для дискримінації осіб за ознаками статі, раси, національності, мови, віросповідання, походження, майнового чи посадового становища, місця проживання, особистих переконань, належності до громадських об'єднань тощо.

4. Інформаційні фейки, що створюються та поширюються для підвищення інтернет-трафіку.

5. Інформаційні фейки, що створюються та поширюються з метою шахрайського заволодіння грошовими активами або майном.

6. Інформаційні фейки, що створюються з метою заподіяння шкоди інформації, яка зберігається в комп'ютері користувача.

7. Інформаційні фейки, що створюються та поширюються з метою привернення уваги до певної особи, компанії чи фірми, проекту тощо.

8. Інформаційні фейки, що створюються та поширюються з метою маніпуляції ринком чи отримання певних переваг економічного характеру.

VIII. Класифікація фейкових новин у залежності від рівня сприйняття достовірності:

1. Інформаційні фейки, що мають явно фейковий характер.

2. Інформаційні фейки, що здатні викликати сумнів щодо їхньої фейковості та спонукати користувачів перевірити правдивість інформації.

3. Інформаційні фейки, що створені настільки переконливо, що сумнівів у їхній неправдивості практично ні у кого не виникає.

Проведений аналіз ряду інформаційних джерел дозволяє зробити такі висновки.

На сьогодні немає єдиної класифікації фейкових новин.

Варіант класифікації з числа існуючих необхідно обирати в залежності від цільової настанови дослідження, що проводиться.

З урахуванням мети даного дослідження у межах роботи аналізуватимуться всі наведені вище класифікації.

2.1.3. Методи виявлення фейкових новин

Питанню аналізу існуючих методів виявлення фейкових новин присвячено значну кількість спеціалізованих наукових робіт. Окремі з них описані в розділі 1. У межах даного підрозділу важливо оцінити систематизовані підходи, функціонал яких важливий з точки зору його можливого використання в методі, що опрацьовуватиметься в даній роботі.

На даний час до числа зазначених підходів можна віднести такі.

Підхід 1. Виявлення як елементу дезінформації спаму.

Для виявлення спаму використовуються статистичні методи машинного навчання. Ці методи дозволяють класифікувати текст як спам чи не спам. Зазначені методи передбачають попереднє опрацювання та обробку тексту, встановлення ознак (bag of words), які забезпечують кращу точність на тестовому наборі даних, та вилучення непотрібних ознак. Після визначення ознак текст можна класифікувати на основі застосування Naive Bayes, Support Machines, TF-IDF або K-найближчих сусідів.

Підхід 2. Синтаксичний аналіз.

Часто для встановлення фейковості новин аналізу слів недостатньо. Для вирішення цього завдання можна застосовувати інші мовні підходи, зокрема такі, як аналіз синтаксису і граматики мови. Для перетворення речень у дерева, що характеризують їх структуру, та для представлення речень у вигляді рекурсивної синтаксичної структури, використовуються Probability Context-Free Grammars (PCFG). Під час синтаксичного аналізу досліджуваний текст представляється у вигляді структури даних, зазвичай у вигляді дерева, яке відповідає синтаксичній структурі тексту, як вхідної послідовності, і яке добре підходить для його подальшої обробки.

Підхід 3. Семантичний аналіз.

Як правило, в більшості випадків правдивість інформаційного повідомлення чи тексту можна передбачити шляхом дослідження коментарів або подібних статей. У випадку, коли більшість схожих статей не відповідає досліджуваній новині, найімовірніше, що вона може бути фальшивою чи упередженою. Аналогічно коментарі до статті можна використовувати для встановлення того, чи є факти, які наведені в статті, достовірними. Основними недоліками такого підходу є складність автоматичного пошуку схожих статей, перевірки відповідності профілю та обліку різних слів, які характеризують одне і те ж саме.

Підхід 4. Мережеві методи.

В Інтернеті є велика кількість метаданих, які можна використовувати для прогнозування надійності інформаційного джерела. Мережеві методи базуються на застосуванні цих даних. Можна виокремити два підходи: аналіз метаданих; перевірка пов’язаних даних і фактів.

Метадані. Мережевий підхід вивчає метадані. Зокрема, можуть аналізуватися URL-адреси, автори, вподобання у соціальних мережах тощо. На основі цього формується висновок про надійність джерела.

Перевірка пов'язаних даних і фактів. Правдивість інформаційних повідомлень також можна встановити, перевіривши факти, які згадуються в них. Одним із підходів для перевірки фактів є генерування дерева відношень. Хоч і перевірка фактів має ряд переваг стосовно правильної класифікації фальшивих новин, сам процес перевірки складно автоматизувати. Це призводить до низької швидкодії зазначеного методу.

Підхід 5. Застосування онлайн-інструментів.

Для відслідкування траєкторії поширення фейкових новин у соціальній мережі Facebook можна використовувати онлайн-інструменти CrowdTangle, Google News Search та Gephi.

Задача дослідження траєкторії поширення фейкової новини розв’язується таким чином.

1) Насамперед необхідно встановити типи користувачів, які залучають фейкові новини у Facebook.

2) Необхідно відстежити поширення контенту в мережі та оцінити ефективність залучення аудиторії на Facebook, Twitter, YouTube, Instagram.

3) Необхідно перевірити охопленість аудиторії поширювачів фейкових новин у Facebook фактчекерами.

4) Необхідно здійснити пошук найбільш популярних джерел фейкової інформації.

Підхід 6. Перевірка фактів.

Вивчаючи інформаційні повідомлення з точки зору знань, необхідно намагатися проаналізувати та встановити неправдиву інформацію, використовуючи

процес, який відомий як перевірка фактів. Перевірка фактів, що спочатку застосовувалася в журналістиці, забезпечує можливість оцінки справжності новин на основі порівняння знань, що можуть бути отримані з контенту новин. При цьому вони позиціонуються як відомі факти.

Підхід 7. Ручна перевірка фактів.

Ручну перевірку фактів можна класифікувати як експертну та краусорсингову. Експертна перевірка фактів базується на використанні для перевірки необхідного змісту контенту експертів з певного середовища. Як правило, вона проводиться для перевірки невеликої групи даних, зокрема, якогось конкретного факту. Цей підхід є легким у використанні і забезпечує високу точність. Однак він є дорогим і погано масштабованим у випадку збільшення кількості фактів, які необхідно перевірити. Краусорсингова перевірка фактів концентрує в собі перевірку фактів на багатьох середовищах. Порівняно з експертною перевіркою фактів, краусорсингова перевірка є менш налаштовуваною, забезпечує меншу достовірність і точність. Для отриманої інформації необхідним є фільтрування невідомих джерел і прийняття рішень у конфліктних ситуаціях.

Підхід 8. Автоматична перевірка фактів.

Метод ручної перевірки фактів має обмежені можливості, оскільки не встигає за кількістю нової інформації, що з’являється, особливо в соціальних медіа. Тому для вирішення досліджуваної задачі розроблені автоматичні методи перевірки фактів. Вони базуються на пошукові інформації (IR) та обробці природної мови (NLP). Загальний процес автоматичної перевірки фактів можна розбити на два етапи: виокремлення фактів (або побудова бази знань); перевірка фактів (або порівняння знань із бази знань). Отримання інформації в основному здійснюється з мережі Інтернет. Це середовище надає масив неструктурованої інформації у вигляді онлайн-документів. Загалом, розглядається чотири типи вебконтенту: текст; табличні дані; структуровані сторінки; анотації. Цей контент містить реляційну інформацію та може використовуватись для отримання фактів різними парсерами. Для того, щоб оцінити справжність інформаційних повідомлень, необхідно додатково порівняти отримані

факти, які потребують підтвердження, з тими фактами, які зберігаються у створеній або вже існуючій базі знань, тобто із справжньою інформацією. Для цього можна використовувати лінгвістичні методи.

Сучасні автоматизовані системи аналізу інформації, як правило, працюють за одним із трьох принципів.

1. Аналіз не змісту, а стилю тексту.

2. Аналіз поширення.

3. Аналіз активності користувача.

Для класифікації інформаційних фейків на основі аналізу стилю тексту часто застосовуються такі алгоритми машинного аналізу.

Метод опорних векторів. Цей метод дозволяє суттєво зменшити потребу в розмічених навчальних примірниках при різних типах навчання. Класифікатори цього типу забезпечують ефективність і точність.

Метод k-найближчих сусідів. Цей метод реалізує алгоритм аналізу на основі подібності. Його можна застосовувати для різної класифікації текстів. За ефективністю він співставний із методом опорних векторів.

Логістична регресія. Цей метод є одним із ключових аналітичних інструментів, які застосовуються в обробці природної мови з метою контрольованого навчання класифікаторів, які приймають рішення на основі порівняння вхідних даних та опорних даних.

«Випадковий ліс». Метод класифікації, який передбачає вирощування великого набору вирішальних дерев і вибору тих з них, що забезпечують найкращі результати.

«Наївна байєсовська модель». Цей інструмент застосовується для створення ймовірнісних класифікаторів фейкових новин. Він демонструє достатню ефективність за наявності складних реалістичних умов.

Штучні нейронні мережі. Моделі, які складаються з вузлів, що реалізують обчислення: штучні нейрони поєднують вхідні дані з ваговими коефіцієнтами. При

цьому вони призначають отриманим відомостям вагу з урахуванням завдання, яке виконується мережею.

Глибинне навчання. Нейронні мережі, які схожі на звичайні та які складаються з великої кількості шарів. Для розпізнавання інформаційних фейків використовуються різноманітні архітектури таких мереж, у тому числі згорткові мережі (convolutional neuronal network, CNN), які потребують мінімальної кількості числа параметрів, попередньої обробки та об’єму роботи. Крім цього, використовуються рекурентні нейромережі, які змінюються з часом динамічно і які здатні запам'ятовувати нову лексико-семантичну інформацію. Також використовуються ієрархічні мережі уваги (hierarchical attention network, HAN), які встановлюють певні характеристики структури документа.

Представлені методи можуть демонструвати непогані результати. Разом з тим, у більшості випадків (як при розпізнаванні фейкових новин, так і в інших задачах класифікації) кращу ефективність мають такі методи, як логістична регресія, «наївна байєсовська модель» та глибинне навчання. Ці методи добре характеризуються на різних наборах даних, а також при роботі з різними схемами обробки.

Наведений аналіз методів виявлення фейкових новин важливий з точки зору визначення авторського підходу до формування методу виявлення фейкових новин в Інтернеті на основі використання нейромережевих засобів.

2.2 Удосконалення структури нейромережі для виявлення фейкових новин

В основу авторського методу виявлення фейкових новин покладаються нейромережеві засоби, а саме багатошарова CNN нейромережа.

CNN нейромережа (згорткова нейронна мережа) – це один із основних інструментів класифікації та розпізнавання об’єктів, текстів, мови тощо. Дана нейромережа за рахунок спеціальної операції, безпосередньо згортки, дозволяє

виділити опорні ознаки зображень, текстів, використовуючи які можна робити певні висновки чи класифікації досліджуваних об’єктів.

Згортка – це математична операція, яка створює набір вагових коефіцієнтів, що передаються у наступні шари для подальшої обробки.

За результатами аналізу наведених вище методів виявлення фейкових новин для опрацювання авторського методу запропоновано використовувати 5-ти шарову згорткову нейромережу та вдосконалити її архітектуру. На рисунку 2.1 представлено початкову структуру нейромережі.

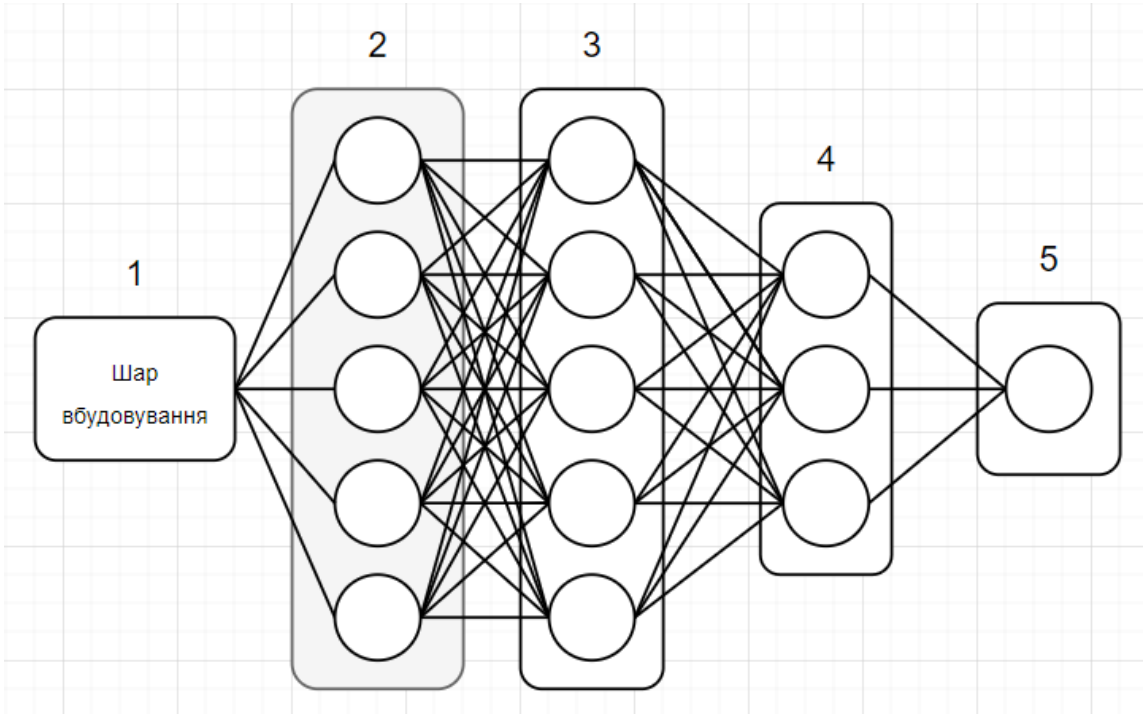


Рисунок 2.1 – Початкова структура нейронної мережі

Дана нейромережа складається з 5-ти шарів.

1. Шар вбудовування (Embedding layer): цей шар перетворює вхідні текстові дані в щільне векторне представлення. Він використовує попередньо навчені вектори слів та встановлює ваги як незмінювані, що в свою чергу означає, що ваги вбудовування залишатимуться незмінними під час навчання моделі. Розмір входу даного шару є розміром словника досліджуваного тексту.

2. Одновимірний шар згорткового перетворення (Conv1D Layer): одновимірний згортковий шар з 64 фільтрами та розміром ядра 5. Він використовується для

застосування згорткового перетворення до вхідних даних. На даному шарі було використано функцію активації ReLU (Rectified Linear Unit).

3. Одновимірний шар максимального об’єднання (MaxPulling1D layer): цей шар використовується для виділення найбільш важливих ознак із вхідних даних (вихідних даних з попереднього шару) та зменшення їх розміру.

4. Шар LTSM (LTSM layer): це шар довгострокової пам’яті з 64 одиницями. LTSM – це тип рекурентного шару, який може захопити послідовну інформацію. Він використовується у даній мережі для моделювання текстових даних.

5. Повнозв’язний шар (Danse layer): цей шар є моделлю перцептрона, в якому усі нейрони з’єднані із нейронами попереднього шару та сигмоїдальною функцією активації. Він відповідає за кінцевий бінарний вихід класифікації. Тобто даний шар безпосередньо дає відповідь на запитання «чи правдива досліджувана новина?».

Дана модель компілюється з використанням бінарної перехресної ентропії, як функції втрат, та оптимізатора Adam. Вона розроблена для бінарної класифікації, а сигмоїдальна функція активації в останньому шарі дозволяє моделі передбачати ймовірності для бінарних міток.

Оптимізатор Adam, що запропонований Деніелом Кінгмо та Джимом Ба у статті «Adam: A Method for Stochastic Optimization» [77], є алгоритмом оптимізації, який використовується для тренування нейронних мереж. Цей оптимізатор комбінує в собі ідеї з інших алгоритмів оптимізації.

До основних характеристик даного оптимізатора включають:

1. Миттєву адаптацію кроку навчання. Він використовує експоненційно зважене середнє значення для оцінки послідовних моментів градієнту. Це дозволяє налаштовувати крок навчання для кожного параметру.

Як відомо, градієнтом називають вектор, що спрямований у бік максимальної зміни функції. У контексті нейронних мереж градієнт функції помилки допомагає визначити, як зміна того чи іншого параметра впливає на значення функції помилки. Тобто, він дозволяє встановити, як потрібно змінити параметри моделі, щоб зробити помилку меншою [78].

2. Біас-коригування. Оптимізатор включає механізм біас-коригування, що компенсує те, що початкові оцінки моментів можуть бути спотвореними нульовими значеннями. Це поліпшує стабільність і збіжність алгоритму.

У випадку застосування оптимізатора Adam процес біас-корегування виглядає наступним чином:

$$\acute{m}_t = \frac{m_t}{1 - \beta_1^t}, \tag{2.1}$$

$$\acute{v}_t = \frac{v_t}{1 - \beta_2^t}, \tag{2.2}$$

де:

- $\acute{m}_t$ та $\acute{v}_t$ – скориговані оцінки першого та другого моментів на кроці *t;*
- m_t та v_t – некориговані перші та другі моменти на кроці *t;*
- β_1 та β_2 – параметри затухання;
- t – номер поточної ітерації.

Біас-коригування важливе для того, щоб у перших кроках навчання алгоритм не надто агресивно збільшував чи зменшував оцінки моментів. Без цього коригування можуть виникати проблеми у вигляді великих початкових значень оцінок, що може впливати на збіжність алгоритму.

Загальна ідея полягає в тому, що на ранніх етапах навчання оцінки моментів можуть бути зміщені через використання $1 - \beta^t$, де t – номер поточної ітерації. Біас-коригування компенсує це зміщення, забезпечуючи більш стабільні та точні оцінки моментів градієнту.

3. Масштабування градієнту. Adam масштабує градієнт перед оновленням параметрів. Це дозволяє ефективніше використовувати крок навчання для різних параметрів.

4. Параметри. Adam має кілька гіперпараметрів, таких як крок навчання, затухання першого моменту, затухання другого моменту та епсілон (маленьке число, щоб уникнути ділення на нуль).

Основна формула для оновлення параметрів за допомогою Adam виглядає наступним чином:

$$\theta_{t+1} = \theta_t - \frac{a}{\sqrt{\acute{\upsilon}_t} + \varepsilon} * \acute{m}_t, \tag{2.3}$$

де:

- $\acute{m}_t$ та $\acute{\upsilon}_t$ – кориговані оцінки моментів, що отримуються з формул (2.1) та (2.2);
- a – крок навчання;
- ε – невелике число для уникнення ділення на нуль.

Оптимізатор Adam часто використовується для тренування нейронних мереж через його ефективність та швидкість збіжності.

При цьому, сигмоїдальна функція активації, що використовується в останньому шарі, є одним із найчастіше використовуваних типів передавальних функцій. Введення функцій сигмоїдального типу було обумовлене обмеженістю нейронних мереж із пороговою функцією активації нейронів – за такої функції активації будь-який із виходів мережі дорівнює або нулю, або одиниці, що обмежує використання мереж не в задачах класифікації. Використання сигмоїдальних функцій дозволило перейти від бінарних виходів нейрона до аналогових. Функції передачі такого типу, як правило, властиві нейронам, що знаходяться у внутрішніх шарах нейронної мережі [79].

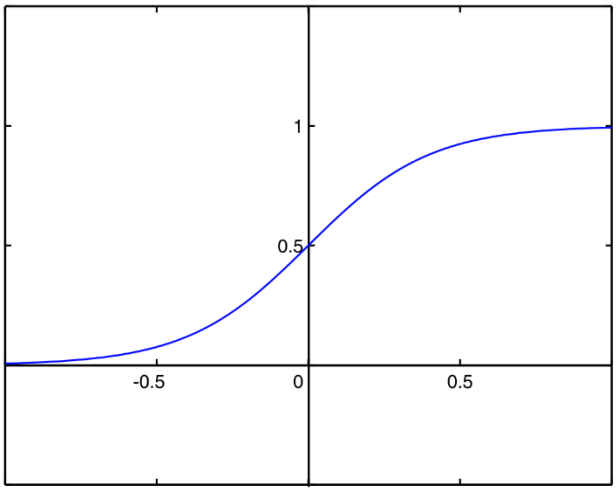


Рисунок 2.2 – Сигмоїдальна функція активації

Провівши аналіз даної моделі було встановлено, що модель демонструє досить хороші результати на наборах даних, на яких безпосередньо проводилось навчання (91,2%), проте коли моделі необхідно розпізнати текст, що не був присутній у навчальній вибірці, показник точності моделі зменшувався до 86,6%. Тому було прийнято рішення оптимізувати архітектуру досліджуваної мережі для отримання більш достовірної класифікації новин на наборах даних, що не були присутніми у навчальній вибірці.

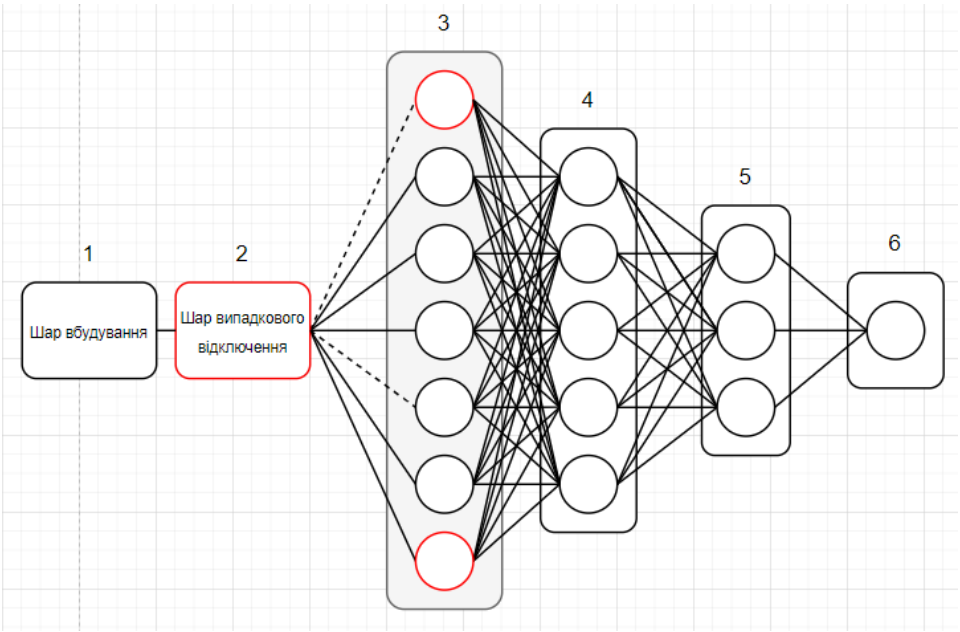


Рисунок 2.3 – Удосконалена архітектура нейромережі

Нова удосконалена структура має наступні зміни:

1. Було додано шар випадкового відключення (Dropout layer): цей шар є захисним шаром, він допомагає запобігти перенавчанню, випадково відключаючи певну частину вхідних одиниць. При дослідженнях було виявлено, що найоптимальнішою кількістю одиниць для відключення є 25% від загальної кількості. Тобто при навчанні нейромережа ігноруватиме 25% вхідних одиниць (вхідних слів) кожної новини. Таким чином замість того, щоб підібрати ваги тільки для навчального набору даних, нейромережа вчиться підбирати відповідь для схожих даних, що не

зустрічалися у навчальному наборі. Цей шар значно покращує роботу мережі із новинами, з якими нейромережі ще не доводилось працювати.

2. Було збільшено розмір ядра з 5 до 7 в одновимірному шарі згорткового перетворення, що дозволяє моделі краще виділяти основні ознаки досліджуваних об’єктів та відкидати дрібниці.

3. Змінено функцію активації одновимірного шару максимального об’єднання з сигмоїдальної функції активації на функцію активації ReLU.

Функція активації ReLU – це нелінійна функція, яка широко використовується в нейронних мережах, особливо в глибоких нейромережах. До основних характеристик даної функції можна віднести:

- Нелінійність. Функція є нелінійною, що дозволяє нейромережам моделювати складні залежності та здатність вирішувати нелінійні завдання. Без нелінійних функцій активації мережа втрачає свою здатність навчання складних завдань.
- Простота та ефективність. ReLU має просту структуру і обчислювально ефективна.

