

УДК 004.8

DOI <https://doi.org/10.32782/IT/2023-2-5>

Костянтин ОНИЩЕНКО

старший викладач кафедри програмної інженерії, Харківський національний університет радіоелектроніки, просп. Науки, 14, м. Харків, Україна, 61166, kostiantyn.onyshchenko@nure.ua

ORCID: 0000-0002-7746-4570

Scopus Author ID: 57209338164

Анатолій АФАНАСЬЄВ

студент кафедри інформатики, Харківський національний університет радіоелектроніки, просп. Науки, 14, м. Харків, Україна, 61166, anatolii.afanasiev@nure.ua

ORCID: 0009-0005-0707-981X

Бібліографічний опис статті: Онищенко, К., Афанасьєв, А. (2023). Оцінка ефективності використання згорткових та рекурентних нейронних мереж у завданні обробки текстових даних. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 2, 41–49, doi: <https://doi.org/10.32782/IT/2023-2-5>

ОЦІНКА ЕФЕКТИВНОСТІ ВИКОРИСТАННЯ ЗГОРТКОВИХ ТА РЕКУРЕНТНИХ НЕЙРОННИХ МЕРЕЖ У ЗАВДАННІ ОБРОБКИ ТЕКСТОВИХ ДАНИХ

Дана робота присвячена оцінці ефективності використання згорткових та рекурентних нейромереж у завданні обробки текстових даних на прикладі виявлення фейкових новин. Нині зусилля світової спільноти спрямовані на боротьбу з подібною інформацією в цілому, що зумовлює актуальність порушеного питання. Проблематика виявлення фейкових новин полягає у достовірності визначення тої чи іншої інформації як фейкової чи правдивої. Метою роботи є порівняння точності визначення фейкових новин для архітектур згорткових та рекурентних нейромереж, в яких закладена модель синтаксичного аналізу текстів статті шляхом формування міток новин з використанням TF-IDF та Word Embedding. Для досягнення поставленої мети було проведено аналіз області застосування та визначено ключові особливості цього типу інформації. Було розглянуто теоретичну основу обраних архітектур та встановлено їх конфігурації відповідно до поставленої задачі. Для практичної реалізації вибраних типів нейромереж було створено експериментальне середовище. Була виявлена відносна ефективність використання рекурентних нейромереж у порівнянні зі згортковими, а також визначено можливі сценарії, в яких отримані результати можуть змінюватися. У результаті аналізу було встановлено, що на доступних даних згорткова нейромережа має більшу швидкість порівняно з рекурентною нейромережею, але при цьому надає менш точні результати класифікації. З урахуванням запропонованого правила порівняння ефективності, ймовірності помилок різних видів та можливості подолання розбіжностей між алгоритмами, отриманий приріст у продуктивності можна вважати незначним. Цей висновок відповідає світовій науковій практиці, яка рекомендує використовувати одну з запропонованих моделей або їх комбінацію під час аналізу текстової інформації, зокрема, у випадку наявності двох класів (фейкові та нефейкові дані), або при перевірці достовірності зображень.

Ключові слова: класифікація, CNN, RNN, ефективність, нейронні мережі.

Kostiantyn ONYSHCHENKO

Senior Lecturer at the Department of Software Engineering, Kharkiv National University of Radio Electronics, Nauky ave., 14, Kharkiv, Ukraine, 61166, kostiantyn.onyshchenko@nure.ua

ORCID: 0000-0002-7746-4570

Scopus Author ID: 57209338164

Anatolii AFANASIEV

Student at the Department of Informatics, Kharkiv National University of Radio Electronics, Nauky ave., 14, Kharkiv, Ukraine, 61166, anatolii.afanasiev@nure.ua

ORCID: 0009-0005-0707-981X

To cite this article: Onyshchenko, K., Afanasiev, A. (2023). Otsinka efektyvnosti vykorystannia zghortkovykh ta rekurentnykh neironnykh merezh u zavdanni obrobky tekstovykh danykh [Evaluation of the effectiveness of using convolutional and recurrent neural networks in the task of text data processing].