Рисунок 2.4 – Функція активації ReLU

Ця архітектура поєднує вбудовування та згорткові шари для захоплення локальних ознак тексту, за якими слідує LTSM-шар для захоплення довгострокових залежностей, використовуючи які повнозв’язний шар створює бінарний вихід класифікації.

Додавання шару випадкового відключення у дану модель призвело до того, що модель стала значно краще працювати із новинами, які не зустрічались їй у навчальному наборі даних. Із цим шаром модель самостійно підбирає ваги для векторних одиниць тексту кожної новини. А збільшення розмірності ядра у шарі згорткового перетворення призвело до того, що модель може виявляти більші ознаки та відкидати дрібниці у кожній новині, проте, в свою чергу, це має і недолік у вигляді збільшення вірогідності перенавчання мережі. Саме тому використання шару випадкового відключення у комбінації із збільшенням розмірності ядра згорткового шару призводить до покращення виявлення ваг та ознак і зниження шансу перенавчання при цьому.

2.3 Статистичні показники для оцінки якості виявлення фейкових новин

Для оцінки якості навчання нейромережевих класифікаторів виявляти фейкові новини пропонується використати наступні статистичні метрики: точність (A), точність (P), чутливість (R), $F1$-показник, площу під ROC-кривою (AUC), а також помилки першого та другого роду.

Точність (A) – це міра здатності класифікатора правильно класифікувати інформацію як фейкову або реальну. Точність (A) можна оцінити так

$$A_{ccuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{2.4}$$

де TP, TN, FP, FN - це, відповідно, справжні позитиви, справжні негативи, помилкові позитиви і помилкові негативи.

Точність (P) – це міра точності класифікатора, згідно якої мале значення вказує на велику кількість помилкових позитивних результатів. Точність (P) є кількісттю позитивних прогнозів, поділених на загальну кількість позитивних передбачених значень класу, і обчислюється за формулою

$$P_{recision} = \frac{TP}{TP + FP}. \tag{2.5}$$

Чутливість (R) є мірою повноти класифікатора (наприклад, низьке значення чутливості вказує на багато помилкових негативних результатів), де кількість справжніх позитивів ділиться на суму справжніх позитивів і кількості помилкових негативів:

$$R_{ecall} = \frac{TP}{TP + FN}. \tag{2.6}$$

$F1$-показник ($F1$) обчислюється як зважена гармонічна середня точності і чутливості класифікатора:

$$F1 = \frac{2 \cdot P_{recision} \cdot R_{ecall}}{P_{recision} + R_{ecall}} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}. \tag{2.7}$$

Площа під ROC-кривою, відома як AUC, є мірою, яка використовується для порівняння алгоритмів навчання та побудови оптимальних моделей навчання. AUC,

що близька до 1, вказує на перевагу системи, яка може точно відрізняти реальні та фейкові новини, тоді як AUC, що близька до 0, вказує на слабку систему (тобто система вважатиме всі фейкові новини реальними і навпаки) [38]. Наприклад, якщо AUC дорівнює 0,9, це означає, що модель на 90% може відрізняти негативні і позитивні класи. AUC зазвичай використовується в задачах незбалансованої класифікації [37], таких як виявлення фейкових новин.

Міра AUC може бути знайдена з виразу

$$AUC = \frac{1 - FPR + TPR}{2}. \quad (2.8)$$

Справжній позитивний рівень або TPR - це скорочення, яке вказує на відсоток позитивних прикладів, які успішно класифіковані. На відміну від цього, FPR, який вказує на хибно позитивний рівень, це відношення випадків, які були неправильно класифіковані як негативні, до всіх інших випадків [39].

Помилки першого роду та помилки другого роду – поняття математичної статистики та її прикладних застосувань. Вони використовуються під час перевірки статистичних гіпотез у різних галузях науки і техніки, коли йдеться про ухвалення «бінарного» рішення на основі якогось критерію, який з деякою ймовірністю може давати помилковий результат. Якщо істинна гіпотеза помилково відкидається, то ця помилка називається помилкою першого роду. Якщо помилково приймається хибна гіпотеза – це помилка другого роду.

Висновки до розділу 2

У ході опрацювання другого розділу було здійснено аналіз того, що собою являють фейкові новини, якими вони бувають і які методи та прийоми боротьби з ними є ефективними. Встановлено, що незважаючи на те, що на сьогодні фейкові новини є досить розповсюдженим явищем, комплексні дослідження цього феномена поки відсутні. Немає чітких критеріїв визначення поняття «фейкові новини». Детерміновано зазначене поняття для даної роботи. Проаналізовано можливі підходи до класифікації фейкових новин і встановлено відсутність єдиної класифікації.

Здійснено оцінку методів виявлення фейкових новин з точки зору використання їх потенціалів для визначення авторського підходу до формування методу виявлення фейкових новин в Інтернеті на основі використання нейромережевих засобів.

Запропоновано авторський підхід до вдосконалення структури нейромережі для виявлення фейкових новин. В основу авторського методу виявлення фейкових новин покладено нейромережеві засоби, а саме багатошарову CNN нейромережу. Вдосконалено архітектуру досліджуваної мережі для отримання більш достовірної класифікації новин на наборах даних, що не були присутніми у навчальній вибірці. Описано та обгрунтовано суть удосконалення, що стосується додавання шару випадкового відключення (Dropout layer), збільшення розміру ядра, зміни функції активації одновимірного шару максимального об’єднання з сигмоїдальної функції активації на функцію активації ReLU.

Для оцінки якості навчання нейромережевих класифікаторів виявляти фейкові новини запропоновано використання таких статистичних метрик, як: точність (A), точність (P), чутливість (R), $F1$-показник, площу під ROC-кривою (AUC), помилки першого роду, помилки другого роду. Наведено їх аналітичні залежності та оцінено можливість їх використання для визначення рівня якості архітектури досліджуваної моделі.

РОЗДІЛ 3

Програмна реалізація методу виявлення фейкових новин на основі використання нейромережевих технологій

3.1 Опис засобів програмної реалізації

З метою реалізації методу виявлення фейкових новин для оцінки достовірності джерел фейкової інформації необхідним є створення відповідного додатку, який надавав би можливість його користувачам проводити тестування методу. В ході проведення досліджень було прийнято рішення щодо розробки додатку у веб-середовищі. Прийняте рішення пояснюється тим, що веб-середовище дозволяє легко обробляти великі обсяги відеоданих за рахунок можливості роботи з розподіленими системами та використовувати хмарні рішення для обробки даних, що гарантує високу продуктивність і швидкість обробки. Додатки, розроблені у веб-середовищі, легко поширювати та оновлювати. А це дозволяє швидко впроваджувати оновлені моделі та забезпечувати користувачам доступ до оновлень без необхідності переінсталяції додатку.

Програмну реалізацію запропонованого методу можна представити у вигляді веб-додатку, в якому основною та ключовою частиною є метод виявлення фейкових новин за допомогою нейромережі.

Сучасні веб-додатки, як правило, складаються з двох компонентів: фронтенду та бекенду.

Для розробки backend частини додатку було обрано мову прогамування PHP та фреймворк даної мови програмування Laravel. Даний фреймворк є потужним інструментом розробки логічних частин багатьох веб-додатків. Даний фреймворк надає можливість працювати з різними базами даних та реалізує усі основні принципи ООП та принципи SOLID.

Основні особливості фреймворку Laravel такі:

- Простий синтаксис. Laravel вражає елегантним та зрозумілим синтаксисом, що спрощує розробку та підтримку коду. Використання фреймворку дозволяє розробникам швидко та ефективно реалізувати функціонал веб-додатків.
- Модульність та розширюваність. Фреймворк надає модульну структуру, що дозволяє розробникам легко організовувати та розширювати функціонал додатків. Застосунки, побудовані на Laravel, можуть бути легко масштабовані та доповнені новими функціями.
- Вбудована ORM. Laravel використовує Eloquent ORM для зручної роботи з базами даних, що спрощує взаємодію з даними та забезпечує безпеку від SQL-ін'єкцій.

ORM – це підхід до роботи з базами даних в програмуванні, який дозволяє використовувати об'єктно-орієнтований код для взаємодії з реляційними базами даних. Замість того, щоб вручну писати SQL-запити для взаємодії з базою даних, розробники можуть використовувати об'єктно-орієнтовані класи та методи для роботи з даними. ORM перетворює дані між об'єктами в програмі та записами в базі даних, спрощуючи взаємодію з базою даних і забезпечуючи більшу абстракцію. Це дозволяє розробникам працювати з даними, як з об'єктами, а не з таблицями бази даних і використовувати об'єктно-орієнтовані концепції, такі як спадкування та асоціації.

- Широкий функціонал. Фреймворк має велику кількість вбудованих функціональностей, таких як система маршрутизації, автентифікації, кешування, тестування та інші, що роблять процес розробки більш ефективним та швидким.

Таким чином, використання фреймворку Laravel дозволяє розробникам ефективно будувати високоякісні та зручні веб-додатки, використовуючи передові технології розробки веб-застосунків.

Оскільки для реалізації backend частини додатку було обрано фреймворк Laravel, то практично очевидним вибором для frontend частини додатку є фреймворк Vue.js, оскільки навіть згідно офіційної документації даних технологій, вони є надзвичайно добре інтегрованими одна в одну. Фреймворк Vue.js є сучасним

фреймворком розробки веб застосунків, що використовує новітні технологій для збірки та компіляції frontend частин веб-додатків, а Laravel, у свою чергу, має вбудований компілятор Vite саме для збірки таких frontend частин додатків.

Vite – це швидкий і сучасний інструмент для розробки веб-додатків, який особливо підходить для розробки великих односторінкових застосунків (SPA) та веб-сайтів. Однією з його ключових особливостей є швидкість завантаження, що робить його відмінним вибором для розробки.

Vue.js – це прогресивний фреймворк для побудови користувацьких інтерфейсів (UI) веб-додатків. Він спрощує розробку веб-інтерфейсів, забезпечуючи декларативний синтаксис та простий API для взаємодії з DOM. Vue.js орієнтований на інкрементальне впровадження, тобто його можна використовувати як в маленьких проектах, так і в більших, поступово нарощуючи його функціонал.

До основних переваг даного фреймворку можна віднести:

- Прогресивність. Можна використовувати частину Vue.js в існуючому проекті, не переключаючи весь код на фреймворк.
- Компонентна архітектура. Vue дозволяє побудову користувацьких інтерфейсів через використання компонентів, що спрощує організацію коду та підтримує його перевикористання.
- Реактивність. Фреймворк використовує систему реактивності, яка автоматично відслідковує зміни даних і оновлює DOM за необхідністю.
- Екосистема та інструменти. Vue.js має широку екосистему, включаючи додатки для стану керування (Vuex), маршрутизації (Vue Router) та інші.
- Крос-платформеність. Даний фреймворк підтримується практично всіма браузерами, що дозволяє додатку функціонувати практично на всіх видах девайсів.

Для реалізації стилістичної складової веб-застосунку було обрано фреймворк мови CSS – Tailwind CSS. Він є сучасним фреймворком для створення веб-інтерфейсів, який надає гнучкі та повністю налаштовувані стилі за допомогою класів. Замість написання CSS правил вручну, розробники можуть використовувати класи

заздалегідь визначених стилів, які Tailwind надає. До основних характеристик даного CSS фреймворку можна віднести таке:

- Модульність. Tailwind CSS розбитий на модулі, які можна включати або виключати в залежності від потреб проекту. Це дозволяє уникнути зайвого завантаження непотрібних стилів.
- Реактивність. Система реактивності Tailwind дозволяє вказувати, які елементи потрібно змінювати у відповідь на зміну даних. Вона автоматично оновлює стилі без необхідності власного написання JavaScript.
- Адаптивність. Фреймворк має вбудовану підтримку адаптивного дизайну. Зокрема, можна визначити різні стилі для різних розмірів екрану, що спрощує створення адаптивних додатків.
- Кастомізація. Хоча Tailwind постачається з широким набором вбудованих стилів, можна легко кастомізувати його, додаючи власні конфігурації та стилі.

Безпосередньо для реалізації методу виявлення фейкових новин було обрано мову програмування Python. Python – це високорівнева, інтерпретована мова програмування, яка відзначається простотою синтаксису та читабельністю коду. Оригінально була розроблена Гвідо ван Россумом та вперше випущена у 1991 році. Python був розроблений, щоб бути доступним і легким для вивчення, а його універсальність дозволяє використовувати його для різноманітних завдань, від веб-розробки до наукових досліджень. До основних характеристик даної мови програмування можна віднести:

- Об'єктно-орієнтована парадигма. Python підтримує об'єктно-орієнтовану парадигму програмування, дозволяючи створювати класи, об'єкти та використовувати концепції спадкування та поліморфізму.
- Читабельний та зрозумілий синтаксис. Синтаксис Python славиться своєю простотою і легкістю читання, що полегшує розробку і розуміння коду.
- Підтримка різних платформ. Python може працювати на різних операційних системах, включаючи Windows, macOS, Linux/UNIX та Android.

- Широкий спектр застосувань. Мова Python використовується у різних галузях, включаючи веб-розробку, штучний інтелект, обробку даних, аналіз та багато інших областей.

Також для програмної реалізації даного методу було використано наступні бібліотеки:

- TensorFlow. TensorFlow – це відкрите програмне забезпечення для розробки моделей машинного навчання та глибокого навчання. Розроблений командою в Google Brain, воно використовує графову модель обчислень, що дозволяє ефективно виконувати операції на різних пристроях. TensorFlow сумісний з різними платформами, включаючи операційні системи та мобільні пристрої, і знаходить застосування в різних галузях, таких як розпізнавання образів, обробка природних мов, генерація контенту та інші. Вбудовані інструменти для навчання та інференції, а також підтримка обчислень на GPU роблять TensorFlow потужним інструментом для завдань машинного навчання. З великою та активною спільнотою розробників, візуалізацією графів обчислень та високим рівнем підтримки, TensorFlow залишається важливим інструментом у світі штучного інтелекту.

- NumPy. NumPy є бібліотекою мови програмування Python, яка надає високорівневі математичні та числові функції. Вона стала стандартом для наукових обчислень у середовищі Python. NumPy використовується для роботи з масивами, векторами, матрицями та великою кількістю математичних функцій, що полегшує обробку та аналіз даних. NumPy дозволяє виконувати операції лінійної алгебри, обробляти та аналізувати дані, генерувати випадкові числа та виконувати багато інших завдань. Його основною структурою даних є N-розмірний масив, що дозволяє ефективно використовувати масиви даних в багатовимірних обчисленнях. Бібліотека також інтегрується з іншими популярними бібліотеками для наукових обчислень та машинного навчання, що робить її важливим інструментом для роботи в цих областях.

- Scikit-learn. Scikit-learn – це бібліотека машинного навчання для мови програмування Python, яка надає ефективні інструменти для роботи з класичними

алгоритмами машинного навчання. Вона розроблена на основі NumPy, SciPy та Matplotlib, що робить її високопродуктивною та відмінною для вивчення та застосування в галузі наукових досліджень та розробки. Дана бібліотека включає широкий спектр алгоритмів машинного навчання, таких як класифікація, регресія, кластеризація, а також інструменти для валідації моделей та вибору параметрів. Бібліотека покликана спростити процес розробки та експериментів з моделями машинного навчання.

Загалом даний набір технологій добре синергує та взаємодіє одна з одною. Функціональну схему даного застосунку можна оцінити з рисунку 3.1.

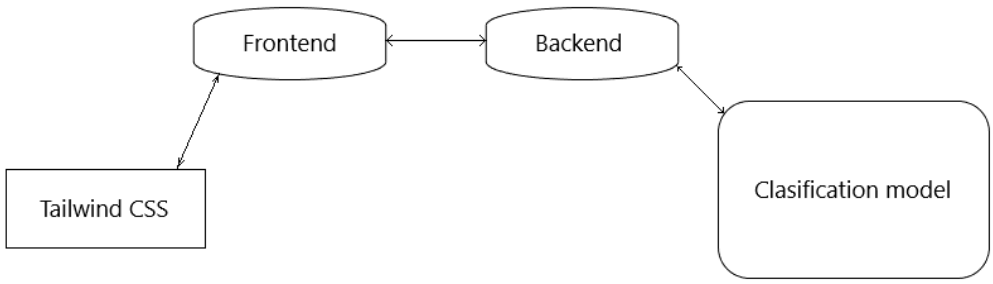


Рисунок 3.1 – Функціональна схема застосунку

3.2 Розробка структури інформаційної системи

Для розробки та візуалізації структури інформаційної системи було використано UML-діаграми.

UML-діаграми – це графічні інструменти, які використовуються для моделювання різних аспектів програмних систем та їх взаємодії. Ці діаграми надають стандартизований спосіб відображення структури та поведінки системи. Вони допомагають команді розробників та іншим учасникам проекту краще зрозуміти архітектуру, функціональність та взаємодію компонентів системи.

У UML існують різні види діаграм, кожна з яких призначена для моделювання конкретного аспекту системи. Це включає діаграми класів, діаграми прецедентів,

діаграми послідовності, діаграми активності, діаграми компонентів, діаграми розгортання та інші. Кожна діаграма має свою власну специфіку та застосування, сприяючи повнішому розумінню різних аспектів програмної системи.

Для розробки послідовності взаємодії із інформаційною системою було розроблено діаграму активності та діаграму послідовності.

Діаграма активності – це вид UML-діаграм, який моделює потік управління та обмін повідомленнями між різними елементами системи в рамках конкретної діяльності або процесу. Вона особливо корисна для візуалізації послідовності дій, рішень та обміну даними між різними компонентами системи. Дану діаграму можна оцінити з рисунку 3.2.

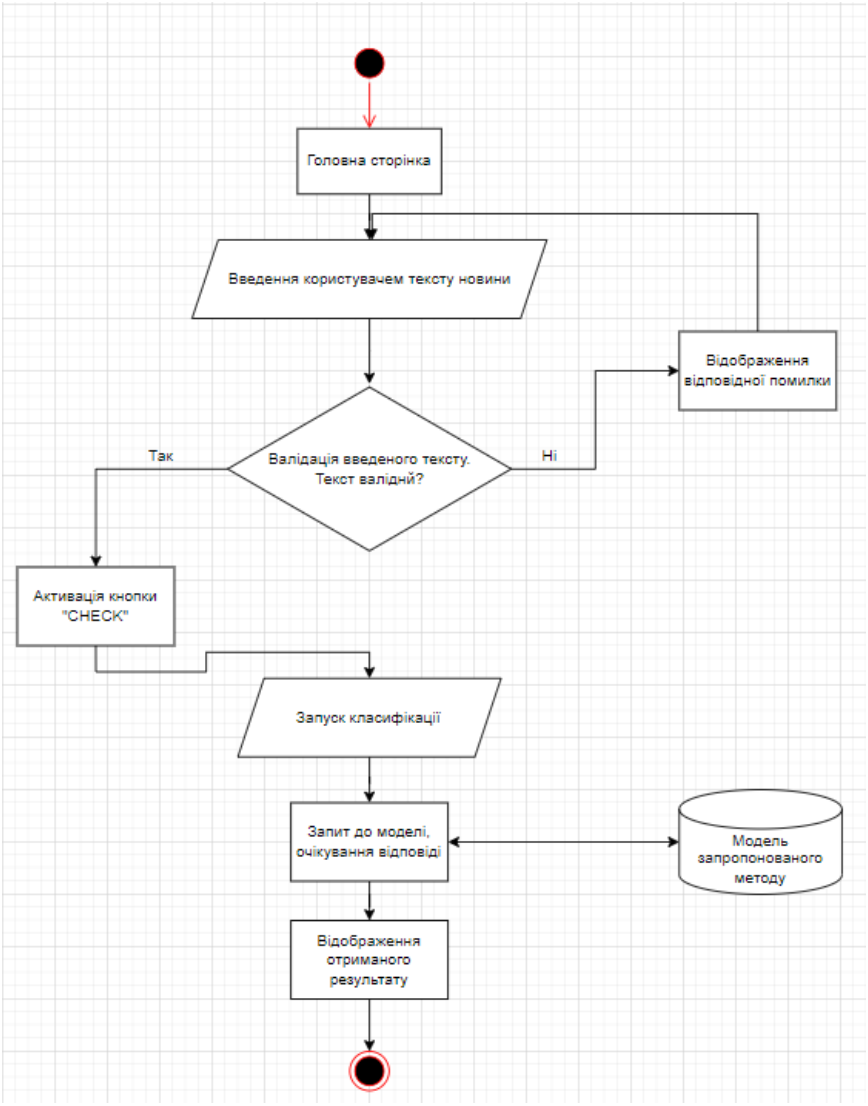


Рисунок 3.2 – UML-діаграма активності

Діаграма послідовностей – це графічний інструмент, який дозволяє моделювати взаємодію між різними об'єктами чи компонентами в системі в рамках конкретного сценарію або виклику подій. У цій діаграмі вказуються об'єкти, що беруть участь у взаємодії, та з'ясовується порядок виконання їх операцій.

Об'єкти представлені вертикальними лініями, а взаємодія між ними показується стрілками, що вказують на порядок викликів та відповідей. Важливо враховувати час виконання операцій та взаємодії між об'єктами для точного відображення сценарію.

Ця діаграма допомагає візуалізувати послідовність подій і викликів між різними об'єктами, що допомагає в розумінні взаємодії компонентів у системі в конкретному контексті. Відповідна діаграма відображена на рисунку 3.3.

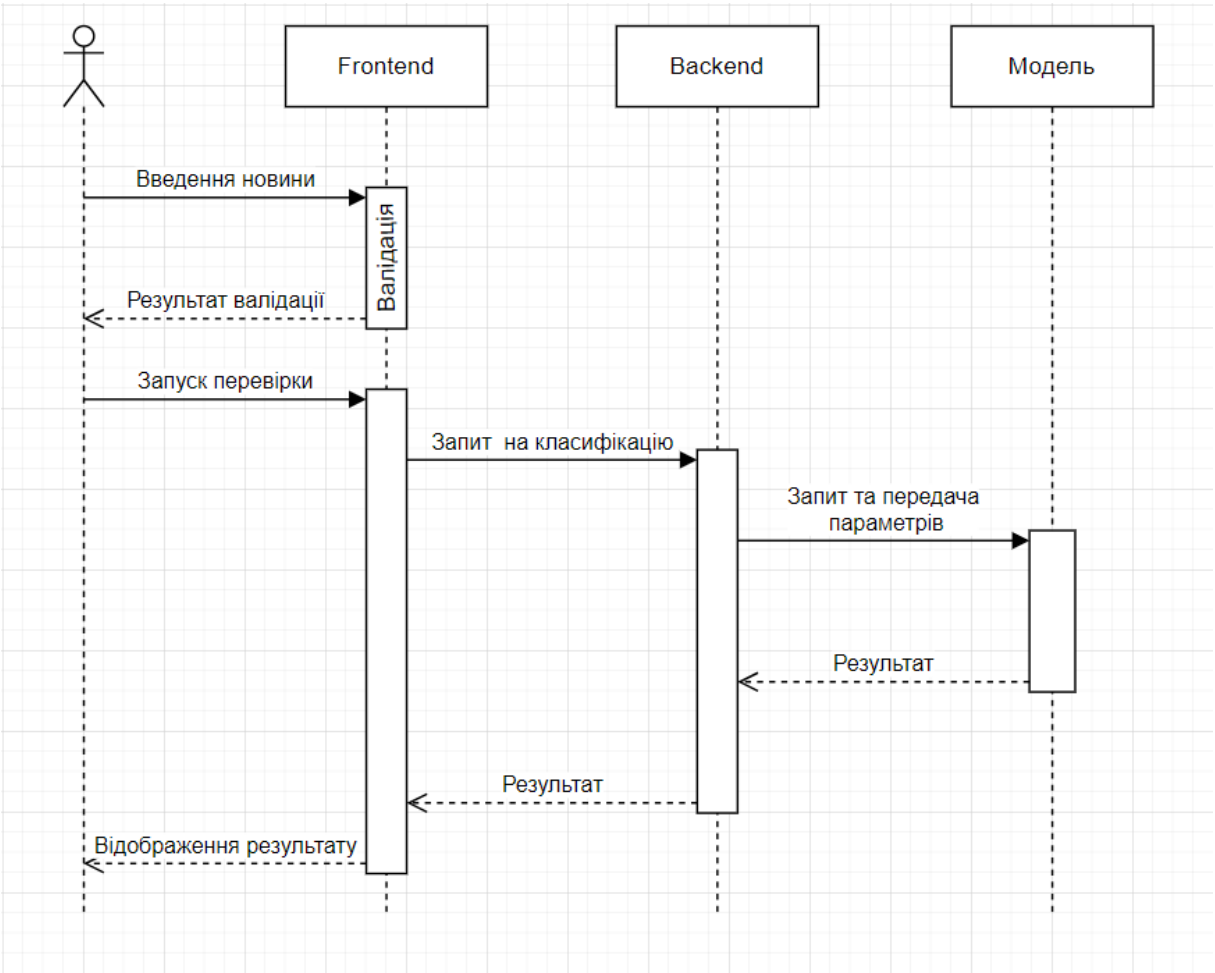


Рисунок 3.3 – UML-діаграма послідовностей

Крім того, для відображення програмної структури моделі методу виявлення фейкових новин було розроблено діаграму класів.