EVALUATION OF THE EFFECTIVENESS OF USING CONVOLUTIONAL AND RECURRENT NEURAL NETWORKS IN THE TASK OF TEXT DATA PROCESSING

This work is dedicated to evaluating the effectiveness of using convolutional and recurrent neural networks in the task of processing textual data, specifically in detecting fake news. Currently, the efforts of the global community are focused on combating such misinformation as a whole, which underscores the relevance of the addressed issue. The problem of detecting fake news lies in accurately determining the authenticity of the information. The objective of this work is to compare the accuracy of identifying fake news between convolutional and recurrent neural network architectures, which incorporate a syntactic analysis model of article texts by forming news labels using TF-IDF and Word Embedding. To achieve this goal, an analysis of the application domain was conducted, and the key characteristics of this type of information were identified. The theoretical foundation of the selected architectures was examined, and their configurations were established in accordance with the defined task. An experimental environment was created for the practical implementation of the chosen types of neural networks. The relative effectiveness of using recurrent neural networks compared to convolutional ones was revealed, and potential scenarios were identified where the obtained results may vary. As a result of the analysis, it was determined that the convolutional neural network performs computations faster than the recurrent neural network on the available data, but it provides less accurate classification results. Taking into account the proposed rule for performance comparison, the marginal increase in productivity can be considered insignificant, considering the probabilities of different types of errors and the potential for resolving discrepancies between algorithms. This conclusion aligns with the global scientific practice, which recommends using one of the proposed models or their combination for analyzing textual information, particularly when dealing with two classes (fake and non-fake data) or verifying the authenticity of images.

Key words: classification, CNN, RNN, efficiency, neural networks.

Вступ. Відповідно до Кембриджського словника фейкові новини – це історії, які виглядають як новини, поширюються в інтернеті або за допомогою інших засобів масової інформації, та, зазвичай, створені для впливу на політичні погляди або як жарт (Cambridge Dictionary, 2023). Насправді, їх історія доволі розлога, однак зі зростанням популярності соціальних мереж, особливо анонімних, ця проблема гостро постала перед світовим суспільством. У якості її каталізатора виступає ще і розвиток технологій спрямованих на коригування відео- та аудіо-інформації, і технологій створення ботів.

Станом на сьогодні кількість фейків зростає як в абсолютному, так і відносному значенні. Деякі з них можна віднести до неправильного суб'єктивного сприйняття реальної інформації, однак суттєва кількість є результатом пропаганди (Wesolowski, 2022). Найбільшими каналами такої інформації є Twitter (BBC News, 2022) та Telegram (Ukraine Crisis Media Center, 2022).

Загалом для боротьби з подібними новинами було створено багато ініціатив, це зокрема французький закон проти маніпуляції інформацією (Blotman, 2019), який був ухвалений задля боротьби з дискредитацією іммігрантів та Європейського Союзу після оголошення результатів Брекзиту (Danzig, 2017). У цьому законі зазначено, що платформи, які перевищують певну кількість відвідувань на

день, повинні мати законного представника у Франції та публікувати свої алгоритми, при цьому необхідно повідомляти про будь-який спонсорований контент, публікуючи ім'я автора та сплачену суму. Закон також вимагає від суддів кваліфікувати фейкові новини за такими трьома критеріями: явність, свідоме поширення в масових масштабах, призведення до порушень громадського порядку або скомпрометування результатів виборів. При цьому початкове рішення приймає спеціально створений комітет з етики.

Створення правової бази для регуляції фейкової інформації спостерігається і в Україні, зокрема це стаття 259 Кримінального кодексу, що регламентує відповідальність за завідомо неправдиве повідомлення про загрозу безпеці громадян (Закон України № 1292-IX, 2021).