Діаграма класів у мові моделювання UML є графічним інструментом, призначеним для відображення структури системи. На цій діаграмі зображуються класи, їх атрибути і методи, а також взаємозв'язки між класами. Вона слугує для візуалізації ключових аспектів об'єктно-орієнтованого програмування та допомагає в розумінні взаємодії між класами в системі.

Діаграма класів відображає важливі аспекти, такі як спадкування (наслідування), асоціації, агрегації та композиції. Кожен клас представляє об'єкт або сутність у системі, а його атрибути і методи вказують на характеристики та поведінку цього класу.

Ця діаграма допомагає розробникам та архітекторам програмного забезпечення легше розуміти структуру системи та взаємозв'язки між компонентами, що полегшує подальшу розробку та обслуговування системи.

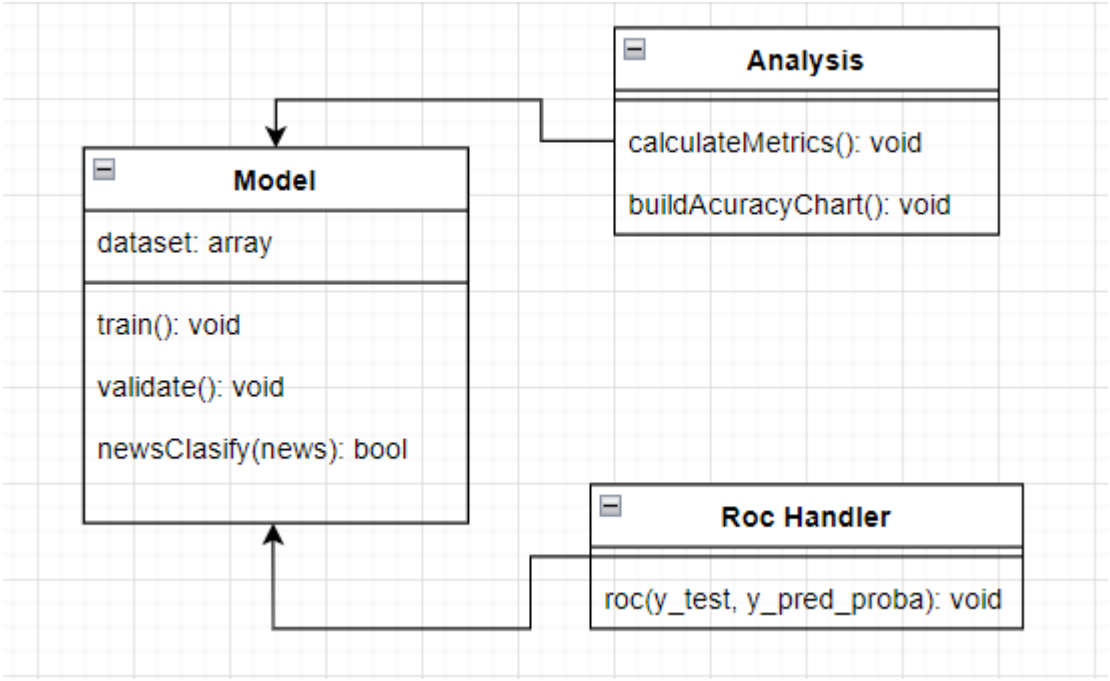


Рисунок 3.4 – UML-діаграма класів

3.3 Розробка прикладних компонентів додатку запропонованого методу

Для реалізації користувачем запропонованого методу виявлення фейкових новин було розроблено інформаційну систему, що працює у веб-середовищі. На рисунку 3.5 можна побачити вигляд головної сторінки інформаційної системи, вона відображається користувачеві при початку роботи з ІС.

Рисунок 3.5 – Головна сторінка інформаційної системи реалізації методу виявлення фейкових новин

Для початку реалізації користувачем запропонованого методу, йому необхідно ввести текст новини, яку він бажає перевірити на достовірність, у відповідне поле (рисунок 3.6). При введені тексту, користувач повинен ввести принаймні 10 слів тексту новини, при невиконанні цієї умови кнопка «CHECK» залишатиметься неактивною та користувачеві буде відображена відповідна помилка валідації, яку можна побачити на рисунку 3.7.

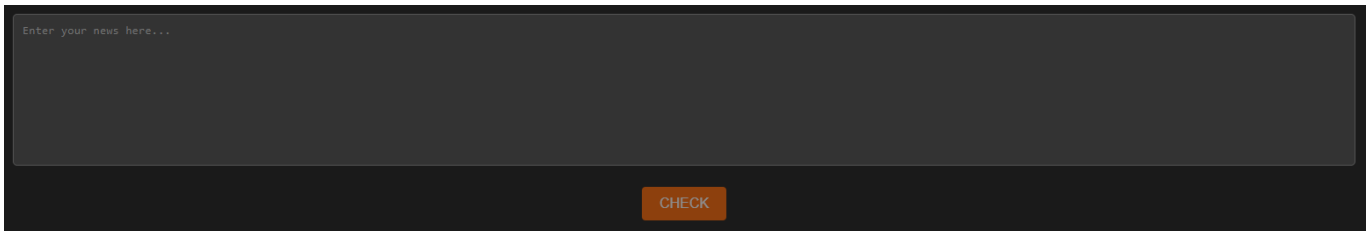


Рисунок 3.6 – Поле для введення досліджуваної новини

Рисунок 3.7 – Відображення помилки валідації введеного тексту новини

Якщо ж користувач ввів повний текст новини та виконав вищезазначену умову валідації, кнопка «CHECK» стане активною, та користувач зможе провести безпосереднє дослідження новини, натиснувши на дану кнопку (див. рисунок 3.8).

Рисунок 3.8 – Вигляд інформаційної системи при коректно введеному тексті досліджуваної новини

Натиснувши відповідну кнопку інформаційної системи для перевірки достовірності досліджуваної новини, відбудеться взаємодія Frontend та Backend частин ІС та запит до безпосередньо моделі запропонованого методу виявлення фейкових новин. У запиті до моделі буде передано текст новин, який було введено

користувачем у відповідному текстовому полі. Після обробки вхідних даних результатом роботи моделі запропонованого методу буде безпосередня класифікація досліджуваної новини. Результати класифікації можуть набувати двох значень: «REAL», «FAKE». Отримавши відповідь із результатом роботи моделі, користувачеві буде відображено відповідні повідомлення на головній сторінці інформаційної системи, які можна побачити на рисунках 3.9 та 3.10.

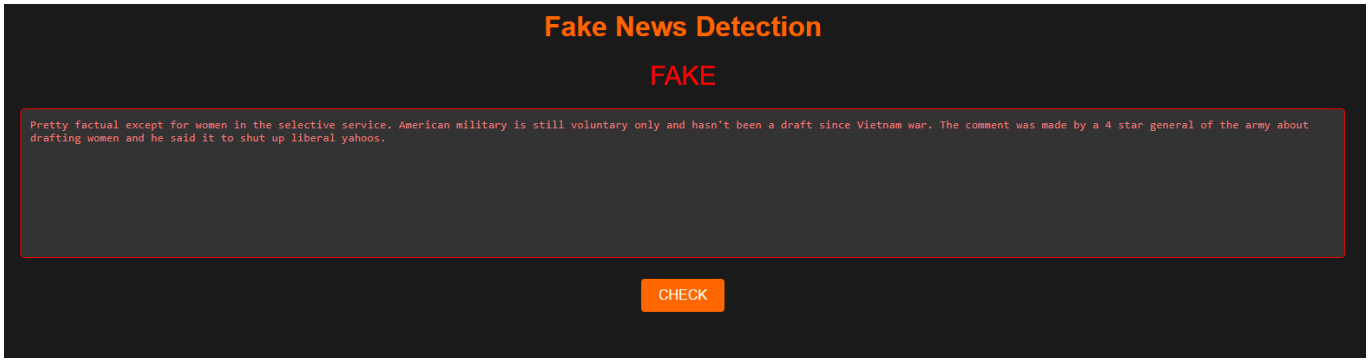


Рисунок 3.9 – Відображення фейкового результату класифікації новини

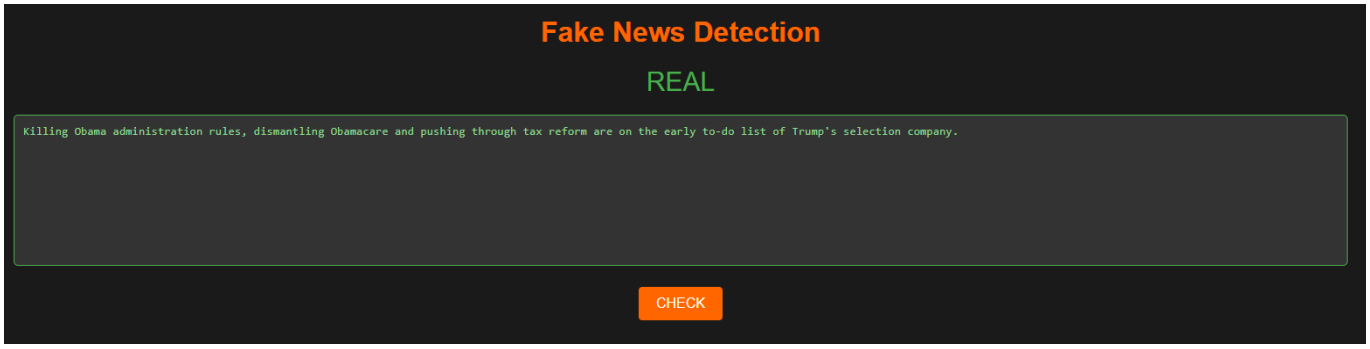


Рисунок 3.10 – Відображення правдивого результату класифікації новини

3.4 Прикладне тестування додатку запропонованого методу

У процесі розробки будь-якого програмного продукту етап тестування функціоналу системи є одним із ключових. Він є необхідним та невід'ємним етапом, оскільки допомагає отримати відомості про коректність роботи програмного додатку відповідно до встановлених вимог. З метою перевірки основних функцій веб-додатку,

який використовує метод виявлення фейкових новин, були розроблені тест-кейси. Перелік тест-кейсів представлено в таблицях нижче.

Таблиця 3.1 – Тест-кейс 01

Назва: Запуск веб-застосунку	
Кроки	Очікуваний результат
1. Запустити застосунок 2. Перейти за посиланням веб-застосунку 3. Зафіксувати отриманий результат	Завантажено головну сторінку
Результат проходження тест-кейсу: пройдений	

Таблиця 3.2 – Тест-кейс 02

Назва: Перевірка введення новин для перевірки	
Кроки	Очікуваний результат
1. Запустити застосунок 2. Перейти за посиланням веб-застосунку 3. Використавши поле форми, ввести текст новини 4. Зафіксувати отриманий результат	Після введення тексту кнопка «CHECK» стала активною
Результат проходження тест-кейсу: пройдений	

Таблиця 3.3 – Тест-кейс 03

Назва: Перевірка класифікації новини	
Кроки	Очікуваний результат

1. Запустити застосунок 2. Перейти за посиланням веб-застосунку 3. Використавши поле форми, ввести текст новини 4. Натиснути на кнопку «CHECK» 5. Зафіксувати отриманий результат	Відображено поле, що відображає класифікацію новини. Поле має значення «REAL» або «FAKE»
Результат проходження тест-кейсу: пройдений	

Таблиця 3.4 – Тест-кейс 04

Назва: Перевірка коректності класифікації новини запропонованим методом виявлення фейкових новин (на наборі даних, на якому було проведено навчання)	
Кроки	Очікуваний результат
1. Запустити застосунок 2. Перейти за посиланням веб-застосунку 3. Обрати новину із навчального набору даних. Обрана новина є правдивою новиною 4. Використавши поле форми, ввести текст новини 5. Натиснути на кнопку «CHECK» 6. Зафіксувати отриманий результат	Відображено поле, що відображає класифікацію новини. Поле має значення «REAL»

Результат проходження тест-кейсу: пройдений

Таблиця 3.5 – Тест-кейс 05

Назва: Перевірка коректності класифікації новини запропонованим методом виявлення фейкових новин (на наборі даних, який не приймав участі у навчанні)	
Кроки	Очікуваний результат
1. Запустити застосунок 2. Перейти за посиланням веб-застосунку 3. Обрати новину із набору даних, що не належить до навчального набору. Обрана новина є фейковою новиною 4. Використавши поле форми, ввести текст новини 5. Натиснути на кнопку «CHECK» 6. Зафіксувати отриманий результат	Відображено поле, що відображає класифікацію новини. Поле має значення «FAKE»
Результат проходження тест-кейсу: пройдений	

За результатами тестування опрацьованої ІС на основі використання вищенаведених тест-кейсів, можна стверджувати, що створений застосунок працює коректно.

Висновки до розділу 3

У ході опрацювання третього розділу було обрано перелік технологій для

програмної реалізації застосунку, що реалізує авторський метод виявлення фейкових новин на основі використання нейромережевих технологій. Здійснено аналіз кожної з них.

Продемонстровано функціональну схему застосунку та UML-діаграми, що пояснюють його структуру та описують взаємодію користувача з ним. Також продемонстровано процес розробки прикладних компонентів інформаційної системи та описано перелік технологій, які були використані для розробки кожного з них.

Розроблено інформаційну систему, що реалізує запропонований метод. Для розробки backend частини додатку обрано мову програмування PHP та фреймворк даної мови програмування Laravel. Для frontend частини додатку обрано фреймворк Vue.js, Для реалізації стилістичної складової веб-застосунку обрано фреймворк мови CSS – Tailwind CSS. Безпосередньо для реалізації методу виявлення фейкових новин було обрано мову програмування Python та бібліотеки TensorFlow, NumPy, Scikit-learn для роботи з нейромережевими засобами.

Проведено практичне тестування інформаційної системи на основі застосування розробленого набору тест-кейсів для перевірки як функціональності інтерфейсу, що взаємодіє з користувачем, так і функціональної частини веб-додатку.

За результатами тестування розроблений додаток було визнано працюючим належним чином.

РОЗДІЛ 4

Експериментальне дослідження запропонованого методу виявлення фейкових новин

4.1 Опис програмного забезпечення для оцінки ефективності запропонованого методу виявлення фейкових новин

Актуальним завданням є проведення оцінки ефективності запропонованого авторського методу виявлення фейкових новин. Для виконання цього завдання вбачається за доцільне здійснити порівняльну оцінку. В якості моделей, що порівнюватимуться, доцільно прийняти запропонований метод з удосконаленою архітектурою та існуючі моделі. В якості існуючих моделей вбачається за доцільне розглянути модель з початковою архітектурою, описаною в пункті 2.2 (надалі TensortFlow classification model), та модель, що базується на використанні LogisticRegression.

Доцільність порівняння авторської моделі з моделлю TensortFlow classification model пояснюється тим, що остання є прототипом авторської моделі. А доцільність аналізу моделі LogisticRegression випливає з такого.

LogisticRegression (логістична регресія) – це метод у машинному навчанні для вирішення задач бінарної класифікації, де модель прогнозує ймовірність належності об'єкта до певного класу. Вона використовує логістичну функцію для визначення ймовірності та логарифмічну функцію втрат для оцінки точності моделі.

Основна ідея логістичної регресії полягає в тому, що вихідна змінна є логарифмічною відносно ймовірності, що дозволяє отримати вивід, який лежить у межах (0; 1). Це важливо для задач класифікації, у яких метою є визначення, чи належить об'єкт до певного класу чи ні.

У логістичній регресії використовуються ваги для кожної ознаки, і модель вирішує, як вони впливають на прогнозовану ймовірність. Під час тренування моделі

ваги оптимізуються за допомогою методу градієнтного спуску чи інших методів оптимізації.

Логістична регресія є популярним інструментом для багатьох завдань класифікації через свою ефективність і здатність працювати з великою кількістю ознак. Вона також застосовується у багатьох галузях, включаючи медицину, фінанси та природничі науки.

Таким чином, універсальність методу LogisticRegression (як у теоретичному, так і прикладному відношенні) є підставою для його залучення до порівняльної оцінки.

Для полегшення проведення порівняльної оцінки (експериментів) та уникнення необхідності адаптації кожної моделі під системні параметри експериментальної електронно-обчислювальної машини пропонується використання програмного середовища «Anaconda».

Anaconda – це інтегроване середовище для обчислювальної обробки даних і наукових обчислень у мові програмування Python. Воно надає зручний інтерфейс для управління пакетами, віртуальними оточеннями та проектами з метою полегшення роботи науковців, дослідників та інших спеціалістів, які використовують Python для обробки даних та аналізу (див. рисунок 4.1).

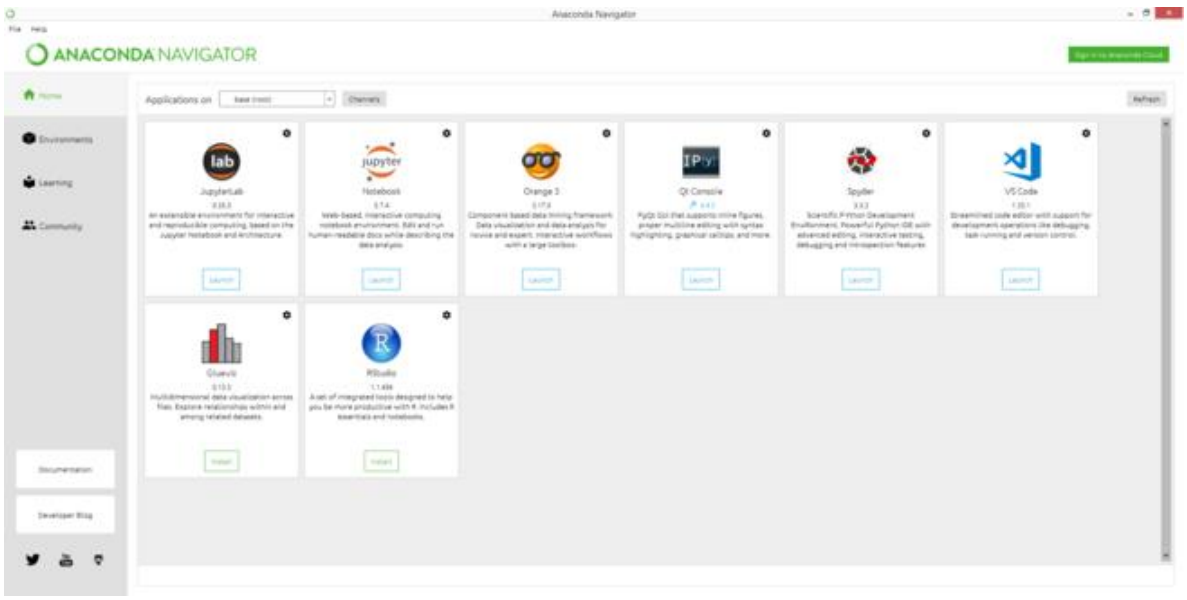


Рисунок 4.1 – Основний екран середовища Anaconda

Anaconda постачається з широким спектром попередньо встановлених наукових бібліотек та інструментів, включаючи NumPy, Pandas, Matplotlib, Jupyter Notebook та інші, що дозволяє користувачам швидко розпочати роботу над проектами без додаткового конфігурування.

Однією з ключових особливостей Anaconda є її система керування пакетами, яка дозволяє легко встановлювати, оновлювати та видаляти пакети. Також Anaconda має можливість створювати віртуальні оточення, що дозволяє ізолювати проекти та їх залежності для уникнення конфліктів.

Зокрема, на вкладці «Environments» в Anaconda Navigator користувач може керувати середовищами Python для своїх проектів (див. рисунок 4.2). Середовища дозволяють ізолювати та управляти версіями пакетів та залежностями між різними проектами.

Основні функції на вкладці «Environments» при цьому такі.

Створення нового середовища: Користувач може створити нове середовище, вибравши версію Python та інші параметри.

Відображення існуючих середовищ: Усі створені середовища відображаються разом із відомостями про використання місця.

Встановлення та оновлення пакетів: Користувач може вибрати конкретне середовище та встановлювати, видаляти або оновлювати пакети, використовуючи графічний інтерфейс.

Імпорт і експорт середовища: Ця функція дозволяє експортувати конфігурацію середовища для подальшого використання або імпортувати існуючі середовища.

Перегляд інформації про пакети: Користувач може переглядати список установлених пакетів в обраному середовищі разом з їх версіями.

Запуск консолі середовища: Дозволяє відкривати термінал або консоль, пов'язану з обраним середовищем, де користувач може виконувати команди та взаємодіяти з Python у межах цього середовища.

Вкладка «Environments» дозволяє користувачам ефективно управляти робочими середовищами Python для різних проектів та експериментів.

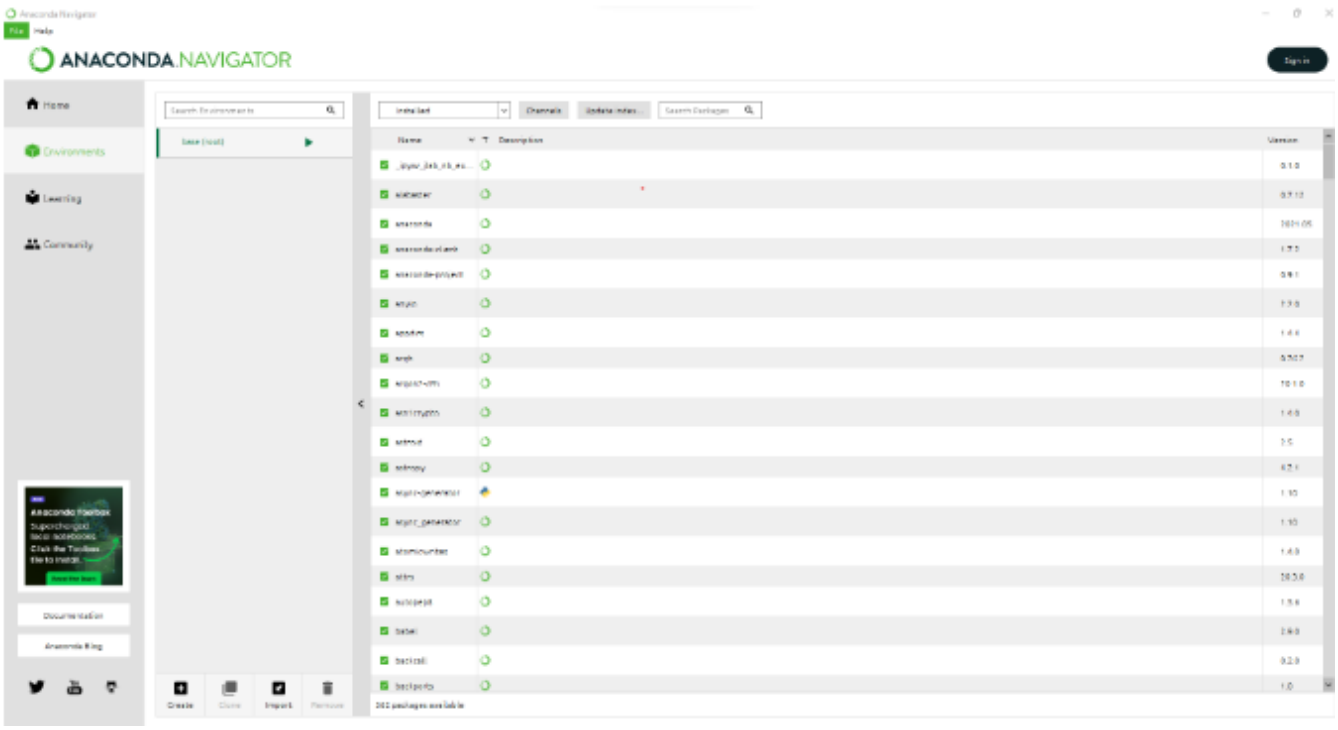


Рисунок 4.2 – Вкладка «Environments» середовища Anaconda

Anaconda Navigator дозволяє також використовувати середовище розробки для Jupyter Notebook, яке є потужним інструментом для виконання коду, вивчення даних, візуалізації та здійснення наукових обчислень в інтерактивному режимі (див. рисунок 4.3). Основні особливості та функції, пов'язані з Jupyter Notebook в Anaconda Navigator, включають:

Управління файлами: З використанням Jupyter Notebook можна працювати з файлами, створювати нові блокноти, відкривати і редагувати існуючі, а також зберігати їх.

Використання ресурсів Anaconda: Jupyter Notebook вбудований у середовище Anaconda, що означає доступ до широкого спектру наукових бібліотек та інструментів для обробки даних, машинного навчання, наукових обчислень та інших областей.

Інтерактивна робота з кодом: Можна виконувати код почергово, додавати коментарі, вставляти графіки та зображення, що допомагає створювати інтерактивні та динамічні документи.

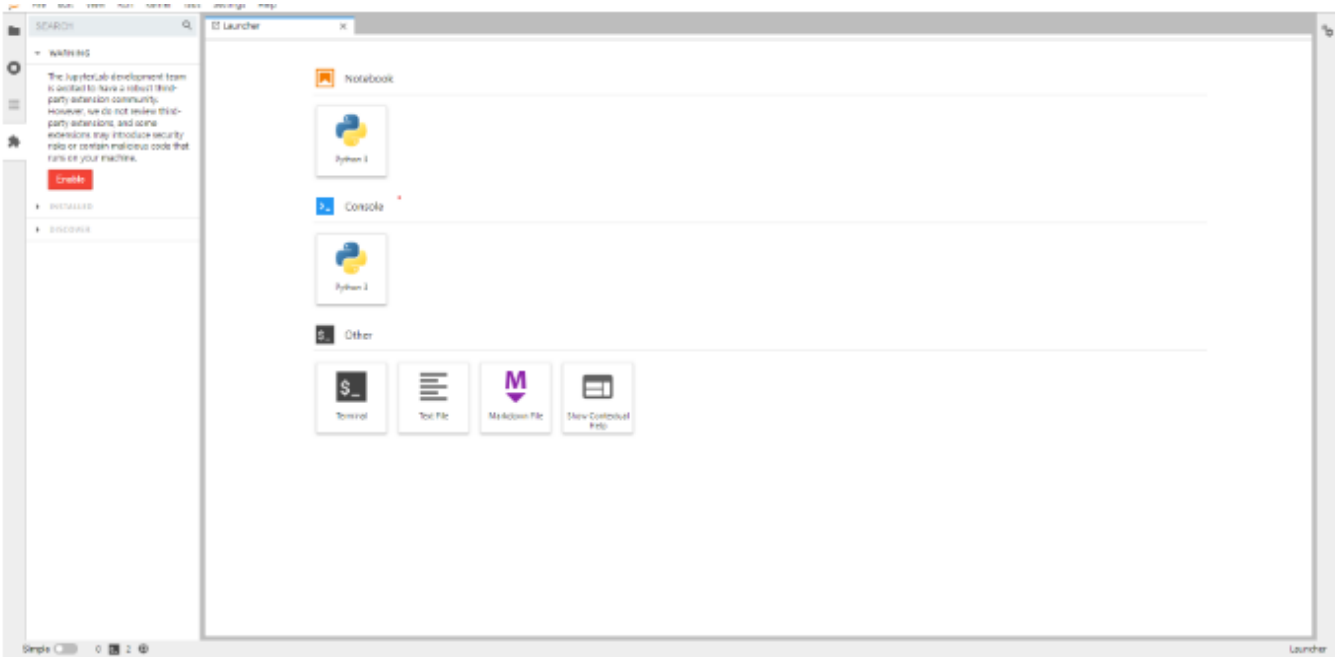


Рисунок 4.3 – Середовище розробки Jupyter Notebook

За допомогою Anaconda Navigator, графічного інтерфейсу для управління середовищем, користувачі можуть зручно керувати своїми проектами та запускати такі інструменти, як Jupyter Notebook або Spyder IDE, безпосередньо з інтерфейсу.

Загалом метою програмного середовища Anaconda є надання високопродуктивних і гнучких можливостей для роботи з даними та наукових обчислень у мові програмування Python.

4.2 Опис набору даних

Важливим етапом роботи методу є навчання відповідної моделі. Для навчання моделі запропонованого авторського методу було використано датасети «PolitiFact» та «LIAR».

PolitiFact – це організація, яка проводить перевірки фактів, новин та оцінює правдивість тверджень політиків та інших громадських джерел інформації у Сполучених Штатах Америки. Їхні результати часто використовуються для аналізу політичних висловлювань та визначення їхньої достовірності. Крім цього, дана організація надає доступ до свого публічного датасету, що має однойменну назву.

Цей датасет є набором політичних новин. Загальний розмір набору даних складає 6235 новин. Він включає такі дані:

- Number,
- Title,
- Text,
- Label.

Number – порядковий номер новини. Title – заголовок новини, зазвичай для повноцінної класифікації новини використання тільки заголовку є недостатнім, оскільки нейромережі недостатньо даних для визначення ваг та ознак, тому доцільнішим для навчання мережі є наступне поле «Text». Text – безпосередньо текст новини з детальним її описом. Label – дане поле має 2 значення «Real» та «Fake», воно використовується для класифікації новини.

Даний набір даних було розділено у відсотковому співвідношенні на дві частини: тестову та навчальну. Навчальний набір даних використовується для тренування нейронної мережі, а тестовий - безпосередньо для валідації результатів роботи моделі. Розмір навчальної вибірки даних склав 4988 новин, а тестової - 1247.

Крім того, для більш детальної валідації результатів роботи моделей методів було використано датасет «LIAR». Датасет «LIAR» є набором текстових даних, який включає інформацію про твердження, піддані факт-чекінгу за допомогою PolitiFact. У цьому датасеті містяться тексти, які мають відмітки достовірності або недостовірності на різних рівнях. Кожне твердження також може бути призначене до певної категорії, такої як політика, економіка, здоров'я тощо. Безпосередня розмітка датасету (набір його колонок) відповідає датасету «PolitiFact».

У таблиці 4.1 наведено детальний опис даних з числа проаналізованих вище наборів даних.

Таблиця 4.1 – Детальний опис використаних наборів даних

Набори даних	Кількість записів	Кількість достовірних новин	Кількість фейкових новин	Розмітка датасету
PolitiFact	6235	3741	2494	Number, Title, Text, Label
LIAR	5912	2792	3120	Number, Title, Text, Label

Отже, для навчання та валідації роботи моделі було використано кілька датасетів, що мають однакову розмітку.

З функціональної точки зору датасет «LIAR» було використано для того, щоб впевнитись, що дійсно додавання нового шару до архітектури нейронної мережі та збільшення розміру ядра одновимірного шару згорткового перетворення, призведе до того, що мережа може визначати головні ознаки новин, які не тільки зустрічались їй у межах навчального набору даних, а й у новинах, що зустрічаються мережі вперше.

4.3 Результати експериментів для порівняння моделей різних методів виявлення фейкових новин

Для оцінки ефективності запропонованого методу було проведено експерименти та порівняно їх результати для досліджуваних методів (запропонованого авторського, TensortFlow classification model, LogisticRegression) на описаних наборах даних («PolitiFact», «LIAR»).

З урахуванням даних п. 2.3 і пп. 4.1-4.2, для демонстрації ефективності авторської моделі прийнято рішення порівняння результатів експериментів здійснити на наступних метриках: загальна точність, f1-score [23], recall та precision [24]. Слід

зауважити, що для розрахунку загального значення метрик (M) f1-score, recall та precision була використана формула

$$M_{avg} = (M_{fake} + M_{real}) /2 , \quad (4.1)$$

де M_{fake} – це значення метрики для фейкових новин; M_{real} – це значення метрики для правдивих новин.

Результати навчання на наборі даних «PolitiFact».

Оскільки значення показників метрики можуть змінюватись від запуску до запуску, то для розрахунку середньої загальної точності було проведено 6 експериментальних запусків моделей на тестовому наборі даних. Для розрахунку середньої загальної точності використано формулу

$$A = \frac{\sum_{i=1}^{N} x_i}{N}, \quad (4.2)$$

де N – загальна кількість експериментів; x_i – значення відповідного показника; i – порядковий номер.

Отримавши середню точність можна розрахувати стандартне відхилення за наступною формулою:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(x_i - A)^2}{N}}. \quad (4.3)$$

Таким чином, значення стандартного відхилення буде відповідати похибці середнього значення, яку можна виразити як ± значення, тобто загальна точність = середня точність ± стандартне відхилення.

Результати експерименту №1

На рисунку 4.4 представлено графік залежності точності запропонованого авторського методу від кроків дослідження. При цьому загальна точність запропонованого методу на даному наборі даних склала 93,32%.

У таблиці 4.2 для вхідного набору даних експерименту 1 наведено результати функціонування досліджуваних моделей за різними метриками.

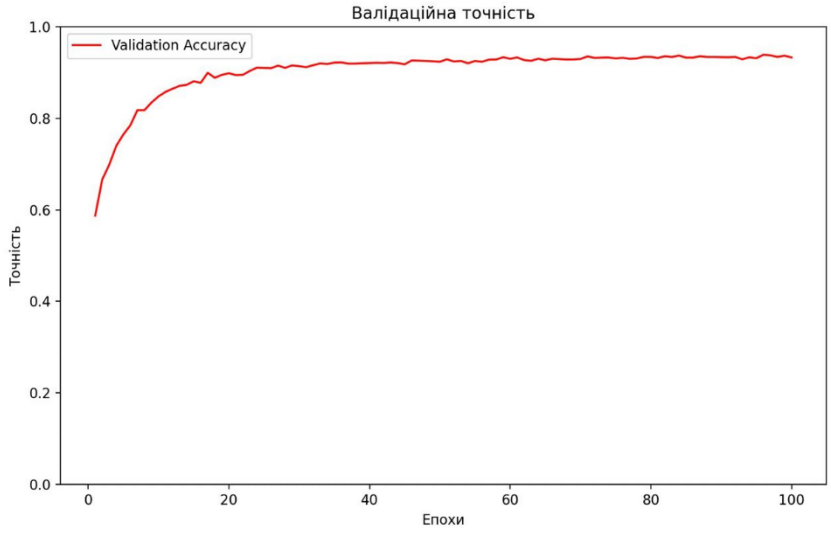


Рисунок 4.4 – Загальна точність на навчальному наборі даних «PolitiFact»

Таблиця 4.2 – Точнісні характеристики досліджуваних методів на наборі даних «PolitiFact»

Метод	Загальна точність	F1-score	recall	precision
Запропонований авторський	**93,32%**	**0,926**	0,919	**0,93**
TensortFlow classification model	89,37%	0,893	0,88	0,896
LogisticRegression	91,4%	0,914	**0,924**	0,916

При цьому приріст значень показників метрик можна оцінити з таблиці 4.3.

Таблиця 4.3 – Приріст значень показників метрик для запропонованого авторського методу у порівнянні з існуючими методами на наборі даних «PolitiFact»

Пари порівнюваних методів	**Загальна точність**	**F1-score**	**recall**	**precision**
Запропонований авторський - TensortFlow	**3,95%**	**0,033**	**0,039**	**0,034**

classification model				
Запропонований авторський - LogisticRegression	**1,920%**	**0,012**	-0,005	**0,014**

Як випливає з таблиці 4.3, запропонований авторський метод демонструє кращі показники за всіма метриками у порівнянні з існуючим методом TensortFlow classification model. Також авторський метод демонструє кращі показники за всіма метриками крім метрики recall у порівнянні з існуючим методом LogisticRegression.

Результати експерименту №2

У таблиці 4.4 для вхідного набору даних експерименту 2 наведено результати функціонування досліджуваних моделей за різними метриками. При цьому загальна точність запропонованого методу на даному наборі даних склала 92,79%.

Таблиця 4.4 – Точнісні характеристики досліджуваних методів на наборі даних «PolitiFact»

Метод	Загальна точність	F1-score	recall	precision
Запропонований авторський	**92,79%**	0,918	**0,931**	0,921
TensortFlow classification model	88,97%	0,888	0,896	0,875
LogisticRegression	92,67%	**0,926**	0,924	**0,933**

При цьому приріст значень показників метрик можна оцінити з таблиці 4.5.

Таблиця 4.5 – Приріст значень показників метрик для запропонованого авторського методу у порівнянні з існуючими методами на наборі даних «PolitiFact»

Пари порівнюваних методів	**Загальна точність**	**F1-score**	**recall**	**precision**
Запропонований авторський - TensortFlow classification model	**3,82%**	**0,03**	**0,035**	**0,046**
Запропонований авторський - LogisticRegression	**0,120%**	-0,008	**0,007**	-0,012

Як випливає з таблиці 4.5, запропонований авторський метод демонструє кращі показники за всіма метриками у порівнянні з існуючим методом TensortFlow classification model. Також авторський метод демонструє кращу загальну точність і показник за метрикою recall у порівнянні з існуючим методом LogisticRegression.

Результати експерименту №3

У таблиці 4.6 для вхідного набору даних експерименту 3 наведено результати функціонування досліджуваних моделей за різними метриками. При цьому загальна точність запропонованого методу на даному наборі даних склала 93,15%.

Таблиця 4.6 – Точнісні характеристики досліджуваних методів на наборі даних «PolitiFact»

Метод	Загальна точність	F1-score	recall	precision
Запропонований авторський	**93,15%**	0,903	**0,942**	**0,939**
TensortFlow classification model	90,86%	**0,905**	0,914	0,899
LogisticRegression	91,02%	0,9	0,912	0,907

При цьому приріст значень показників метрик можна оцінити з таблиці 4.7.

Таблиця 4.7 – Приріст значень показників метрик для запропонованого авторського методу у порівнянні з існуючими методами на наборі даних «PolitiFact»

Пари порівнюваних методів	**Загальна точність**	**F1-score**	**recall**	**precision**
Запропонований авторський - TensortFlow classification model	**2,29%**	-0,002	**0,028**	**0,04**
Запропонований авторський - LogisticRegression	**2,130%**	**0,003**	**0,030**	**0,032**

Як випливає з таблиці 4.7, запропонований авторський метод демонструє кращі показники за всіма метриками крім F1-score у порівнянні з існуючим методом TensortFlow classification model. Також авторський метод демонструє кращі показники за всіма метриками у порівнянні з існуючим методом LogisticRegression.

Результати експерименту №4

У таблиці 4.8 для вхідного набору даних експерименту 4 наведено результати функціонування досліджуваних моделей за різними метриками. При цьому загальна точність запропонованого методу на даному наборі даних склала 94,29%.

Таблиця 4.8 – Точнісні характеристики досліджуваних методів на наборі даних «PolitiFact»

Метод	Загальна точність	F1-score	recall	precision
Запропонований авторський	**94,29%**	**0,941**	**0,937**	**0,94**

TensortFlow classification model	89,74%	0,897	0,901	0,897
LogisticRegression	92,37%	0,923	0,919	0,921

При цьому приріст значень показників метрик можна оцінити з таблиці 4.9.

Таблиця 4.9 – Приріст значень показників метрик для запропонованого авторського методу у порівнянні з існуючими методами на наборі даних «PolitiFact»

Пари порівнюваних методів	**Загальна точність**	**F1-score**	**recall**	**precision**
Запропонований авторський - TensortFlow classification model	**4,55%**	**0,044**	**0,036**	**0,043**
Запропонований авторський - LogisticRegression	**1,920%**	**0,018**	**0,018**	**0,019**

Як випливає з таблиці 4.9, запропонований авторський метод демонструє кращі показники за всіма метриками у порівнянні як з існуючим методом TensortFlow classification model, так і в порівнянні з існуючим методом LogisticRegression.

Результати експерименту №5

У таблиці 4.10 для вхідного набору даних експерименту 5 наведено результати функціонування досліджуваних моделей за різними метриками. При цьому загальна точність запропонованого методу на даному наборі даних склала 94,34%.

Таблиця 4.10 – Точнісні характеристики досліджуваних методів на наборі даних «PolitiFact»

Метод	Загальна точність	F1-score	recall	precision
Запропонований авторський	**94,34%**	**0,945**	**0,941**	**0,942**
TensortFlow classification model	88,86%	0,883	0,896	0,885
LogisticRegression	92,52%	0,925	0,933	0,919

При цьому приріст значень показників метрик можна оцінити з таблиці 4.11.

Таблиця 4.11 – Приріст значень показників метрик для запропонованого авторського методу у порівнянні з існуючими методами на наборі даних «PolitiFact»

Пари порівнюваних методів	**Загальна точність**	**F1-score**	**recall**	**precision**
Запропонований авторський - TensortFlow classification model	**5,48%**	**0,062**	**0,045**	**0,057**
Запропонований авторський - LogisticRegression	**1,820%**	**0,020**	**0,008**	**0,023**

Як випливає з таблиці 4.11, запропонований авторський метод демонструє кращі показники за всіма метриками у порівнянні як з існуючим методом TensortFlow classification model, так і в порівнянні з існуючим методом LogisticRegression.

Результати експерименту №6

У таблиці 4.12 для вхідного набору даних експерименту 6 наведено результати функціонування досліджуваних моделей за різними метриками. При цьому загальна точність запропонованого методу на даному наборі даних склала 91,42%.

Таблиця 4.12 – Точнісні характеристики досліджуваних методів на наборі даних «PolitiFact»

Метод	Загальна точність	F1-score	recall	precision
Запропонований авторський	91,42%	**0,916**	**0,914**	**0,914**
TensortFlow classification model	88,84%	0,889	0,881	0,882
LogisticRegression	**91,56%**	0,915	0,911	**0,914**

При цьому приріст значень показників метрик можна оцінити з таблиці 4.13.

Таблиця 4.13 – Приріст значень показників метрик для запропонованого авторського методу у порівнянні з існуючими методами на наборі даних «PolitiFact»

Пари порівнюваних методів	**Загальна точність**	**F1-score**	**recall**	**precision**
Запропонований авторський - TensortFlow classification model	**2,58%**	**0,027**	**0,033**	**0,032**
Запропонований авторський - LogisticRegression	-0,140%	**0,001**	**0,003**	**0,000**

Як випливає з таблиці 4.13, запропонований авторський метод демонструє кращі показники за всіма метриками у порівнянні з існуючим методом TensortFlow classification model. Також авторський метод демонструє кращі показники за всіма метриками крім загальної точності у порівнянні з існуючим методом LogisticRegression.

За результатами узагальнення даних експериментів 1-6 можна зробити висновок, що середня точність запропонованого методу складає 93,22% (згідно із формулою (4.2)). З урахуванням (4.3), за результатами проведення відповідних обчислень встановлено, що стандартне відхилення складає 0,99%. Звідси можна зробити висновок про те, що запропонований метод проявляє себе достатньо ефективним на навчальному наборі даних «PolitiFact».

Однак, для більш детальної валідації роботи методу доцільно продіагностувати його на новому наборі даних «LIAR».

Результати навчання на наборі даних «LIAR».

Результати експерименту №1

На рисунку 4.5 представлено графік залежності точності запропонованого авторського методу від кроків дослідження. При цьому загальна точність запропонованого методу на даному наборі даних склала 91,36%.

У таблиці 4.14 для вхідного набору даних експерименту 1 наведено результати функціонування досліджуваних моделей за різними метриками.

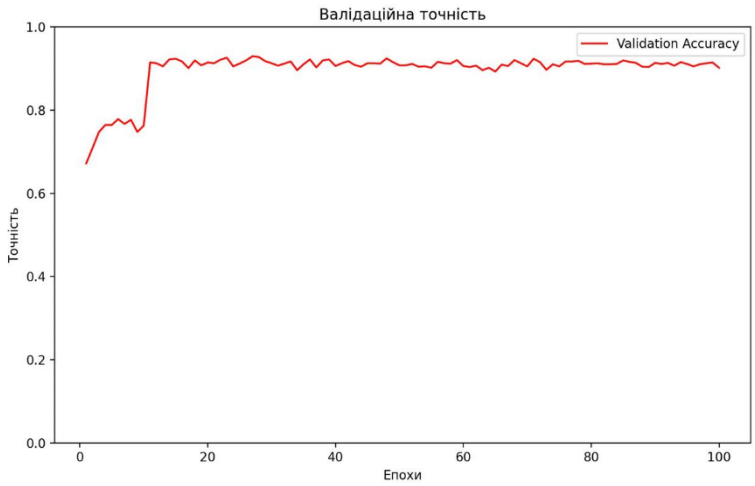


Рисунок 4.5 – Загальна точність на навчальному наборі даних «LIAR»

Таблиця 4.14 – Точнісні характеристики досліджуваних методів на наборі даних «LIAR»

Метод	Загальна точність	F1-score	recall	precision
Запропонований авторський	**91,36%**	**0,913**	0,902	**0,915**
TensortFlow classification model	89,11%	0,891	0,889	0,892
LogisticRegression	91,18%	0,911	**0,908**	**0,915**

При цьому приріст значень показників метрик можна оцінити з таблиці 4.15.

Таблиця 4.15 – Приріст значень показників метрик для запропонованого авторського методу у порівнянні з існуючими методами на наборі даних «LIAR»

Пари порівнюваних методів	**Загальна точність**	**F1-score**	**recall**	**precision**
Запропонований авторський - TensortFlow classification model	**2,25%**	**0,022**	**0,013**	**0,023**
Запропонований авторський - LogisticRegression	**0,18%**	**0,002**	-0,006	**0**

Як випливає з таблиці 4.15, запропонований авторський метод демонструє кращі показники за всіма метриками у порівнянні з існуючим методом TensortFlow classification model. Також авторський метод демонструє кращі показники за всіма метриками крім метрики recall у порівнянні з існуючим методом LogisticRegression.

Результати експерименту №2

У таблиці 4.16 для вхідного набору даних експерименту 2 наведено результати функціонування досліджуваних моделей за різними метриками. При цьому загальна точність запропонованого методу на даному наборі даних склала 91,34%.

Таблиця 4.16 – Точнісні характеристики досліджуваних методів на наборі даних «LIAR»

Метод	Загальна точність	F1-score	recall	precision
Запропонований авторський	**91,34%**	0,891	**0,917**	0,875
TensortFlow classification model	86,37%	0,811	0,821	0,829
LogisticRegression	91,21%	**0,896**	0,893	**0,895**

При цьому приріст значень показників метрик можна оцінити з таблиці 4.17.

Таблиця 4.17 – Приріст значень показників метрик для запропонованого авторського методу у порівнянні з існуючими методами на наборі даних «LIAR»

Пари порівнюваних методів	**Загальна точність**	**F1-score**	**recall**	**precision**
Запропонований авторський - TensortFlow classification model	**4,97%**	**0,08**	**0,096**	**0,046**
Запропонований авторський - LogisticRegression	**0,13%**	-0,005	**0,024**	-0,02

Як випливає з таблиці 4.17, запропонований авторський метод демонструє кращі показники за всіма метриками у порівнянні з існуючим методом TensortFlow

classification model. Також авторський метод демонструє кращу загальну точність і показник за метрикою recall у порівнянні з існуючим методом LogisticRegression.

Результати експерименту №3

У таблиці 4.18 для вхідного набору даних експерименту 3 наведено результати функціонування досліджуваних моделей за різними метриками. При цьому загальна точність запропонованого методу на даному наборі даних склала 91,76%.

Таблиця 4.18 – Точнісні характеристики досліджуваних методів на наборі даних «LIAR»

Метод	Загальна точність	F1-score	recall	precision
Запропонований авторський	**91,76%**	0,897	**0,923**	**0,901**
TensortFlow classification model	87,93%	**0,912**	0,904	0,813
LogisticRegression	89,62%	0,845	0,879	0,899

При цьому приріст значень показників метрик можна оцінити з таблиці 4.19.