Світова тенденція спрямована на боротьбу з подібною інформацією в цілому є позитивною, однак проблема полягає у визначенні того чи є інформацією фейковою чи ні. Якщо рішення базуватиметься не на фактах, а на експертній оцінці, то таку ситуацію можна розглядати як маніпулятивну і спрямовану на авторитаризацію країни.

Для демократичних країн процес визначення фейкових новин можна автоматизувати з використанням штучного інтелекту. Початково у якості базового методу застосовували наївний Байес (Jain, 2022) чи SVM

(Reddy, 2019, с. 945–946), однак з розвитком нейромереж найбільшої популярності здобули згорткові та рекурентні нейромережі.

Варто зазначити, що з огляду на численні дослідження, ці моделі можуть давати різний результат в залежності від предметної області (Pai, 2020). Тому було прийнято рішення розглянути ефективність використання цих моделей для визначення фейкових новин, з метою оптимізації їх вибору при побудові відповідної програмної системи.

Аналіз предметної галузі та наявних рішень. У першу чергу завдання визначення фейкових новин стосується процесу обробки природної мови (Natural Language Processing, NLP). Розглянемо існуючі рішення для вирішення задач NLP.

TF-IDF характеристика. Програмні системи не здатні напряму обробляти текстову інформацію, тому першочергово кожен текст має набути кількісного вигляду. Існує декілька основних методик подібної конвертації, однак наразі базовою є TF-IDF (Stecanella, 2019). TF – скорочення від term frequency – це частота кожного вживаного слова (точніше набору слів, які більш доцільно називати термінами). IDF – скорочення від inverse document frequency – це зворотне число термінів на документ. В цілому показник TF-IDF вказує на те, наскільки рідкісним є певний термін. Наприклад, вставні слова, сполучники чи вигукі будуть найпоширенішими, а, відповідно, матимуть низький TF-IDF.

Саме по собі отримання TF-IDF характеристики дозволяє лише ранжувати ті чи інші терміни, однак не перетворити текст у числове представлення. Для виконання цієї дії існує технологія Word Embedding, яка здійснює відображення слів чи фраз у вектори дійсних чисел. Формальна зазначена технологія є набором різноманітних методів, одним з яких є GloVe (Global Vectors). GloVe – це алгоритм для перетворення немаркованих даних (у нашому випадку термінів), у безперервні вектори, для зменшення розмірності (Pennington, 2022). Вектори GloVe попередньо навчені на даних з Вікіпедії та Gigaword 5, тож доволі добре вловлюють семантику речень. Варто зауважити, що цей алгоритм спрямований саме на тексти англійською мовою, а не українською.

Розглянемо як саме відрізняється фейкова новина від справжньої (без прив'язки до архітектурних моделей нейромереж). Задля демонстрації скористаємось набором даних щодо президентських виборів у 2016 році в Сполучених Штатах (Risdal, 2016). Він містить 20 015 новин, з яких 11 941 фейкові,

а 8074 справжні. Справжніми новинами у цьому набору виступають дані з відомих авторитетних веб-сайтів новин, наприклад, New York Times, Washington Post тощо. Вони містять багато різноманітної інформації, зокрема назва статті, текст, зображення, автор, веб-сайт та багато чого іншого. У подальшому будемо використовувати лише назву та текст. Розглянемо TF-IDF характеристику для заголовків, аби продемонструвати різницю між справжніми та фейковими новинами (рис. 1). Якщо заголовку немає, то встановлюємо його значення рівне «notitle».

Заголовки фейкових новин часто містять такі слова, як «notitle», «IN», «THE», «CLINTON» і багато незв'язних цифр, що представляють спеціальні символи. З цього можна зробити декілька цікавих висновків. По-перше, більшість фейкових новин не мають заголовків. Ці фейкові новини широко поширюються у вигляді твітів із кількома ключовими словами та гіперпосиланнями на новини в соціальних мережах. По-друге, у фейках більшість символів у великому регістрі. Мета – привернути увагу читачів, а справжні новини містять менше великих літер та загалом пишуться у стандартному форматі. По-третє, справжні новини містять більш детальні описи. Наприклад, імена (Джеб Буш, Мітч МакКоннелл тощо) і дієслова (left, claim, debate, poll тощо). Тепер для більш глибокого розуміння розглянемо проблему пошуку фейків з точки зору обчислювальної лінгвістики, психологічного позиціонування, лексичного різноманіття та аналізу настроїв.