Таблиця 4.19 – Приріст значень показників метрик для запропонованого авторського методу у порівнянні з існуючими методами на наборі даних «LIAR»

Пари порівнюваних методів	**Загальна точність**	**F1-score**	**recall**	**precision**
Запропонований авторський - TensortFlow classification model	**3,83%**	-0,015	**0,019**	**0,088**

Запропонований авторський - LogisticRegression	**2,14%**	**0,052**	**0,044**	**0,002**

Як випливає з таблиці 4.19, запропонований авторський метод демонструє кращі показники за всіма метриками крім F1-score у порівнянні з існуючим методом TensortFlow classification model. Також авторський метод демонструє кращі показники за всіма метриками у порівнянні з існуючим методом LogisticRegression.

Результати експерименту №4

У таблиці 4.20 для вхідного набору даних експерименту 4 наведено результати функціонування досліджуваних моделей за різними метриками. При цьому загальна точність запропонованого методу на даному наборі даних склала 92,47%.

Таблиця 4.20 – Точнісні характеристики досліджуваних методів на наборі даних «LIAR»

Метод	Загальна точність	F1-score	recall	precision
Запропонований авторський	**92,47%**	**0,912**	**0,917**	**0,923**
TensortFlow classification model	88,84%	0,873	0,874	0,877
LogisticRegression	91,43%	0,901	0,901	0,911

При цьому приріст значень показників метрик можна оцінити з таблиці 4.21.

Таблиця 4.21 – Приріст значень показників метрик для запропонованого авторського методу у порівнянні з існуючими методами на наборі даних «LIAR»

Пари порівнюваних методів	**Загальна точність**	**F1-score**	**recall**	**precision**

Запропонований авторський - TensortFlow classification model	**3,63%**	**0,039**	**0,043**	**0,046**
Запропонований авторський - LogisticRegression	**1,04%**	**0,011**	**0,016**	**0,012**

Як випливає з таблиці 4.21, запропонований авторський метод демонструє кращі показники за всіма метриками у порівнянні як з існуючим методом TensortFlow classification model, так і в порівнянні з існуючим методом LogisticRegression.

Результати експерименту №5

У таблиці 4.22 для вхідного набору даних експерименту 5 наведено результати функціонування досліджуваних моделей за різними метриками. При цьому загальна точність запропонованого методу на даному наборі даних склала 92,34%.

Таблиця 4.22 – Точнісні характеристики досліджуваних методів на наборі даних «LIAR»

Метод	Загальна точність	F1-score	recall	precision
Запропонований авторський	**92,34%**	**0,924**	**0,932**	**0,912**
TensortFlow classification model	88,23%	0,854	0,819	0,863
LogisticRegression	91,67%	0,874	0,846	0,886

При цьому приріст значень показників метрик можна оцінити з таблиці 4.23.

Таблиця 4.23 – Приріст значень показників метрик для запропонованого авторського методу у порівнянні з існуючими методами на наборі даних «LIAR»

Пари порівнюваних методів	**Загальна точність**	**F1-score**	**recall**	**precision**
Запропонований авторський - TensortFlow classification model	**4,11%**	**0,07**	**0,113**	**0,049**
Запропонований авторський - LogisticRegression	**0,67%**	**0,05**	**0,086**	**0,026**

Як випливає з таблиці 4.23, запропонований авторський метод демонструє кращі показники за всіма метриками у порівнянні як з існуючим методом TensortFlow classification model, так і в порівнянні з існуючим методом LogisticRegression.

Результати експерименту №6

У таблиці 4.24 для вхідного набору даних експерименту 6 наведено результати функціонування досліджуваних моделей за різними метриками. При цьому загальна точність запропонованого методу на даному наборі даних склала 90,12%.

Таблиця 4.24 – Точнісні характеристики досліджуваних методів на наборі даних «LIAR»

Метод	Загальна точність	F1-score	recall	precision
Запропонований авторський	90,12%	**0,911**	**0,893**	**0,901**
TensortFlow classification model	87,69%	0,843	0,854	0,877
LogisticRegression	**90,43%**	0,899	0,872	**0,911**

При цьому приріст значень показників метрик можна оцінити з таблиці 4.25.

Таблиця 4.25 – Приріст значень показників метрик для запропонованого авторського методу у порівнянні з існуючими методами на наборі даних «LIAR»

Пари порівнюваних методів	**Загальна точність**	**F1-score**	**recall**	**precision**
Запропонований авторський - TensortFlow classification model	**2,43%**	**0,068**	**0,039**	**0,024**
Запропонований авторський - LogisticRegression	-0,31%	**0,012**	**0,021**	-0,01

Як випливає з таблиці 4.25, запропонований авторський метод демонструє кращі показники за всіма метриками у порівнянні з існуючим методом TensortFlow classification model. Також авторський метод демонструє кращі показники за всіма метриками крім загальної точності та precision у порівнянні з існуючим методом LogisticRegression.

За результатами узагальнення даних експериментів 1-6 можна зробити висновок, що середня точність запропонованого методу складає 91,57% (згідно із формулою (4.2)). З урахуванням (4.3), за результатами проведення відповідних обчислень встановлено, що стандартне відхилення складає 0,78%. Звідси можна зробити висновок про те, що запропонований метод проявляє себе достатньо ефективним на навчальному наборі даних «LIAR».

При цьому, на новому наборі даних «LIAR» значення загальної точності для запропонованого авторського методу є дещо нижчим, ніж на навчальному наборі даних «PolitiFact», але аналогічні погіршення результатів роботи можна спостерігати і для існуючих моделей з числа досліджуваних.

Таким чином, запропонований авторський метод загалом проявляє себе більш ефективним у порівнянні з існуючими методами з числа досліджуваних. Отже, його можна застосовувати, як для встановлення достовірності новин, так і для їх

класифікації у режимі реального часу.

Висновки до розділу 4

У ході опрацювання четвертого розділу проведено оцінку ефективності запропонованого авторського методу виявлення фейкових новин нейромережевими засобами. В якості моделей, що порівнювались, виступали запропонований метод з удосконаленою архітектурою та існуючі моделі TensortFlow classification model і LogisticRegression. Обгрунтовано доцільність використання програмного середовища «Anaconda» для полегшення проведення порівняльної оцінки (експериментів) та уникнення необхідності адаптації кожної моделі під системні параметри експериментальної електронно-обчислювальної машини. Наведено опис можливостей цього програмного середовища.

Обгрунтовано доцільність використання датасетів «PolitiFact» та «LIAR» для навчання моделі запропонованого авторського методу.

Для оцінки ефективності запропонованого методу проведено експерименти та порівняно їх результати для запропонованого авторського методу, методів TensortFlow classification model і LogisticRegression на наборах даних «PolitiFact», «LIAR». За результатами проведених експериментів встановлено, що для запропонованого авторського методу на наборі даних «PolitiFact» середня загальна точність становить 93,22%, а стандартне відхилення - 0,99%. На наборі даних «LIAR» середня загальна точність становить 91,57%, а стандартне відхилення складає 0,78%. При цьому, запропонований авторський метод загалом проявляє себе більш ефективним у порівнянні з існуючими методами з числа досліджуваних. Отже, його можна застосовувати, як для встановлення достовірності новин, так і для їх класифікації у режимі реального часу.

Загальні висновки

Під час виконання кваліфікаційної роботи магістра розроблено метод виявлення в Інтернеті фейкових новин нейромережевими засобами.

При розробці методу опрацьовано наступні задачі:

– проведено аналіз існуючих моделей і підходів для виявлення в Інтернеті фейкових новин. У ході аналізу досліджено базові алгоритми виявлення фейкових новин, до яких віднесено алгоритми машинного та глибокого навчання, а також алгоритми виявлення фейкових новин, що базуються на використанні результатів аналізу сентименту новинного контенту й емоцій у коментарях користувачів;

– розроблено метод виявлення в Інтернеті фейкових новин нейромережевими засобами. При цьому, для отримання більш достовірної класифікації новин на наборах даних, що не були присутніми у навчальній вибірці, запропоновано підхід до вдосконалення структури багатошарової CNN нейромережі. Суть удосконалення стосується додавання шару випадкового відключення (Dropout layer), збільшення розміру ядра, зміни функції активації одновимірного шару максимального об’єднання з сигмоїдальної функції активації на функцію активації ReLU;

– підготовлено набір даних для навчання нейронної мережі. Обгрунтовано доцільність використання для цього датасетів «PolitiFact» та «LIAR»;

– проведено навчання нейронної мережі виявляти фейкові новини;

– розроблено інформаційну систему, що реалізує запропонований метод. Для розробки backend частини додатку обрано мову програмування PHP та фреймворк даної мови програмування Laravel. Для frontend частини додатку обрано фреймворк Vue.js, Для реалізації стилістичної складової веб-застосунку обрано фреймворк мови CSS – Tailwind CSS. Безпосередньо для реалізації методу виявлення фейкових новин було обрано мову програмування Python та бібліотеки TensorFlow, NumPy, Scikit-learn для роботи з нейромережевими засобами;

– проведено оцінку ефективності запропонованого авторського методу виявлення в Інтернеті фейкових новин. В якості моделей, що порівнювались, виступали запропонований метод з удосконаленою архітектурою та існуючі моделі TensortFlow classification model і LogisticRegression. Обгрунтовано доцільність використання програмного середовища «Anaconda» для полегшення проведення порівняльної оцінки. За результатами проведених експериментів встановлено, що для запропонованого авторського методу на наборі даних «PolitiFact» середня загальна точність становить 93,22%, а стандартне відхилення - 0,99%. На наборі даних «LIAR» середня загальна точність становить 91,57%, а стандартне відхилення складає 0,78%. При цьому, запропонований авторський метод загалом проявляє себе більш ефективним у порівнянні з існуючими методами з числа досліджуваних. Отже, його можна застосовувати, як для встановлення достовірності новин, так і для їх класифікації у режимі реального часу.

Перелік посилань

1. Bondielli, A.; Marcelloni, F. A survey on fake news and rumour detection techniques. Inf. Sci. 2019, 497, 38–55. [CrossRef]

2. Islam, M.R.; Liu, S.; Wang, X.; Xu, G. Deep learning for misinformation detection on online social networks: A survey and new perspectives. Soc. Netw. Anal. Min. 2020, 10, 82. [CrossRef] [PubMed]

3. Bahad, P.; Saxena, P.; Kamal, R. Fake News Detection using Bi-directional LSTM-Recurrent Neural Network. Procedia Comput. Sci. 2019, 165, 74–82. [CrossRef]

4. Machová, K.; Mach, M.; Porezaný, M. Deep Learning in the Detection of Disinformation about COVID-19 in Online Space. Sensors 2022, 22, 9319. [CrossRef]

5. Liu, Y.; Wu, Y.-F.B. Fned: A deep network for fake news early detection on social media. ACM Trans. Inf. Syst. (TOIS) 2020, 38, 1–33. [CrossRef]

6. Zhou, X.; Jain, A.; Phoha, V.V.; Zafarani, R. Fake News Early Detection: An Interdisciplinary Study. arXiv 2019, arXiv:1904.11679.

7. DataReportal. Digital 2021 Global Digital Overview; DataReportal: Singapore, 2021.

8. Kwak, H.; Lee, C.; Park, H.; Moon, S. What is Twitter, a Social Network or a News Media? In Proceedings of the 19th International Conference on World Wide Web; Association for Computing Machinery: New York, NY, USA, 2010; pp. 591–600. [CrossRef]

9. Friggeri, A.; Adamic, L.; Eckles, D.; Cheng, J. Rumor cascades. In Proceedings of the International AAAI Conference on Web and Social Media, Ann Arbor, MI, USA, 1–4 June 2014; Volume 8.

10. Zhou, X.; Zafarani, R. A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities. ACM Comput. Surv. 2020, 53, 1–40. [CrossRef]

11. Conroy, N.K.; Rubin, V.L.; Chen, Y. Automatic deception detection: Methods for finding fake news. Proc. Assoc. Inf. Sci. Technol. 2015, 52, 1–4. [CrossRef]

12. Habib, A.; Asghar, M.Z.; Khan, A.; Habib, A.; Khan, A. False information detection in online content and its role in decision making: A systematic literature review. Soc. Netw. Anal. Min. 2019, 9, 50. [CrossRef]

13. Sansonetti, G.; Gasparetti, F.; D'aniello, G.; Micarelli, A. Unreliable Users Detection in Social Media: Deep Learning Techniques for Automatic Detection. IEEE Access 2020, 8, 213154–213167. [CrossRef]

14. Vosoughi, S.; Roy, D.; Aral, S. The spread of true and false news online. Science 2018, 359, 1146–1151. [CrossRef]

15. Zhao, Z.; Resnick, P.; Mei, Q. Enquiring Minds: Early Detection of Rumors in Social Media from Enquiry Posts. In Proceedings of the 24th International Conference on World Wide Web; International World Wide Web Conferences Steering Committee: Geneva, Switzerland, 2015; pp. 1395–1405. [CrossRef]

16. Kumar, S.; West, R.; Leskovec, J. Disinformation on the web: Impact, characteristics, and detection of wikipedia hoaxes. In Proceedings of the 25th International Conference on World Wide Web, Montreal, QC, Canada, 11–15 April 2016; pp. 591–602.

17. Potthast, M.; Kiesel, J.; Reinartz, K.; Bevendorff, J.; Stein, B. A stylometric inquiry into hyperpartisan and fake news. arXiv 2017, arXiv:1702.05638.

18. Tchakounté, F.; Calvin, K.A.; Ari, A.A.A.; Mbogne, D.J.F.J.J.o.K.S.U.-C.; Sciences, I. A smart contract logic to reduce hoax propagation across social media. J. King Saud Univ.-Comput. Inf. Sci. 2020, 34, 3070–3078. [CrossRef]

19. Rath, B.; Gao, W.; Ma, J.; Srivastava, J. Utilizing computational trust to identify rumor spreaders on Twitter. Soc. Netw. Anal. Min. 2018, 8, 64. [CrossRef]

20. Vosoughi, S.; Roy, D.; Aral, S. The spread of true and false news online. Science 2018, 359, 1146–1151. [CrossRef]

21. Al-Rakhami, M.S.; Al-Amri, A.M. Lies Kill, Facts Save: Detecting COVID-19 Misinformation in Twitter. IEEE Access 2020, 8, 155961–155970. [CrossRef]

22. Alkhodair, S.A.; Ding, S.H.; Fung, B.C.; Liu, J. Detecting breaking news rumors of emerging topics in social media. Inf. Process. Manag. 2020, 57, 102018. [CrossRef]

23. Zubair, T.; Raquib, A.; Qadir, J. Combating Fake News, Misinformation, and Machine Learning Generated Fakes: Insight's from the Islamic Ethical Tradition. ICR J. 2019, 10, 189–212. [CrossRef]

24. Allcott, H.; Gentzkow, M. Social media and fake news in the 2016 election. J. Econ. Perspect. 2017, 31, 211–236. [CrossRef]

25. Shu, K.; Sliva, A.; Wang, S.; Tang, J.; Liu, H. Fake News Detection on Social Media: A Data Mining Perspective. arXiv 2017, arXiv:1708.01967.

26. Trends, G. "Fake News—Explore—Google Trends". Available online: https://trends.google.com/trends/explore?date=2010-01- 01%202022-07-14&q=%22fake%20news%22 (accessed on 20 July 2022).

27. Langin, K. Fake news spreads faster than true news on Twitter—Thanks to people, not bots. Sci. Mag. 2018. Available online: https://www.science.org/content/article/fake-news-spreads-faster-true-news-twitter-thanks-people-not-bots (accessed on 20 February 2022).

28. Zubiaga, A.; Aker, A.; Bontcheva, K.; Liakata, M.; Procter, R. Detection and Resolution of Rumours in Social Media: A Survey. ACM Comput. Surv. 2018, 51, 1–36. [CrossRef]

29. Evolvi, G. Hate in a tweet: Exploring internet-based islamophobic discourses. Religions 2018, 9, 307. [CrossRef]

30. Al-Makhadmeh, Z.; Tolba, A. Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach. Computing 2020, 102, 501–522. [CrossRef]

31. Feng, S.; Banerjee, R.; Choi, Y. Syntactic Stylometry for Deception Detection. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers); Association for Computational Linguistics: Jeju Island, Republic of Korea, 2012; pp. 171–175.

32. de Oliveira, N.R.; Medeiros, D.S.; Mattos, D.M. A sensitive stylistic approach to identify fake news on social networking. IEEE Signal Process. Lett. 2020, 27, 1250–1254. [CrossRef]

33. Zhou, X.; Jain, A.; Phoha, V.V.; Zafarani, R. Fake news early detection: A theory-driven model. Digit. Threat. Res. Pract. 2020, 1, 1–25. [CrossRef]

34. Lin, L.; Chen, Z. Social rumor detection based on multilayer transformer encoding blocks. Concurr. Comput. Pract. Exp. 2021, 33, e6083. [CrossRef]

35. Wang, W.Y. “Liar, Liar Pants on Fire”: A New Benchmark Dataset for Fake News Detection. arXiv 2017, arXiv:1705.00648.

36. Patwa, P.; Sharma, S.; Pykl, S.; Guptha, V.; Kumari, G.; Akhtar, M.S.; Ekbal, A.; Das, A.; Chakraborty, T. Fighting an infodemic: COVID-19 fake news dataset. In Proceedings of the International Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situation; Springer: Berlin/Heidelberg, Germany, 2021; pp. 21–29.

37. Eke, C.I.; Norman, A.A.; Shuib, L.; Nweke, H.F. Sarcasm identification in textual data: Systematic review, research challenges and open directions. Artif. Intell. Rev. 2020, 53, 4215–4258. [CrossRef]

38. Alonso, M.A.; Vilares, D.; Gómez-Rodríguez, C.; Vilares, J. Sentiment analysis for fake news detection. Electronics 2021, 10, 1348. [CrossRef]

39. Elhadad, M.K.; Li, K.F.; Gebali, F. Detecting Misleading Information on COVID-19. IEEE Access 2020, 8, 165201–165215. [CrossRef]

40. Alghamdi, O.; Lin, Y.; Luo, S.: A comparative study of machine learning and deep learning techniques for fake news detection. Information 2022, 13, 576, 28 p.

41. Guo, M.; Xu, Z.; Liu, L.; Guo, M.; Zhang, Y. An Adaptive Deep Transfer Learning Model for Rumor Detection without Sufficient Identified Rumors. Math. Probl. Eng. 2020, 2020, 7562567. [CrossRef]

42. Varshney, D.; Vishwakarma, D.K. Vishwakarma, Hoax news-inspector: A real-time prediction of fake news using content resemblance over web search results for authenticating the credibility of news articles. J. Ambient Intell. Humaniz. Comput. 2020, 12, 8961–8974. [CrossRef]

43. Kim, Y.; Kim, H.K.; Kim, H.; Hong, J.B. Do Many Models Make Light Work? Evaluating Ensemble Solutions for Improved Rumor Detection. IEEE Access 2020, 8, 150709–150724. [CrossRef]

44. Yaakub, M.R.; Latiffi, M.I.A.; Zaabar, L.S. A review on sentiment analysis techniques and applications. IOP Conf. Ser. Mater. Sci. Eng. 2019, 551, 012070. [CrossRef]

45. Santhoshkumar, S.; Babu, L.D. Earlier detection of rumors in online social networks using certainty-factor-based convolutional neural networks. Soc. Netw. Anal. Min. 2020, 10, 20. [CrossRef]

46. Tian, L.; Zhang, X.; Wang, Y.; Liu, H. Early detection of rumours on twitter via stance transfer learning. In Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, 14–17 April 2020, Proceedings, Part I 42; Springer: Cham, Switzerland, 2020; Volume 12035, p. 575.

47. Albahar, M. A hybrid model for fake news detection: Leveraging news content and user comments in fake news. IET Inf. Secur. 2021, 15, 169–177. [CrossRef]

48. Ghanem, B.; Rosso, P.; Rangel, F. An emotional analysis of false information in social media and news articles. ACM Trans. Internet Technol. (TOIT) 2020, 20, 1–18. [CrossRef]

49. Kumari, R.; Ashok, N.; Ghosal, T.; Ekbal, A. What the fake? Probing misinformation detection standing on the shoulder of novelty and emotion. Inf. Process. Manag. 2022, 59, 102740. [CrossRef]

50. Zhang, X.; Cao, J.; Li, X.; Sheng, Q.; Zhong, L.; Shu, K. Mining dual emotion for fake news detection. In Proceedings of the WWW ’21: The Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 3465–3476.

51. Zimbra, D.; Abbasi, A.; Zeng, D.; Chen, H. The state-of-the-art in Twitter sentiment analysis: A review and benchmark evaluation. ACM Trans. Manag. Inf. Syst. (TMIS) 2018, 9, 1–29. [CrossRef]

52. Feng, Z. Hot news mining and public opinion guidance analysis based on sentiment computing in network social media. Pers. Ubiquitous Comput. 2019, 23, 373–381. [CrossRef]

53. Imran, A.S.; Daudpota, S.M.; Kastrati, Z.; Batra, R. Cross-cultural polarity and emotion detection using sentiment analysis and deep learning on COVID-19 related tweets. IEEE Access 2020, 8, 181074–181090. [CrossRef]

54. Pota, M.; Ventura, M.; Catelli, R.; Esposito, M. An effective BERT-based pipeline for Twitter sentiment analysis: A case study in Italian. Sensors 2020, 21, 133. [CrossRef]

55. Dang, C.N.; Moreno-García, M.N.; Prieta, F.D.L. An approach to integrating sentiment analysis into recommender systems. Sensors 2021, 21, 5666. [CrossRef]

56. Li, Q.; Hu, Q.; Lu, Y.; Yang, Y.; Cheng, J. Multi-level word features based on CNN for fake news detection in cultural communication. Pers. Ubiquitous Comput. 2020, 24, 259–272.

57. Correia, F.; Madureira, A.M.; Bernardino, J. Deep Neural Networks Applied to Stock Market Sentiment Analysis. Sensors 2022, 22, 4409. [CrossRef] [PubMed]

58. Subramani, S.; Wang, H.; Vu, H.Q.; Li, G. Domestic violence crisis identification from facebook posts based on deep learning. IEEE Access 2018, 6, 54075–54085. [CrossRef]

59. Hochreiter, S.; Schmidhuber, J. Long short-term memory. Neural Comput. 1997, 9, 1735–1780. [CrossRef]

60. Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. arXiv 2014, arXiv:1406.1078.

61. Zhang, X.; Chen, F.; Huang, R. A combination of RNN and CNN for attention-based relation classification. Procedia Comput. Sci. 2018, 131, 911–917. [CrossRef]

62. Zhou, C.; Sun, C.; Liu, Z.; Lau, F. A C-LSTM neural network for text classification. arXiv 2015, arXiv:1511.08630.

63. Gururangan, S.; Marasovi´c, A.; Swayamdipta, S.; Lo, K.; Beltagy, I.; Downey, D.; Smith, N.A. Don’t stop pretraining: Adapt language models to domains and tasks. arXiv 2020, arXiv:2004.10964.

64. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv 2018, arXiv:1810.04805.

65. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. arXiv 2019, arXiv:1907.11692.

66. Khan, J.Y.; Khondaker, M.T.I.; Afroz, S.; Uddin, G.; Iqbal, A. A benchmark study of machine learning models for online fake news detection. Mach. Learn. Appl. 2021, 4, 100032. [CrossRef]

67. Horne, L.; Matti, M.; Pourjafar, P.; Wang, Z. GRUBERT: A GRU-Based Method to Fuse BERT Hidden Layers for Twitter Sentiment Analysis. In Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop; Association for Computational Linguistics: Suzhou, China, 2020; pp. 130–138

68. Nakamura, K.; Levy, S.; Wang, W.Y. r/fakeddit: A new multimodal benchmark dataset for fine-grained fake news detection. arXiv 2019, arXiv:1911.03854.

69. Ajao, O.; Bhowmik, D.; Zargari, S. Sentiment aware fake news detection on online social networks. In Proceedings of the ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 2507–2511.

70. Bhutani, B.; Rastogi, N.; Sehgal, P.; Purwar, A. Fake news detection using sentiment analysis. In Proceedings of the 2019 Twelfth International Conference on Contemporary Computing (IC3), Noida, India, 8–10 August 2019; pp. 1–5.

71. Giachanou, A.; Rosso, P.; Crestani, F. Leveraging emotional signals for credibility detection. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 877–880.

72. Kumar, S.; Asthana, R.; Upadhyay, S.; Upreti, N.; Akbar, M. Fake news detection using deep learning models: A novel approach. Trans. Emerg. Telecommun. Technol. 2020, 31, e3767. [CrossRef]

73. Kaliyar, R.K.; Kumar, P.; Kumar, M.; Narkhede, M.; Namboodiri, S.; Mishra, S. DeepNet: An efficient neural network for fake news detection using news-user

engagements. In Proceedings of the 2020 5th International Conference on Computing, Communication and Security (ICCCS), Patna, India, 14–16 October 2020; pp. 1–6.

74. Kirchknopf, A.; Slijepˇcevi´c, D.; Zeppelzauer, M. Multimodal Detection of Information Disorder from Social Media. In Proceedings of the 2021 International Conference on Content-Based Multimedia Indexing (CBMI), Lille, France, 28–30 June 2021; pp. 1–4.

75. Xie, J.; Liu, S.; Liu, R.; Zhang, Y.; Zhu, Y. SeRN: Stance extraction and reasoning network for fake news detection. In Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 2520–2524.

76. Raza, S.; Ding, C. Fake news detection based on news content and social contexts: A transformer-based approach. Int. J. Data Sci. Anal. 2022, 13, 335–362. [CrossRef]

77. [Електронний ресурс]. – Режим доступу: https://arxiv.org/abs/1412.6980

78. [Електронний ресурс]. – Режим доступу: https://robotdreams.cc/uk/blog/331-gradiyentniy-spusk-algoritm-ta-priklad-na-python)

79. [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Artificial_neuron

ДОДАТКИ

Додаток А

Світлини наукових публікацій, виконаних при роботі над кваліфікаційною роботою магістра

(ксерокопії титульної сторінки, сторінки змісту та всіх сторінок із публікацією)

Перелік наукових публікацій:

- Боровик Д. О. Актуальність задачі автоматизації виявлення фейкових новин і огляд підходів та інформаційних систем, що її реалізують // Матеріали Всеукраїнської науково-практичної Інтернет-конференції (13-19 березня 2023 року) «Автоматизація та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку» – Черкаси: ЧНУ, 2023. – С. 45-47;
- Боровик Л. В., Боровик Д. О. Підвищення інформаційної ефективності виявлення недостовірної інформації в Інтернеті // Збірник тез доповідей XIX Міжнародної науково-практичної конференції (10 листопада 2023 року) «Військова освіта і наука: сьогодення та майбутнє». - К.: ВІКНУ, 2023. – С. 23-24.
- Боровик Д. О., Бармак О. В. Удосконалений метод виявлення фейкових новин на основі використання CNN нейромережі // Науковий журнал «Вісник Хмельницького національного університету. Серія: Технічні науки» Хмельницький, 2023.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Черкаський національний університет
імені Богдана Хмельницького
Черкаський інститут банківської справи
Чорноморський державний університет імені Петра Могили

Всеукраїнська науково-практична
Інтернет-конференція

Автоматизація та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку

13-19 березня 2023 року

м. Черкаси

Автоматизація та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку: матеріали Всеукраїнської науково-практичної Internet-конференції. – Черкаси, 2023. - 196 с. – [Укр. мова.]

ПРОГРАМНИЙ КОМІТЕТ

Голова – **Черевко Олександр Володимирович**, доктор економічних наук, ректор Черкаського національного університету імені Богдана Хмельницького, Черкаси

Голуб Сергій Васильович – доктор технічних наук, професор кафедри програмного забезпечення автоматизованих систем, Черкаський державний технологічний університет

Гриценко Валерій Григорович – доктор педагогічних наук, доцент кафедри автоматизація та комп'ютерно-інтегрованих технологій Черкаського національного університету імені Богдана Хмельницького, Черкаси

Засядько Аліна Анатоліївна – доктор технічних наук, професор кафедри менеджменту та інформаційних технологій Черкаського інституту ДВНЗ «Університет банківської справи», Черкаси

Канашевич Георгій Вікторович – доктор технічних наук, професор, завідувач кафедри технології та обладнання машинобудівних виробництв Черкаського державного технологічного університету, Черкаси

Квасніков Володимир Павлович – доктор технічних наук, професор, завідувач кафедри комп'ютеризованих електротехнічних систем та технологій Національного авіаційного університету, Київ

Ладанюк Анатолій Петрович – доктор технічних наук, професор, заслужений діяч науки і техніки України, академік Міжнародної академії комп'ютерних наук і систем, Національний університет харчових технологій, Київ

Ляшенко Юрій Олексійович – доктор фізико-математичних наук, професор, директор навчально-наукового Інституту інформаційних та освітніх технологій Черкаського національного університету імені Богдана Хмельницького, Черкаси

Мусієнко Максим Павлович – доктор технічних наук, професор, професор кафедри автоматизації та комп'ютерно-інтегрованих технологій Черкаського національного університету імені Богдана Хмельницького, Черкаси

Осауленко Ігор Анатолійович – доктор технічних наук, доцент, завідувач кафедри інтелектуальних систем прийняття рішень

Черкаського національного університету імені Богдана Хмельницького, Черкаси

Прокопенко Тетяна Олександрівна – доктор технічних наук, завідувач кафедри інформаційних технологій проектування, Черкаський державний технологічний університет, Черкаси

Сергієнко Володимир Петрович – академік АН України, заслужений працівник освіти України, доктор педагогічних наук, професор, кафедра комп'ютерної інженерії факультету інформатики Національного педагогічного університету імені М.П. Драгоманова, Київ

Спірін Олег Михайлович – доктор педагогічних наук, професор, проректор з наукової роботи та цифровізації Університету менеджменту освіти НАПН України, Київ

Тесля Юрій Миколайович – доктор технічних наук, професор, Черкаський державний технологічний університет, Черкаси

Тітов В'ячеслав Андрійович – доктор технічних наук, професор, завідувач кафедри технології виробництва літальних апаратів НТУУ КПІ, Київ

Триус Юрій Васильович – доктор педагогічних наук, професор, завідувач кафедри комп'ютерних наук та системного аналізу Черкаського державного технологічного університету, Черкаси

ОРГАНІЗАЦІЙНИЙ КОМІТЕТ

Дідук Віталій Андрійович – кандидат технічних наук, доцент, завідувач кафедри автоматизації та комп'ютерно-інтегрованих технологій (голова)

Гриценко Валерій Григорович – доктор педагогічних наук, доцент

Луценко Галина Василівна – доктор педагогічних наук наук, доцент

Романенко Тетяна Василівна – доктор педагогічних наук, доцент

Гладка Людмила Іванівна – кандидат фізико-математичних наук, доцент

Кісіль Тетяна Юріївна, кандидат технічних наук, доцент

Красношлик Наталія Олександрівна – кандидат технічних наук, доцент

Піскун Олександр Варфоломійович – кандидат технічних наук, доцент

Подолян Оксана Миколаївна – кандидат фізико-математичних наук, доцент

Сердюк Олександр Анатолійович – кандидат економічних наук, доцент

Власенко Володимир Миколайович – старший викладач

Засядьвовк Наталія Олександрівна – викладач

Ожиндович Людмила Михайлівна – провідний фахівець

ТЕХНІЧНИЙ КОМІТЕТ

Поліщук Максим Миколайович.

Боровик Дмитро Олегович
Хмельницький національний університет, м. Хмельницький

АКТУАЛЬНІСТЬ ЗАДАЧІ АВТОМАТИЗАЦІЇ ВИЯВЛЕННЯ ФЕЙКОВИХ НОВИН І ОГЛЯД ПІДХОДІВ ТА ІНФОРМАЦІЙНИХ СИСТЕМ, ЩО ЇЇ РЕАЛІЗУЮТЬ

На сьогодні соціальні мережі стали основним джерелом новин у світі. Поширення фейкових новин у соціальних мережах є серйозною глобальною проблемою, яка завдає шкоди в різних сферах: політиці, економіці, військовій справі тощо. Фейкові новини негативно впливають на життя громадян, викликають негативні настрої, а реакція громадськості на них несе в собі емоції здивування, страху та огиди. Тому задача виявлення фейкових новин є актуальною, в цілому, а задача автоматизації такого виявлення є актуальною, зокрема.

Для вирішення останньої на даний час застосовується машинне навчання. При цьому підходи до виявлення фейкових новин поділяються на дві великі групи: з попереднім навчанням та з самонавчанням. Алгоритми першої групи потребують навчання та перевірки на двох окремих множинах вхідних даних, які дозволяють точно підібрати вагові коефіцієнти і забезпечують високу ефективність кінцевої системи. Алгоритми з самонавчанням не потребують окремого етапу навчання для забезпечення результату. Вони застосовуються тоді, коли ручна класифікація вхідних даних для навчання є дуже трудомістким завданням, а також коли потрібно, щоб система могла сама підлаштовуватися при зміні умов реального середовища застосування [1].

До недоліків цих підходів можна віднести таке.

Недостатня ефективність. На сьогоднішній день більшість систем виявлення фейкових новин базуються на аналізі ключових слів і їх контексту. Однак ці системи не завжди можуть ефективно виявляти фейкові новини, оскільки деякі автори можуть навмисно використовувати неконтрольовані ключові слова для підвищення рейтингу своїх матеріалів.

Відсутність стандартів. На даний час не існує загальноприйнятих стандартів, що ускладнює процес виявлення та

обробки фейкової інформації. Це може призводити до суб'єктивних рішень при визначенні фейкових новин та ризику поширення недостовірної інформації.

Залежність від джерела. Більшість систем виявлення фейкових новин використовують певні джерела, щоб оцінювати достовірність новин. Однак, це може призводити до спотворення даних і викривлення результатів, якщо саме джерело не є надійним або має власні інтереси.

З метою автоматизації виявлення фейкових новин створено ряд інформаційних систем. Серед найбільш поширених систем слід виокремити такі.

Factmata - система виявлення фейкових новин, яка використовує машинне навчання для аналізу текстів і виявлення неточностей та помилок. Factmata аналізує різні аспекти тексту, такі як тон, емоції, стиль, контекст і джерела інформації.

OpenAI GPT - система штучного інтелекту, яка використовує глибоке навчання для аналізу текстів та виявлення фейкових новин. OpenAI GPT може навчатися на великих обсягах текстової інформації і використовувати її для виявлення неточностей і помилок.

NewsGuard - система, яка використовує відгуки та оцінки експертів, щоб визначити достовірність новинних джерел. NewsGuard оцінює джерела за такими критеріями, як точність, розуміння контексту, збір доказів, а також розуміння етичних стандартів.

Fakebox - система виявлення фейкових новин, яка використовує машинне навчання для аналізу текстів та виявлення неточностей і помилок. Fakebox використовує штучну нейронну мережу, яка навчається на реальних прикладах фейкових новин.

Hoaxy - система, яка використовує аналіз соціальних мереж для виявлення фейкових новин. Hoaxy дозволяє відстежувати поширення новин у соціальних мережах і виявляти джерела та мережі, що поширюють фейкові новини.

Кожна з наведених систем має свої переваги та недоліки, які залежать від характеристик і специфіки використання систем [2-3].

Попередній аналіз загальних недоліків дозволяє зробити висновок про те, що на даний час залишається актуальним завдання підвищення ефективності інформаційних систем виявлення фейкових

новин, а також, що для підвищення ефективності існують потенційні можливості удосконалення окремих підходів.

Список використаних джерел

1. https://jrnl.nau.edu.ua/index.php/Infosecurity/article/view/14942/21990
2. https://www.mdpi.com/2078-2489/13/12/576
3. https://www.kaggle.com/code/therealsampat/fake-news-detection

Гавриш О.С.[1], к.ф.-м.н., доцент
Обруч Ю.Ю.[2], завідувач відділу комп'ютерно-технічних та телекомунікаційних досліджень
Баранов А.Д.[1], бакалавр
[1], бакалавр
1- Черкаський державний технологічний університет, Черкаси
2 - Черкаський науково-дослідний експертно-криміналістичний центр МВС України, Черкаси

СИСТЕМА ЗАХИСТУ ІНФОРМАЇ ДЛЯ ВІДОМЧОЇ НАУКОВО-ДОСЛІДНОЇ УСТАНОВИ

У науково-дослідному експертно-криміналістичному центрі (НДЕКЦ) зберігається і обробляється величезна кількість різних даних пов'язана з розробкою та впровадження в експертно-криміналістичну діяльність органів Міністерства внутрішніх справ України науково-практичних методів і засобів, спрямованих на боротьбу зі злочинністю [1]. Для забезпечення безпеки критичної інформації та інформаційних ресурсів в установі необхідно підтримувати в актуальному стані комплексну систему захисту інформації (КСЗІ) згідно вимогам нормативних документів [2].

Метою роботи є модернізація (підтримання в актуальному стані) КСЗІ в НДЕКЦ, яка дозволяє за допомогою організаційних, апаратно-технічних та програмних засобів, досягти максимальної ефективності захисту, що виключають несанкціонований доступ до інформації.

Попереднім етапом створення або модернізації КСЗІ є проведення аудиту стану інформаційно-телекомунікаційної системи (ІТС) та її складових частин. Під час аудиту проводиться аналіз нормативно-правових актів, які регламентують встановлення обмеження доступу до певних видів інформації, що обробляється, зберігається та передається в ІТС, визначається перелік інформації, що обробляється в ІТС, проводиться класифікація щодо необхідності

ЗМІСТ

Довідка: ВХНУ ТН 19/10/23

Видання: Вісник Хмельницького національного університету. Технічні науки

Категорія фаховості видання: фахове видання України, у якому можуть публікуватися результати дисертаційних робіт на здобуття наукових ступенів доктора наук, кандидата наук та ступеня доктора філософії, категорії «Б» філософії, категорії «Б» (наказ МОН №1643 від 28.12.2019, наказ МОН №409 від 17.03.2020).

Напрям – технічні науки за спеціальностями – 101, 121, 122, 123, 124, 125, 141, 151, 161, 172, 181, 182 (28.12.2019), спеціальності – 131, 132, 133 (17.03.2020)

Назва статті: УДОСКОНАЛЕНИЙ МЕТОД ВИЯВЛЕННЯ ФЕЙКОВИХ НОВИН НА ОСНОВІ ВИКОРИСТАННЯ CNN НЕЙРОМЕРЕЖІ

Автори: Дмитро БОРОВИК, Олександр БАРМАК (Хмельницький національний університет»)

Номер, у який прийнято статтю: №5 до друку рекомендовано буде до 25 жовтня 2023 року.

19.10.2023

Начальник відділу
інтелектуальної власності та трансферу технологій Ю.В.Кравчик

ВІДДІЛ КАДРІВ І.С.Мартинюк

Додаток Б

Слайди презентації

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Кафедра комп'ютерних наук

МЕТОД ВИЯВЛЕННЯ ФЕЙКОВИХ НОВИН ДЛЯ ОЦІНКИ ДОСТОВІРНОСТІ ДЖЕРЕЛ МАСОВОЇ ІНФОРМАЦІЇ

Виконав студент групи КНм-22-1:
Боровик Д. О.
Науковий керівник:
д.т.н., проф. Бармак О.В.

Актуальність роботи

- Зростання ролі онлайн соціальних мереж (ОСМ)
- Поширення новин в режимі реального часу незалежно від їх достовірності
- Поширення фейкових новин засобами ОСМ
- Стрімкий розвиток в останній період штучного інтелекту

МЕТА РОБОТИ

Розробка методу виявлення в Інтернеті фейкових новин нейромережевими засобами

Часткові завдання дослідження

- провести аналіз існуючих моделей і підходів для виявлення в Інтернеті фейкових новин
- розробити метод виявлення в Інтернеті фейкових новин нейромережевими засобами

- підготувати набір даних для навчання нейронної мережі
- провести навчання нейронної мережі виявляти фейкові новини

- розробити інформаційну систему, що реалізує запропонований метод
- оцінити отримані результати виявлення в Інтернеті фейкових новин запропонованим методом за загальними статистичними показниками

Об’єкт та предмет дослідження

Об'єкт

Об’єктом дослідження є процес виявлення в Інтернеті фейкових новин нейромережевими засобами

Предмет

Предметом дослідження є моделі нейронної мережі, методи виявлення в Інтернеті фейкових новин

Методи дослідження

- Метод застосування згорткової нейронної мережі - для виявлення в Інтернеті фейкових новин
- Методи сучасних інформаційних технологій - для програмної реалізації методу виявлення фейкових новин
- Методи вищої математики, теорії ймовірностей і математичної статистики - для дослідження ефективності методу виявлення фейкових новин

Наукова новизна

Наукова новизна одержаних результатів:

1. Розроблено метод виявлення в Інтернеті фейкових новин нейромережевими засобами

2. Розроблено інформаційну систему реалізації запропонованого методу виявлення в Інтернеті фейкових новин нейромережевими засобами

Публікації за матеріалами магістерської роботи

Матеріали наукових конференцій:

1. Боровик Д. О. Актуальність задачі автоматизації виявлення фейкових новин і огляд підходів та інформаційних систем, що її реалізують // Матеріали Всеукраїнської науково-практичної Інтернет-конференції (13-19 березня 2023 року) «Автоматизація та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку» – Черкаси: ЧНУ, 2023. – С. 45-47;
2. Боровик Л. В., Боровик Д. О. Підвищення інформаційної ефективності виявлення недостовірної інформації в Інтернеті // Збірник тез доповідей XIX Міжнародної науково-практичної конференції (10 листопада 2023 року) «Військова освіта і наука: сьогодення та майбутнє». - К.: ВІКНУ, 2023. – С. 23-24.

Публікація у фаховому науковому виданні:

1. Боровик Д. О., Бармак О. В. Удосконалений метод виявлення фейкових новин на основі використання CNN нейромережі // Науковий журнал «Вісник Хмельницького національного університету. Серія: Технічні науки» Хмельницький, 2023.

Завдання 1

Аналіз існуючих моделей і підходів для виявлення в Інтернеті фейкових новин

- проведено аналіз предметної області та існуючих публікацій щодо виявлення в Інтернеті фейкових новин
- здійснено функціональний аналіз відповідних базових алгоритмів
- здійснено постановку завдання та сформульовано часткові завдання дослідження

9

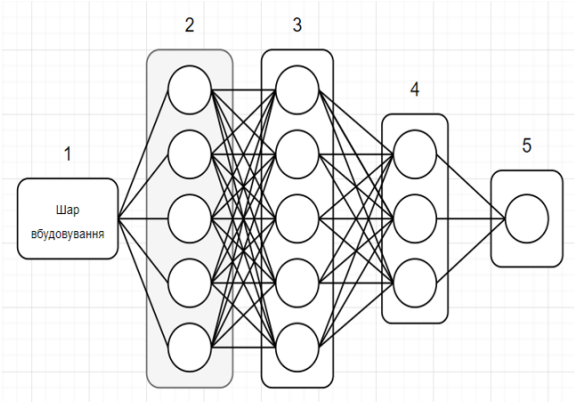
Завдання 2

Розробка методу виявлення в Інтернеті фейкових новин нейромережевими засобами

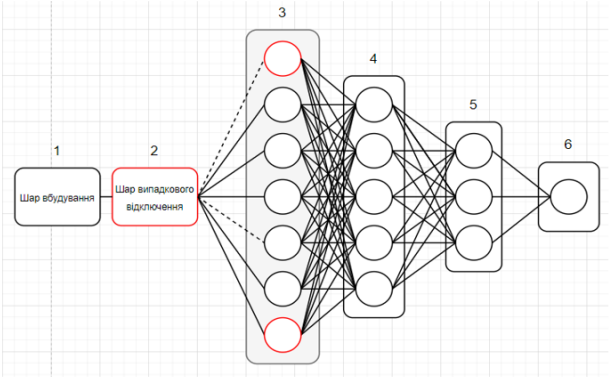
- здійснено аналіз того, що собою являють фейкові новини, якими вони бувають і які методи та прийоми боротьби з ними є ефективними
- проаналізовано можливі підходи до класифікації фейкових новин і встановлено відсутність єдиної класифікації
- здійснено оцінку методів виявлення фейкових новин з точки зору використання їх потенціалів для визначення авторського підходу до формування методу виявлення фейкових новин в Інтернеті на основі використання нейромережевих засобів;
- запропоновано авторський підхід до вдосконалення структури нейромережі для виявлення фейкових новин
- удосконалено архітектуру досліджуваної мережі для отримання більш достовірної класифікації новин на наборах даних, що не були присутніми у навчальній вибірці
- для оцінки якості навчання нейромережевих класифікаторів виявляти фейкові новини запропоновано використання статистичних метрик

10

СУТЬ УДОСКОНАЛЕННЯ СТРУКТУРИ НЕЙРОМЕРЕЖІ ДЛЯ ВИЯВЛЕННЯ ФЕЙКОВИХ НОВИН



Початкова структура нейронної мережі



Удосконалена архітектура нейромережі

СТАТИСТИЧНІ МЕТРИКИ ДЛЯ ОЦІНКИ ЯКОСТІ НАВЧАННЯ НЕЙРОМЕРЕЖЕВИХ КЛАСИФІКАТОРІВ ВИЯВЛЯТИ ФЕЙКОВІ НОВИНИ

Точність (A)

$$A_{ccuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

TP, TN, FP, FN - це, відповідно, справжні позитиви, справжні негативи, помилкові позитиви і помилкові негативи.

Точність (P)

$$P_{recision} = \frac{TP}{TP + FP}$$

Чутливість (R)

$$R_{ecall} = \frac{TP}{TP + FN}$$

$F1$ **- показник**

$$F1 = \frac{2 \cdot P_{recision} \cdot R_{ecall}}{P_{recision} + R_{ecall}} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$$

Міра AUC

$$AUC = \frac{1 - FPR + TPR}{2}$$

Завдання 3

Підготовка набору даних для навчання нейронної мережі

Завдання 4

Навчання нейронної мережі виявляти фейкові новини

- **Обгрунтовано доцільність використання датасетів «PolitiFact» та «LIAR» для навчання моделі запропонованого авторського методу**

13

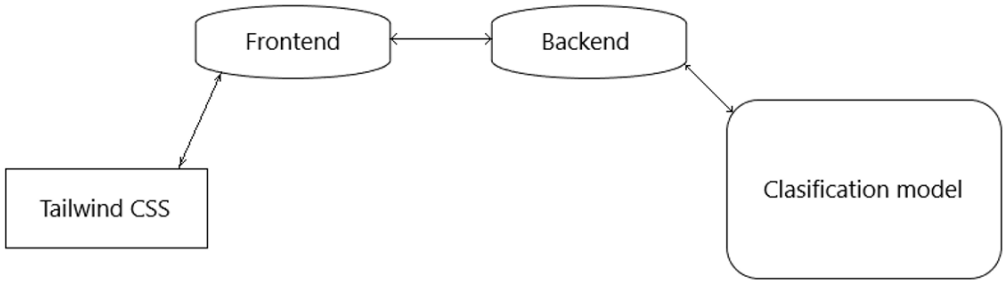
Завдання 5

Розробка інформаційної системи, що реалізує запропонований метод

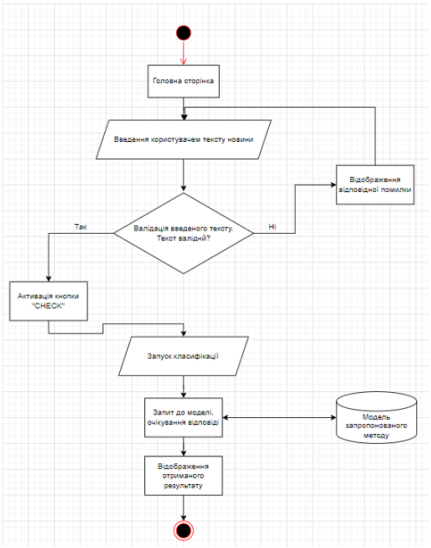
- обрано перелік технологій для програмної реалізації застосунку, що реалізує авторський метод виявлення фейкових новин на основі використання нейромережевих технологій, а також здійснено аналіз кожної з них
- сформовано функціональну схему застосунку та UML-діаграми, що пояснюють його структуру й описують взаємодію користувача з ним
- розроблено інформаційну систему, що реалізує запропонований метод
- проведено практичне тестування інформаційної системи на основі застосування розробленого набору тест-кейсів для перевірки як функціональності інтерфейсу, що взаємодіє з користувачем, так і функціональної частини веб-додатку

14

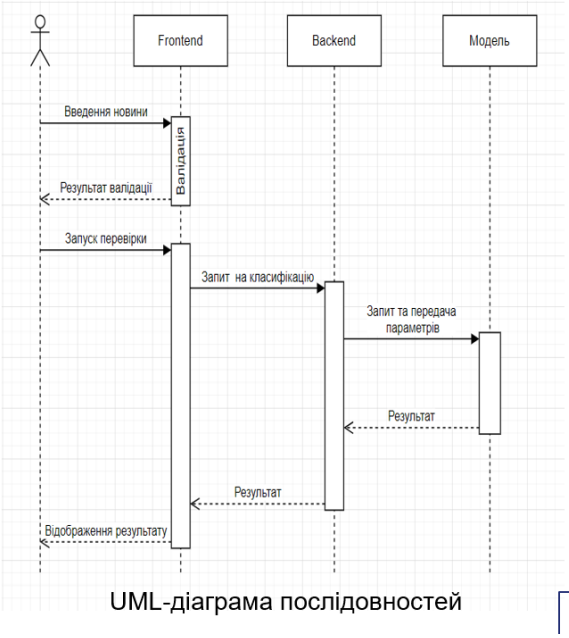
ФУНКЦІОНАЛЬНА СХЕМА ІНФОРМАЦІЙНОЇ СИСТЕМИ



UML-ДІАГРАМИ

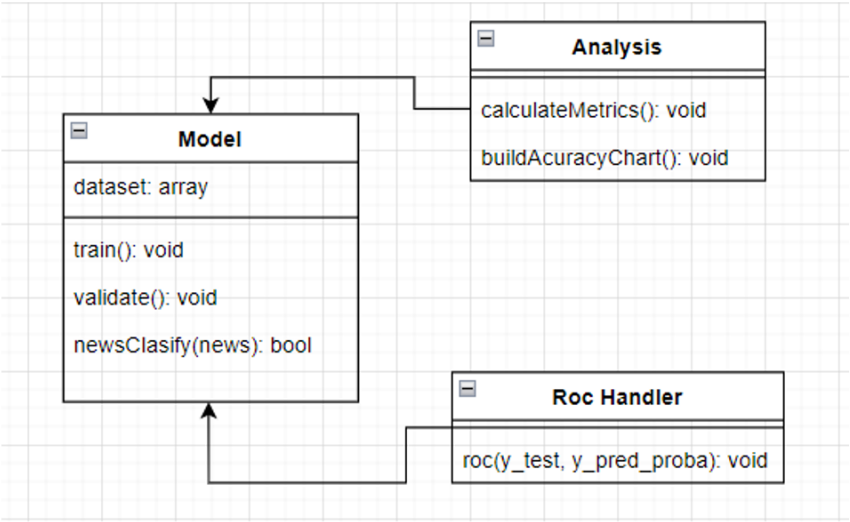


UML-діаграма активності



UML-діаграма послідовностей

UML-ДІАГРАМИ



UML-діаграма класів

ІНФОРМАЦІЙНА СИСТЕМА РЕАЛІЗАЦІЇ ЗАПРОПОНОВАНОГО МЕТОДУ

Головна сторінка інформаційної системи реалізації методу виявлення фейкових новин

Поле для введення досліджуваної новини

Відображення помилки валідації введеного тексту новини

Вигляд інформаційної системи при коректно введеному тексті досліджуваної новини

Відображення фейкового результату класифікації новини

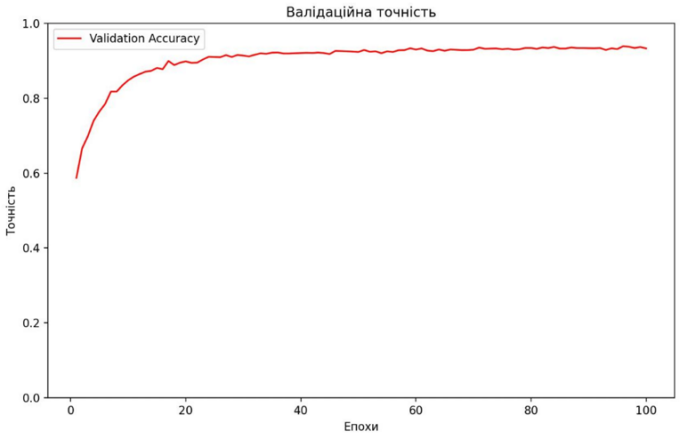
Відображення правдивого результату класифікації новини

Завдання 6

Оцінка отриманих результатів виявлення в Інтернеті фейкових новин запропонованим методом

- проведено оцінку ефективності запропонованого авторського методу. В якості моделей для порівняння обрано існуючі моделі TensortFlow classification model і LogisticRegression. Порівняння проведено на наборах даних «PolitiFact», «LIAR»
- для полегшення проведення порівняльної оцінки обгрунтовано доцільність застосування програмного середовища «Anaconda»
- за результатами проведених експериментів встановлено, що для запропонованого авторського методу на наборі даних «PolitiFact» середня загальна точність становить 93,22%, а стандартне відхилення - 0,99%. На наборі даних «LIAR» середня загальна точність становить 91,57%, а стандартне відхилення складає 0,78%. При цьому, запропонований авторський метод загалом проявляє себе більш ефективним у порівнянні з існуючими методами з числа досліджуваних

ОКРЕМІ РЕЗУЛЬТАТИ ОЦІНКИ ЕФЕКТИВНОСТІ ЗАПРОПОНОВАНОГО АВТОРСЬКОГО МЕТОДУ



Загальна точність на навчальному наборі даних «PolitiFact»

ОКРЕМІ РЕЗУЛЬТАТИ ОЦІНКИ ЕФЕКТИВНОСТІ ЗАПРОПОНОВАНОГО АВТОРСЬКОГО МЕТОДУ

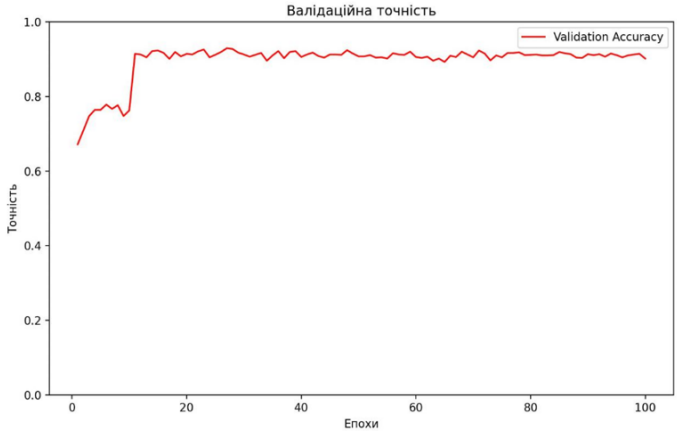
Точнісні характеристики досліджуваних методів на наборі даних «PolitiFact»

Метод	Загальна точність	F1-score	recall	precision
Запропонований авторський	**93,32%**	**0,926**	0,919	**0,93**
TensortFlow classification model	89,37%	0,893	0,88	0,896
LogisticRegression	91,4%	0,914	**0,924**	0,916

Приріст значень показників метрик для запропонованого авторського методу у порівнянні з існуючими методами на наборі даних «PolitiFact»

Пари порівнюваних методів	**Загальна точність**	**F1-score**	**recall**	**precision**
Запропонований авторський - TensortFlow classification model	**3,95%**	**0,033**	**0,039**	**0,034**
Запропонований авторський - LogisticRegression	**1,920%**	**0,012**	-0,005	**0,014**

ОКРЕМІ РЕЗУЛЬТАТИ ОЦІНКИ ЕФЕКТИВНОСТІ ЗАПРОПОНОВАНОГО АВТОРСЬКОГО МЕТОДУ



Загальна точність на навчальному наборі даних «LIAR»

ОКРЕМІ РЕЗУЛЬТАТИ ОЦІНКИ ЕФЕКТИВНОСТІ ЗАПРОПОНОВАНОГО АВТОРСЬКОГО МЕТОДУ

Точнісні характеристики досліджуваних методів на наборі даних «LIAR»

Метод	Загальна точність	F1-score	recall	precision
Запропонований авторський	**91,36%**	**0,913**	0,902	**0,915**
TensortFlow classification model	89,11%	0,891	0,889	0,892
LogisticRegression	91,18%	0,911	**0,908**	**0,915**

Приріст значень показників метрик для запропонованого авторського методу у порівнянні з існуючими методами на наборі даних «LIAR»

Пари порівнюваних методів	**Загальна точність**	**F1-score**	**recall**	**precision**
Запропонований авторський - TensortFlow classification model	**2,25%**	**0,022**	**0,013**	**0,023**
Запропонований авторський - LogisticRegression	**0,18%**	**0,002**	-0,006	**0**

ВИСНОВКИ

Отримані результати забезпечують можливість підвищення ефективності встановлення достовірності новин і їх класифікації у режимі реального часу

Додаток В

Програмний код

Лістинг № 1 main.py

```
import numpy as np
import pandas as pd
import itertools
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
import roc_handle
from functools import reduce


df = pd.read_csv('news.csv')

shape = df.shape
df.head()

labels = df['label']
labels.head()

x_train, x_test, y_train, y_test = train_test_split(df['text'], labels, test_size=0.2, random_state=7)
tfidf_vectorizer = TfidfVectorizer(stop_words='english', max_df=0.7)
tfidf_train = tfidf_vectorizer.fit_transform(x_train)
tfidf_test = tfidf_vectorizer.transform(x_test)

pac = PassiveAggressiveClassifier(max_iter=50)
pac.fit(tfidf_train, y_train)

dense_matrix = tfidf_test.tocoo()

test_data = list(dense_matrix.data)

big_len = len(test_data)
min_len = len(y_test)

step = round(big_len/min_len)

end = step
start = 0

y_roc_pred = []

for i in range(min_len):
    res = reduce(lambda a, b: a + b, test_data[start:end]) / step
    y_roc_pred.append(res)
    start += step
    end += step

y_roc_test = list(map(lambda x: 1 if x == 'REAL' else 0, y_test))

roc_handle.roc(y_roc_test, y_roc_pred)

y_pred = pac.predict(tfidf_test)
score = accuracy_score(y_test, y_pred)
print(f'Accuracy: {round(score * 100, 2)}%')

res = confusion_matrix(y_test, y_pred, labels=['FAKE', 'REAL'])
print(f'Result: {res}')

while True:
    print('Enter your text:')
    text = input()
    tfidf_input = tfidf_vectorizer.transform([text])

    y_pred = pac.predict(tfidf_input)
    print(y_pred)
```

Лістинг № 2 roc.py

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
import matplotlib.pyplot as plt


data = pd.read_csv(url)

X = data[['student', 'balance', 'income']]
y = data['default']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)

log_regression = LogisticRegression()



log_regression.fit(X_train, y_train)

y_pred_proba = log_regression.predict_proba(X_test)[::, 1]
fpr, tpr, _ = metrics.roc_curve(y_test, y_pred_proba)


plt.plot(fpr, tpr)
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```

Лістинг № 3 roc_handle.py

```
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
import matplotlib.pyplot as plt


def roc(y_test, y_pred_proba):
    fpr, tpr, _ = metrics.roc_curve(y_test, y_pred_proba)

    plt.plot(fpr, tpr)
    plt.ylabel('True Positive Rate')
    plt.xlabel('False Positive Rate')
    plt.show()
```

Лістинг № 4 requirements.txt

```
numpy
pandas
scikit-learn
matplotlib
```

Лістинг № 5 model_clasifier.py

```
namespace App\ServiceLayer\ImportCatalog\Jobs;

use App\Domain\Entities\Catalog\News;
use App\Domain\Entities\Catalog\ News Set;
use Illuminate\Bus\Queueable;
use Illuminate\Contracts\Queue\ShouldBeUnique;
use Illuminate\Contracts\Queue\ShouldQueue;
use Illuminate\Foundation\Bus\Dispatchable;
use Illuminate\Queue\InteractsWithQueue;
use Illuminate\Queue\SerializesModels;
use Illuminate\Support\Facades\DB;

class ImportSetsManufacturers implements ShouldQueue, ShouldBeUnique
{
    use Dispatchable, InteractsWithQueue, Queueable, SerializesModels, Fill News FromData;

    public function handle(): void
    {
        foreach (News Set::cursor() as $set) {
            if(count($set-> News Manufacturers->unique()) >= 1){
                if(count($set-> News Manufacturers->unique()) === 1){
                    $set->manufacturer_id = $set-> News Manufacturers->unique()->first()->id;
                }
                else if(count($set-> News Manufacturers->unique()) > 1){
                    $all_manufacturers = $set-> News Manufacturers->unique();
                    if(!$set->manufacturer_id){
                        $set->manufacturer_id = $all_manufacturers->first()->id;
                    }
                    $key = $all_manufacturers->search(function($item) use ($set) {
                        return $item->id == $set->manufacturer_id;
                    });

$all_manufacturers->pull($key);

                    foreach ($all_manufacturers as $manufacturer){
                        $newSet = News Set::query()

->where('manufacturer_id', $manufacturer->id)

->where('initial_set_id', $set->id)
                            ->first();
                        if(!$newSet){
                            $newSet = $set->replicate();
                            $newSet->name = $set->name . ' - ' . $manufacturer->name;

$newSet->manufacturer_id = $manufacturer->id;

$newSet->initial_set_id = $set->id;
                            $newSet->save();
                            $newSet->slug = $newSet->getKey();
                            $newSet->save();
                        }
                        $ News s = News::where('set_id', $set->id)

->where('manufacturer_id', $manufacturer->id)
                            ->get();
                        $ News s->map(function($p) use ($newSet){
                            $setSeq = DB::table('News _set_sequence')
                                ->where(' News _id', $p->id)
```

```
                                ->where('set_id',
$p->set_id)
                                ->first();
                            if($setSeq){
                                DB::table('   News
_set_sequence')
                                    ->where('
News _id', $p->id)

->where('set_id', $p->set_id)

->update(['set_id' => $newSet->id]);
                            } else {
                                $checkIfExists   =
DB::table(' News _set_sequence')
                                    ->where('
News _id', $p->id)

->where('set_id', $newSet->id)
                                    ->first();

if(!$checkIfExists){
                                    DB::table('
News _set_sequence')->insert([
                                        '     News
_id' => $p->id,
                                        'set_id'
=> $newSet->id,
                                        'index'
=> 0,
                                    ]);
                                }
                            }
                            $p->set_id          =
$newSet->id;
                            $p->save();

                        });

                    }
                }
                $set->save();
            }
        }
    }
}
```

Лістинг № 6 model_evaluator.py

```
class    OfferFactory    implements    ShouldQueue,
ShouldBeUnique
{
    use     Dispatchable,     InteractsWithQueue,
Queueable, SerializesModels, CatchFailed {
        CatchFailed::fail                insteadof
InteractsWithQueue;
    }

    public News $ News;
    protected LoggerInterface $logger;

    public function __construct(News $ News)
    {
        $this-> News = $ News;
    }

    public     function     handle(LoggerInterface
$logger): void
    {
        $this->logger = $logger;
        $ News = $this-> News;

        $offerType = is_null($News ->package_id) ?
OfferType::SINGLE : OfferType::PACKAGE;

        switch ($offerType) {
            case OfferType::PACKAGE:
                $offer                           =
$this->makePackageOffer($News);
                break;
            case OfferType::SINGLE:
            default:
                $offer                           =
$this->makeSingleOffer($News);
        }

        $this->saveOfferAsPrimary($offer);
    }

    private function getOffer(News $ News, string
$type): Offer
    {
        $attrs = [
            'type'       => $type,
            'primary_id' => $ News ->getKey(),
        ];

        return
Offer::withTrashed()->where($attrs)->first()    ??
new Offer($attrs);
    }

    private function makeSingleOffer(News $ News):
Offer
```

```
    {
        $offer      =      $this->getOffer($News,
OfferType::SINGLE);
        $offer->getKey();

        $offer->setRelation('primary', $ News);
        $offer->setRelation('    News    s',    new
Collection([$News]));

        $offer->prices                          =
$this->createPricesForOffer($offer);
        $offer->active_price                    =
$offer->activePriceFilter();
        $offer->properties          =          new
OfferProperties();
        $offer->deleted_at = null;

        $offer->save();
        $offer-> News s()->sync([
            $ News ->getKey() => [
                'quantity' => 1, // Single News
attach
            ],
        ]);

        return $offer;
    }

    private     function     makePackageOffer(News
$ News): Offer
    {
        $package News s = $ News ->items;

        if (0 === $package News s->count()) {
            $package News s->push($News);
        }

        $offer      =      $this->getOffer($News,
OfferType::PACKAGE);
        $offer->getKey();

        $prices                                 =
$this->createPricesForOffer($offer);
        $ News s = [];

        $offer->setRelation('primary', $ News);
        $offer->setRelation('  News  s',  $package
News s);

        /** @var News $package News */
        foreach ($package News s as $package News)
{
            $prices                             =
$prices->merge($this->createPricesForOffer($offer
, $package News ->getKey())->all());
            $quantity  =  data_get($package  News,
'pivot.quantity', 1);

            $ News s[$package News ->getKey()] = [
                'quantity' => $quantity,
            ];
        }

        $offer->prices = $prices;
        $offer->active_price                    =
$offer->activePriceFilter();
        $offer->properties          =          new
OfferProperties();
        $offer->deleted_at = null;

        $offer->save();
        $offer-> News s()->sync($News s);

        return $offer;
    }

    private  function  createPricesForOffer(Offer
$offer, $ News Id = null, $withDeliveryAssembly =
true): SimpleCollection
    {
        $collection = collect();

        $prices = [
            PriceType::TOP     => 'retail_price',
            PriceType::LOWER   => 'retail_price',
        ];

        /** @var News $ News */
        $  News  News  =  is_null($productId)  ||
$productId === $offer->primary_id ?
            $offer->primary :
            $offer->products->find($productId);

        foreach ($prices as $type => $field) {
            if              (($value              =
$product->getAttribute($field)) > 0) {
                $collection->push(new Price([
                    'type'     => $type,
                    'product'  => $productId,
                    'value'    => (float) $value,
                    'discount' => 0,
                ]));
            }
        }
```

```
        $promotion                                  =
$product->promotions()->first();
        if ($promotion instanceof Promotion) {
            $promoPriceValue                        =
$promotion->priceForProduct($product);

            $collection->push(new Price([
                'type'     => PriceType::PROMO,
                'product'  => $productId,
                'value'              =>    (float)
$promoPriceValue,
                'discount'  =>  (int)  round(100  -
$promoPriceValue / $product->retail_price * 100),
            ]));
        }

        if ($withDeliveryAssembly) {
цены на доставку и сборку так же из продукта
            $collection
                ->push(new Price([
                    'type'                      =>
PriceType::DELIVERY,
                    'product'  => $productId,
                    'value'           =>   (float)
$product->delivery_price,
                    'discount' => 0,
                ]))
                ->push(new Price([
                    'type'                      =>
PriceType::ASSEMBLING,
                    'product'  => $productId,
                    'value'           =>   (float)
$product->assembling_price,
                    'discount' => 0,
                ]));
        }

        return $collection;
    }

    private   function   saveOfferAsPrimary(Offer
$offer)
    {
        $this->product->offer_id                =
$offer->getKey();
        $this->product->save();
    }
}
```

Лістинг № 7 feature_selection.py

```
import DataPrep
import pandas as pd
import numpy as np
from      sklearn.feature_extraction.text      import
CountVectorizer
from      sklearn.feature_extraction.text      import
TfidfTransformer
from      sklearn.feature_extraction.text      import
TfidfVectorizer
from sklearn.pipeline import Pipeline
import nltk
import nltk.corpus
from nltk.tokenize import word_tokenize
from gensim.models.word2vec import Word2Vec



#we will start with simple bag of words technique
#creating feature vector - document term matrix
countV = CountVectorizer()
train_count                                          =
countV.fit_transform(DataPrep.train_news['Statement']
.values)

print(countV)
print(train_count)

#print training doc term matrix
#we have matrix of size of (10240, 12196) by calling
below
def get_countVectorizer_stats():

    #vocab size
    train_count.shape

    #check vocabulary using below command
    print(countV.vocabulary_)

    #get feature names
    print(countV.get_feature_names()[:25])



#create tf-df frequency features
#tf-idf
tfidfV = TfidfTransformer()
train_tfidf = tfidfV.fit_transform(train_count)

def get_tfidf_stats():
    train_tfidf.shape
```

```
    #get train data feature names
    print(train_tfidf.A[:10])


#bag of words - with n-grams
#countV_ngram                                          =
CountVectorizer(ngram_range=(1,3),stop_words='english
')
#tfidf_ngram                                           =
TfidfTransformer(use_idf=True,smooth_idf=True)

tfidf_ngram                                            =
TfidfVectorizer(stop_words='english',ngram_range=(1,4
),use_idf=True,smooth_idf=True)



#POS Tagging
tagged_sentences = nltk.corpus.treebank.tagged_sents()

cutoff = int(.75 * len(tagged_sentences))
training_sentences = DataPrep.train_news['Statement']

print(training_sentences)

#training POS tagger based on words
def features(sentence, index):
    """ sentence: [w1, w2, ...], index: the index of
the word """
    return {
        'word': sentence[index],
        'is_first': index == 0,
        'is_last': index == len(sentence) - 1,
        'is_capitalized':   sentence[index][0].upper()
== sentence[index][0],
        'is_all_caps':    sentence[index].upper()    ==
sentence[index],
        'is_all_lower':   sentence[index].lower()   ==
sentence[index],
        'prefix-1': sentence[index][0],
        'prefix-2': sentence[index][:2],
        'prefix-3': sentence[index][:3],
        'suffix-1': sentence[index][-1],
        'suffix-2': sentence[index][-2:],
        'suffix-3': sentence[index][-3:],
        'prev_word':   ''   if   index   ==   0   else
sentence[index - 1],
        'next_word': '' if index == len(sentence) - 1
else sentence[index + 1],
        'has_hyphen': '-' in sentence[index],
        'is_numeric': sentence[index].isdigit(),
        'capitals_inside': sentence[index][1:].lower()
!= sentence[index][1:]
    }


#helper  function  to  strip  tags  from  tagged  corpus

def untag(tagged_sentence):
    return [w for w, t in tagged_sentence]



#Using Word2Vec
with open("glove.6B.50d.txt", "rb") as lines:
    w2v   =   {line.split()[0]:   np.array(map(float,
line.split()[1:]))
           for line in lines}



#model = gensim.models.Word2Vec(X, size=100) # x be
tokenized text
#w2v = dict(zip(model.wv.index2word, model.wv.syn0))


class MeanEmbeddingVectorizer(object):
    def __init__(self, word2vec):
        self.word2vec = word2vec
        # if a text is empty we should return a vector
of zeros
        # with the same dimensionality as all the other
vectors
        self.dim = len(word2vec.itervalues().next())

    def fit(self, X, y):
        return self

    def transform(self, X):
        return np.array([
            np.mean([self.word2vec[w] for w in words if
w in self.word2vec]
                    or [np.zeros(self.dim)], axis=0)
            for words in X
        ])
```

Лістинг № 8 classifier.py

```
import DataPrep
import FeatureSelection
import numpy as np
import pandas as pd
```

```
import pickle
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.pipeline import Pipeline
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import  LogisticRegression
from sklearn.linear_model import SGDClassifier
from sklearn import svm
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import KFold
from sklearn.metrics import confusion_matrix, f1_score, classification_report
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import learning_curve
import matplotlib.pyplot as plt
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import average_precision_score

#string to test
doc_new = ['obama is running for president in 2016']

#the feature selection has been done in FeatureSelection.py module. here we will create models using those features for prediction

#first we will use bag of words techniques

#building classifier using naive bayes
nb_pipeline = Pipeline([
        ('NBCV',FeatureSelection.countV),
        ('nb_clf',MultinomialNB())])

nb_pipeline.fit(DataPrep.train_news['Statement'],DataPrep.train_news['Label'])
predicted_nb = nb_pipeline.predict(DataPrep.test_news['Statement'])
np.mean(predicted_nb == DataPrep.test_news['Label'])


#building classifier using logistic regression
logR_pipeline = Pipeline([
        ('LogRCV',FeatureSelection.countV),
        ('LogR_clf',LogisticRegression())
        ])
logR_pipeline.fit(DataPrep.train_news['Statement'],DataPrep.train_news['Label'])
predicted_LogR = logR_pipeline.predict(DataPrep.test_news['Statement'])
np.mean(predicted_LogR == DataPrep.test_news['Label'])


#building Linear SVM classfier
svm_pipeline = Pipeline([
        ('svmCV',FeatureSelection.countV),
        ('svm_clf',svm.LinearSVC())
        ])

svm_pipeline.fit(DataPrep.train_news['Statement'],DataPrep.train_news['Label'])
predicted_svm = svm_pipeline.predict(DataPrep.test_news['Statement'])
np.mean(predicted_svm == DataPrep.test_news['Label'])


#using SVM Stochastic Gradient Descent on hinge loss
sgd_pipeline = Pipeline([
        ('svm2CV',FeatureSelection.countV),
        ('svm2_clf',SGDClassifier(loss='hinge', penalty='l2', alpha=1e-3, n_iter=5))
        ])

sgd_pipeline.fit(DataPrep.train_news['Statement'],DataPrep.train_news['Label'])
predicted_sgd = sgd_pipeline.predict(DataPrep.test_news['Statement'])
np.mean(predicted_sgd == DataPrep.test_news['Label'])


#random forest
random_forest = Pipeline([
        ('rfCV',FeatureSelection.countV),

('rf_clf',RandomForestClassifier(n_estimators=200,n_jobs=3))
        ])

random_forest.fit(DataPrep.train_news['Statement'],DataPrep.train_news['Label'])
predicted_rf = random_forest.predict(DataPrep.test_news['Statement'])
np.mean(predicted_rf == DataPrep.test_news['Label'])
```

```
#User defined functon for K-Fold cross validatoin
def build_confusion_matrix(classifier):

    k_fold = KFold(n_splits=5)
    scores = []
    confusion = np.array([[0,0],[0,0]])

    for train_ind, test_ind in
k_fold.split(DataPrep.train_news):
        train_text =
DataPrep.train_news.iloc[train_ind]['Statement']
        train_y =
DataPrep.train_news.iloc[train_ind]['Label']

        test_text =
DataPrep.train_news.iloc[test_ind]['Statement']
        test_y =
DataPrep.train_news.iloc[test_ind]['Label']

        classifier.fit(train_text,train_y)
        predictions = classifier.predict(test_text)

        confusion +=
confusion_matrix(test_y,predictions)
        score = f1_score(test_y,predictions)
        scores.append(score)

    return (print('Total statements classified:',
len(DataPrep.train_news)),
    print('Score:', sum(scores)/len(scores)),
    print('score length', len(scores)),
    print('Confusion matrix:'),
    print(confusion))

#K-fold cross validation for all classifiers
build_confusion_matrix(nb_pipeline)
build_confusion_matrix(logR_pipeline)
build_confusion_matrix(svm_pipeline)
build_confusion_matrix(sgd_pipeline)
build_confusion_matrix(random_forest)
nb_pipeline_ngram = Pipeline([
        ('nb_tfidf',FeatureSelection.tfidf_ngram),
        ('nb_clf',MultinomialNB())])

nb_pipeline_ngram.fit(DataPrep.train_news['Statement'
],DataPrep.train_news['Label'])
predicted_nb_ngram =
nb_pipeline_ngram.predict(DataPrep.test_news['Stateme
nt'])
np.mean(predicted_nb_ngram ==
DataPrep.test_news['Label'])


#logistic regression classifier
logR_pipeline_ngram = Pipeline([
        ('LogR_tfidf',FeatureSelection.tfidf_ngram),

('LogR_clf',LogisticRegression(penalty="l2",C=1))
        ])

logR_pipeline_ngram.fit(DataPrep.train_news['Statemen
t'],DataPrep.train_news['Label'])
predicted_LogR_ngram =
logR_pipeline_ngram.predict(DataPrep.test_news['State
ment'])
np.mean(predicted_LogR_ngram ==
DataPrep.test_news['Label'])


#linear SVM classifier
svm_pipeline_ngram = Pipeline([
        ('svm_tfidf',FeatureSelection.tfidf_ngram),
        ('svm_clf',svm.LinearSVC())
        ])

svm_pipeline_ngram.fit(DataPrep.train_news['Statement
'],DataPrep.train_news['Label'])
predicted_svm_ngram =
svm_pipeline_ngram.predict(DataPrep.test_news['Statem
ent'])
np.mean(predicted_svm_ngram ==
DataPrep.test_news['Label'])


#sgd classifier
sgd_pipeline_ngram = Pipeline([
         ('sgd_tfidf',FeatureSelection.tfidf_ngram),
         ('sgd_clf',SGDClassifier(loss='hinge',
penalty='l2', alpha=1e-3, n_iter=5))
         ])

sgd_pipeline_ngram.fit(DataPrep.train_news['Statement
'],DataPrep.train_news['Label'])
predicted_sgd_ngram =
sgd_pipeline_ngram.predict(DataPrep.test_news['Statem
ent'])
np.mean(predicted_sgd_ngram ==
DataPrep.test_news['Label'])
```

```
#random forest classifier
random_forest_ngram = Pipeline([
        ('rf_tfidf',FeatureSelection.tfidf_ngram),

('rf_clf',RandomForestClassifier(n_estimators=300,n_jobs=3))
        ])

random_forest_ngram.fit(DataPrep.train_news['Statement'],DataPrep.train_news['Label'])
predicted_rf_ngram = random_forest_ngram.predict(DataPrep.test_news['Statement'])
np.mean(predicted_rf_ngram == DataPrep.test_news['Label'])



#K-fold cross validation for all classifiers
build_confusion_matrix(nb_pipeline_ngram)
build_confusion_matrix(logR_pipeline_ngram)
build_confusion_matrix(svm_pipeline_ngram)
build_confusion_matrix(sgd_pipeline_ngram)
build_confusion_matrix(random_forest_ngram)
#n-grams & tfidf confusion matrix and F1 scores

#Naive bayes
# [841 3647]
print(classification_report(DataPrep.test_news['Label'], predicted_nb_ngram))
print(classification_report(DataPrep.test_news['Label'], predicted_LogR_ngram))
print(classification_report(DataPrep.test_news['Label'], predicted_svm_ngram))
print(classification_report(DataPrep.test_news['Label'], predicted_sgd_ngram))
print(classification_report(DataPrep.test_news['Label'], predicted_rf_ngram))

DataPrep.test_news['Label'].shape

"""
Out of all the models fitted, we would take 2 best performing model. we would call them candidate models
from the confusion matrix, we can see that random forest and logistic regression are best performing
in terms of precision and recall (take a look into false positive and true negative counts which appeares
to be low compared to rest of the models)
"""
```

```
#grid-search parameter optimization
#random forest classifier parameters
parameters = {'rf_tfidf__ngram_range': [(1, 1), (1, 2),(1,3),(1,4),(1,5)],
              'rf_tfidf__use_idf': (True, False),
              'rf_clf__max_depth': (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)
}

gs_clf = GridSearchCV(random_forest_ngram, parameters, n_jobs=-1)
gs_clf = gs_clf.fit(DataPrep.train_news['Statement'][:10000],DataPrep.train_news['Label'][:10000])

gs_clf.best_score_
gs_clf.best_params_
gs_clf.cv_results_

#logistic regression parameters
parameters = {'LogR_tfidf__ngram_range': [(1, 1), (1, 2),(1,3),(1,4),(1,5)],
              'LogR_tfidf__use_idf': (True, False),
              'LogR_tfidf__smooth_idf': (True, False)
}

gs_clf = GridSearchCV(logR_pipeline_ngram, parameters, n_jobs=-1)
gs_clf = gs_clf.fit(DataPrep.train_news['Statement'][:10000],DataPrep.train_news['Label'][:10000])

gs_clf.best_score_
gs_clf.best_params_
gs_clf.cv_results_

#Linear SVM
parameters = {'svm_tfidf__ngram_range': [(1, 1), (1, 2),(1,3),(1,4),(1,5)],
              'svm_tfidf__use_idf': (True, False),
              'svm_tfidf__smooth_idf': (True, False),
              'svm_clf__penalty': ('l1','l2'),
}

gs_clf = GridSearchCV(svm_pipeline_ngram, parameters, n_jobs=-1)
gs_clf = gs_clf.fit(DataPrep.train_news['Statement'][:10000],DataPrep.train_news['Label'][:10000])
```

```
gs_clf.best_score_
gs_clf.best_params_
gs_clf.cv_results_

#by running above commands we can find the model with best performing parameters


#running both random forest and logistic regression models again with best parameter found with GridSearch method
random_forest_final = Pipeline([

('rf_tfidf',TfidfVectorizer(stop_words='english',ngram_range=(1,3),use_idf=True,smooth_idf=True)),

('rf_clf',RandomForestClassifier(n_estimators=300,n_jobs=3,max_depth=10))
        ])

random_forest_final.fit(DataPrep.train_news['Statement'],DataPrep.train_news['Label'])
predicted_rf_final                                    = random_forest_final.predict(DataPrep.test_news['Statement'])
np.mean(predicted_rf_final                            == DataPrep.test_news['Label'])
print(metrics.classification_report(DataPrep.test_news['Label'], predicted_rf_final))

logR_pipeline_final = Pipeline([
        #('LogRCV',countV_ngram),

('LogR_tfidf',TfidfVectorizer(stop_words='english',ngram_range=(1,5),use_idf=True,smooth_idf=False)),

('LogR_clf',LogisticRegression(penalty="l2",C=1))
        ])

logR_pipeline_final.fit(DataPrep.train_news['Statement'],DataPrep.train_news['Label'])
predicted_LogR_final                                  = logR_pipeline_final.predict(DataPrep.test_news['Statement'])
np.mean(predicted_LogR_final                          == DataPrep.test_news['Label'])
#accuracy = 0.62
print(metrics.classification_report(DataPrep.test_news['Label'], predicted_LogR_final))

#saving best model to the disk
model_file = 'final_model.sav'
pickle.dump(logR_pipeline_ngram,open(model_file,'wb'))


#Plotting learing curve
def plot_learing_curve(pipeline,title):
    size = 10000
    cv = KFold(size, shuffle=True)

    X = DataPrep.train_news["Statement"]
    y = DataPrep.train_news["Label"]

    pl = pipeline
    pl.fit(X,y)

    train_sizes,    train_scores,    test_scores    = learning_curve(pl,    X,    y,    n_jobs=-1,    cv=cv, train_sizes=np.linspace(.1, 1.0, 5), verbose=0)

    train_scores_mean = np.mean(train_scores, axis=1)
    train_scores_std = np.std(train_scores, axis=1)
    test_scores_mean = np.mean(test_scores, axis=1)
    test_scores_std = np.std(test_scores, axis=1)

    plt.figure()
    plt.title(title)
    plt.legend(loc="best")
    plt.xlabel("Training examples")
    plt.ylabel("Score")
    plt.gca().invert_yaxis()

    # box-like grid
    plt.grid()

    # plot the std deviation as a transparent range at each training set size
    plt.fill_between(train_sizes, train_scores_mean - train_scores_std,          train_scores_mean          + train_scores_std, alpha=0.1, color="r")
    plt.fill_between(train_sizes, test_scores_mean - test_scores_std, test_scores_mean + test_scores_std, alpha=0.1, color="g")

    # plot the average training and test score lines at each training set size
    plt.plot(train_sizes,   train_scores_mean,   'o-', color="r", label="Training score")
```

```
    plt.plot(train_sizes,   test_scores_mean,   'o-',
color="g", label="Cross-validation score")

    # sizes the window for readability and displays the
plot
    # shows error from 0 to 1.1
    plt.ylim(-.1,1.1)
    plt.show()



#below command will plot learing curves for each of the
classifiers
plot_learing_curve(logR_pipeline_ngram,"Naive-bayes
Classifier")
plot_learing_curve(nb_pipeline_ngram,"LogisticRegress
ion Classifier")
plot_learing_curve(svm_pipeline_ngram,"SVM
Classifier")
plot_learing_curve(sgd_pipeline_ngram,"SGD
Classifier")
plot_learing_curve(random_forest_ngram,"RandomForest
Classifier")

"""
by plotting the learning cureve for logistic
regression, it can be seen that cross-validation score
is stagnating throughout and it
is unable to learn from data. Also we see that there
are high errors that indicates model is simple and we
may want to increase the
model complexity.
"""



#plotting Precision-Recall curve
def plot_PR_curve(classifier):

    precision,       recall,       thresholds       =
precision_recall_curve(DataPrep.test_news['Label'],
classifier)
    average_precision                                 =
average_precision_score(DataPrep.test_news['Label'],
classifier)

    plt.step(recall, precision, color='b', alpha=0.2,
             where='post')
    plt.fill_between(recall,  precision,  step='post',
alpha=0.2,
                     color='b')

    plt.xlabel('Recall')
    plt.ylabel('Precision')
    plt.ylim([0.0, 1.05])
    plt.xlim([0.0, 1.0])
    plt.title('2-class Random Forest Precision-Recall
curve: AP={0:0.2f}'.format(
              average_precision))

plot_PR_curve(predicted_LogR_ngram)
plot_PR_curve(predicted_rf_ngram)
def  show_most_informative_features(model,  vect,  clf,
text=None, n=50):
    # Extract the vectorizer and the classifier from
the pipeline
    vectorizer = model.named_steps[vect]
    classifier = model.named_steps[clf]

     # Check to make sure that we can perform this
computation
    if not hasattr(classifier, 'coef_'):
        raise TypeError(
            "Cannot compute most informative features
on {}.".format(
                classifier.__class__.__name__
            )
        )

    if text is not None:
        # Compute the coefficients for the text
        tvec = model.transform([text]).toarray()
    else:
        # Otherwise simply use the coefficients
        tvec = classifier.coef_

    # Zip the feature names with the coefs and sort
    coefs = sorted(
        zip(tvec[0], vectorizer.get_feature_names()),
        reverse=True
    )

    # Get the top n and bottom n coef, name pairs
    topn  = zip(coefs[:n], coefs[:-(n+1):-1])

   output = []

    # If text, add the predicted value to the output.
    if text is not None:
        output.append("\"{}\"".format(text))
        output.append(
```

```
            "Classified                              as:
{}".format(model.predict([text]))
        )
        output.append("")

    # Create two columns with most negative and most
positive features.
    for (cp, fnp), (cn, fnn) in topn:
        output.append(
            "{:0.4f}{: >15}    {:0.4f}{: >15}".format(
                cp, fnp, cn, fnn
            )
        )
    #return "\n".join(output)
```

```
        print(output)

show_most_informative_features(logR_pipeline_ngram,ve
ct='LogR_tfidf',clf='LogR_clf')
show_most_informative_features(nb_pipeline_ngram,vect
='nb_tfidf',clf='nb_clf')
show_most_informative_features(svm_pipeline_ngram,vec
t='svm_tfidf',clf='svm_clf')
show_most_informative_features(sgd_pipeline_ngram,vec
t='sgd_tfidf',clf='sgd_clf')
```

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ
кафедра комп'ютерних наук

ВІДГУК НАУКОВОГО КЕРІВНИКА

на кваліфікаційну роботу магістра

гр. КНм-22-1 Боровика Дмитра Олеговича за темою: *Метод виявлення фейкових новин для оцінки достовірності джерел масової інформації*

1. Актуальність теми

Актуальність теми роботи обумовлена рядом причин. Зокрема, зростанням ролі онлайн соціальних мереж (ОСМ), поширенням новин в режимі реального часу незалежно від їх достовірності, поширенням фейкових новин засобами ОСМ, стрімким розвитком в останній період штучного інтелекту. У роботі актуальність обраної теми знайшла достатнє обгрунтування.

2. Відповідність роботи предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до наукових робіт

Тема кваліфікаційної роботи студента повністю відповідає предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до кваліфікаційної роботи магістра. Робота вирішує науково-прикладну задачу у сфері комп'ютерних наук (а саме застосування інтелектуальних методів обробки інформації (моделей нейромереж) до розпізнавання фейкових новин в Інтернеті). У роботі проведено дослідження, яке розвиває існуючі знання та процедури.

3. Професійні та особистісні якості магістранта

Під час опрацювання магістерської роботи магістрант продемонстрував такі професійні якості, як вміння ставити задачі, проводити їх аналіз і декомпозицію, здійснювати підбір і застосування наукових методів для вирішення часткових завдань дослідження, вміння знаходити причинно-наслідкові залежності та формувати висновки, вміння застосовувати сучасні інформаційні технології. Крім цього, магістрант проявив такі особистісні якості, як наполегливість, відповідальність, доброчесність, працелюбність, коректність.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

У ході роботи магістрант проявив достатній ступінь самостійності. Зокрема, ним особисто було: проведено аналіз існуючих моделей і підходів для виявлення в Інтернеті фейкових новин; удосконалено метод виявлення в Інтернеті фейкових новин нейромережевими засобами; підготовлено набір даних для навчання нейронної мережі та

здійснено її навчання виявляти фейкові новини; розроблено інформаційну систему, що реалізує запропоноване ним удосконалення; проведено оцінку отриманих результатів.

5. Наукова новизна та оригінальність запропонованих підходів

Магістерська робота характеризується наявністю наукової новизни. Відповідні положення сформульовані коректно та відображають її сутність.

Результати магістерської роботи достатньо оприлюднені та апробовані. Зокрема, матеріали роботи оприлюднені у науковій статті, що опублікована у фаховому науковому виданні. Також матеріали роботи доповідалися на двох наукових конференціях в галузі інформаційних технологій.

6. Ступінь оволодіння методами дослідження

У магістерській роботі магістрантом реалізовано комплексний підхід щодо застосування загальнонаукових і спеціальних методів наукових досліджень. Зокрема, використано методи застосування згорткової нейронної мережі, сучасних інформаційних технологій, вищої математики, теорії ймовірностей і математичної статистики. Під час їх застосування до вирішення окремих завдань дослідження магістрант продемонстрував достатній ступінь володіння ними.

7. Повнота та якість розкриття теми роботи

Тема роботи розкрита достатньо повно та якісно. Визначена мета роботи досягнута.

8. Логічність, послідовність, аргументованість, літературна грамотність викладу матеріалу

Викладення матеріалу магістерської роботи є достатньо логічним, послідовним і має достатній рівень аргументації. Робота написана літературною мовою, граматична якість викладення матеріалу роботи задовільна.

9. Можливість практичного застосування кваліфікаційної роботи, окремих її частин

Робота має теоретико-прикладний характер. Результати роботи можна застосовувати, як для встановлення достовірності новин, так і для їх класифікації у режимі реального часу.

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

Рекомендую допустити роботу до захисту.

Кваліфікаційна робота заслуговує на оцінку «Відмінно».

Науковий керівник _____ д.т.н., проф. Олександр Бармак

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ
кафедра комп'ютерних наук

ВІДГУК ОПОНЕНТА

на кваліфікаційну роботу магістра

гр. КНм-22-1 Боровика Дмитра Олеговича за темою: *Метод виявлення фейкових новин для оцінки достовірності джерел масової інформації*

1. Актуальність обраної теми

Обрана магістрантом тема є достатньо актуальною на даний час, особливо з урахуванням тенденцій щодо розвитку штучного інтелекту. Актуальність обраної теми достатньо обгрунтована в роботі.

2. Відповідність роботи предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до наукових робіт

Тема кваліфікаційної роботи студента повністю відповідає предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до кваліфікаційної роботи магістра. Робота вирішує науково-прикладну задачу у сфері комп'ютерних наук (а саме застосування нейромережевих технологій до розпізнавання фейкових новин в Інтернеті). У роботі присутнє дослідження, яке розвиває існуючі підходи, та отримані нові результати, які за певними показниками – вищі, ніж в аналогічних підходах.

3. Повнота розкриття мети та завдань дослідження

Часткові завдання дослідження сформульовані коректно та повністю відповідають меті роботи.

4. Наявність наукової новизни

Магістерська робота характеризується наявністю наукової новизни. Положення наукової новизни сформульовані коректно та відображають їх сутність.

Матеріали магістерської роботи достатньо апробовані на двох наукових конференціях та оприлюднені у фаховому науковому виданні у вигляді наукової статті.

5. Зміст кожного розділу роботи

Робота складається з чотирьох розділів.

У першому розділі – «Аналіз сучасного стану використання інформаційних технологій для виявлення в Інтернеті фейкових новин» автором проведено аналіз існуючих моделей і підходів для виявлення в Інтернеті фейкових новин. У ході аналізу досліджено базові алгоритми виявлення фейкових новин.

У другому розділі – «Метод виявлення фейкових новин для оцінки достовірності джерел масової інформації» магістрантом удосконалено метод виявлення в Інтернеті

фейкових новин нейромережевими засобами за рахунок удосконалення структури багатошарової CNN нейромережі.

У третьому розділі – «Програмна реалізація методу виявлення фейкових новин на основі використання нейромережевих технологій» автором розроблено інформаційну систему, що реалізує запропонований метод. Для розробки backend частини додатку використано мову програмування PHP та фреймворк даної мови програмування Laravel. Для frontend частини додатку застосовано фреймворк Vue.js. Реалізація методу виявлення фейкових новин здійснена на основі мови програмування Python.

У четвертому розділі – «Експериментальне дослідження запропонованого методу виявлення фейкових новин» магістрантом проведено оцінку ефективності запропонованого авторського методу. В якості моделей, що порівнювались із запропонованим методом з удосконаленою архітектурою, ним вдало обрано існуючі моделі TensortFlow classification model і LogisticRegression. Також у розділі обгрунтовано доцільність використання програмного середовища «Anaconda» для полегшення проведення порівняльної оцінки.

6. Ступінь розкриття теми роботи

Тема роботи розкрита достатньою мірою та мета досягнута.

7. Якість оформлення кваліфікаційної роботи

Робота написана літературною мовою та оформлена згідно відповідних вимог достатньо якісно.

8. Недоліки кваліфікаційної роботи

Разом з тим, у роботі є і певні недоліки. Так, у першому розділі магістерської роботи проведено функціональний аналіз не всіх базових алгоритмів виявлення фейкових новин, які мають відношення до досліджуваної теми.

Проте зазначений недолік загалом не впливає на комплексну позитивну оцінку роботи.

9. Загальний висновок (допускається чи не допускається до захисту), якої оцінки заслуговує кваліфікаційна робота.

Рекомендую допустити роботу до захисту.

Кваліфікаційна робота заслуговує на оцінку «Відмінно».

Опонент _____ д.т.н., проф. Сергій ЛИСЕНКО

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНИХ НАУК

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ МАГІСТРА ДО ЗАХИСТУ ЗА РЕЗУЛЬТАТАМИ АНАЛІЗУ ЗВІТУ ПОДІБНОСТІ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод виявлення фейкових новин для оцінки достовірності джерел масової інформації

Автор: Д.О. Боровик

Спеціальність: 122 – Компютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: д.т.н., професор, зав. кафедри КН О.В.Бармак

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) за програмою Anti-Plagiarism виявлені 1% запозичень вказують на джерела посилань та відомі терміни.
2) За програмою UNICHECK виявлені 15.1% є фрагментарними – містять поширені конструкції та схеми нейромереж (у роботі вказані посилання на ці джерела), загальновідомі терміни, скорочення та визначення.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості, складає 1% і 15.1% відповідно, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ О. В. Бармак

Гарант ОП _____ Р. В. Багрій

Завідувач кафедри КН _____ О. В. Бармак

Tue Dec 26 11:35:28 EET 2023, Петровський Сергій Степанович, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 12%**

	Документ		Сумарний збіг по Базі Даних	
ID: 124784 Назва: КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА на тему Метод виявлення фейкових новин для оцінки достовірності джерел масової інформації Додано в БД: 2023-12-26 Автора: Д.О. Боровик Керівники: О.В. Бармак Консультанти: Опоненти:	Символи	Лексеми	Символи	Лексеми
	114021	1577	2592 (2%)	44 (3%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

UNICHECK
by Turnitin

Ім'я користувача:
Кафедра КН

ID перевірки:
1016036355

Дата перевірки:
26.12.2023 12:31:19 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
26.12.2023 12:32:45 EET

ID користувача:
100005671

Назва документа: **КНм-22-1 Боровик**

Кількість сторінок: **98** **Кількість слів:** **19461** **Кількість символів:** **147052** **Розмір файлу:** **1.52 MB** **ID файлу:** **1015728905**

15.1%
Схожість

Найбільша схожість: **7.79%** **з Інтернет-джерелом** (https://www.mdpi.com/1424-8220/23/4/1748/htm)

14.4% Джерела з Інтернету	944	Сторінка 100
1.79% Джерела з Бібліотеки	56	Сторінка 110

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи	4