Обчислювальна лінгвістика. Перше, що варто зауважити, фейкові новини в середньому містять менше слів, ніж справжні новини: 4360 слів проти 3943 (84 речення проти 69). Крім того діапазон слів у несправжніх новинах значно більший за реальні дані. Незважаючи на меншу кількість символів та речень, самі речення у фейкових новинах коротші. Це пов'язано з тим, що редактори та журналісти користуються певними газетними нормами, що включають довжину та вибір слів, відсутність граматичних помилок тощо. Натомість фейкові новини на цих правилах, зазвичай, не ґрунтуються. Відповідно до отриманих статистичних даних, справжні новини мають менше знаків питання, ніж фейкові. Причини можуть полягати в тому, що у фейкових новинах багато риторичних питань, які використовуються, щоб свідомо підкреслити ідеї та підсилити настрої. Окрім цього, можна зауважити, що обидва типи новин мають дуже мало вигуків, а, отже, і знаків оклику, хоча їхня кількість все ж більша саме у фейкових новинах.

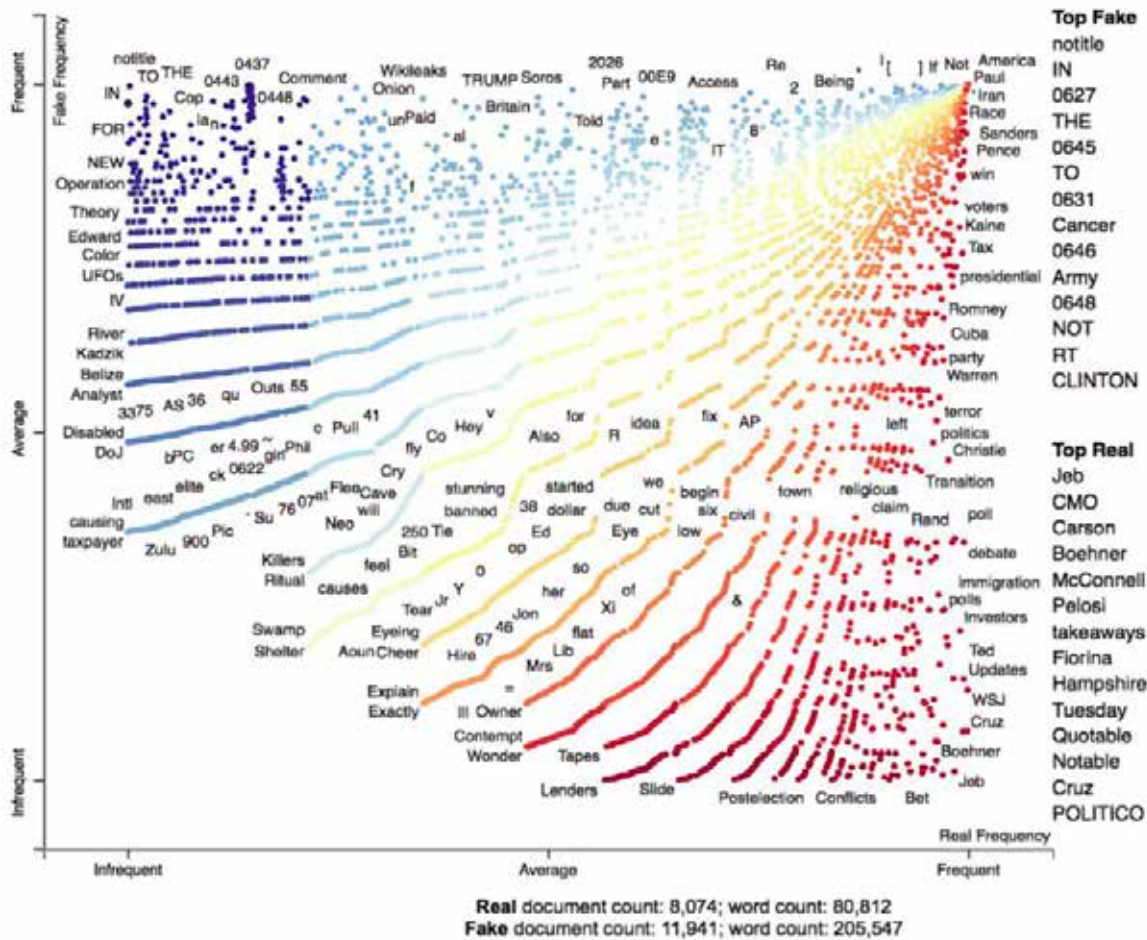


Рис. 1. TF-IDF характеристика для слів із заголовків

Постановка завдання. З’ясувавши основні нюанси, які можуть вплинути на перебіг дослідження формалізуємо мету роботи.

Дано набір з m текстових новин, який можна зобразити наступним чином:

$$\overset{\circ}{A} = \{A_i^T\}_i^m.$$

У задачі виявлення фейкових новин необхідно передбачити, чи є статті з $\overset{\circ}{A}$ фейковими чи ні. При цьому набір міток, що вказуватимуть на правдивість інформації можна представити у наступному вигляді:

$$y = \{1, 0\}_i^m,$$

де 1 – позначає справжню новину, а 0 – фейкову.

Набір функцій X_i^T з $A_i^T \in \overset{\circ}{A}$ має отримуватися шляхом синтаксичного аналізу текстів статті, зокрема з використанням TF-IDF та Word Embedding.

Метою чинної роботи є порівняння точності визначення фейкових новин для архітектур

CNN та RNN, в яких закладена наступна модель формування міток новин з $\overset{\circ}{A}$:

$$f : \{X_i : T\}_i^m \in \mathbb{X} \rightarrow y.$$

Дослідження ефективності. Перед тим як безпосередньо визначити експериментальні умови, наведемо опис алгоритму відповідно до якого здійснюватиметься підготовка даних до аналізу. Спочатку з тексту видаляється всі не буквені вирази (наприклад, числа, коми, крапки та інші розділові знаки) за допомогою бібліотеки *re*, яка дає доступ до регулярний виразів. Після очищення текст потрапляє до функції обробки (рис. 2). Основними в цій функції є методи, що надаються бібліотекою *nlTK*, яка дозволяє обробляти природну мову, виходячи із вбудованих корпусів слів. Першочергово необхідно видалити з набору ті слова, що не несуть інформаційного навантаження (наприклад, «and», «or» тощо). Ці слова заважатимуть коректному проведенню аналізу.

Наступним кроком необхідно нівелювати лінгвістичну варіативність, що породжується

використанням морфем. Для цього застосовується операція стемінгу – процес скорочення слова до його основи. Наприклад, слова «eating» та «eaten» будуть замінені на слово «eat». Після цієї обробки масив слів скорочується до набору основ.

Вбудований в бібліотеку *nlk* клас, що здійснює зазначений процес, не завжди працює коректно, тому для покращення результату було вирішено поєднати стемінг з іншою операцією – лематизацією. Вона дозволяє привести словоформу до леми (нормальної словникової форми). Наприклад, при обробці слова «good» та «better» матимуть різні основи, однак лема у цих слів одна – «good». Після проведення зазначених дій набір слів потребує ще однієї перевірки на неінформативний вміст.

Після обробки тексту, утворюється словник необхідний для коректного визначення TF-IDF характеристики. Задля врахування емоційного забарвлення слів наступним кроком є знаходження для кожного слова частотно-полярної характеристики, шляхом перемноження показників TF-IDF та полярності. Останній можна отримати за допомогою використання класу *SentimentIntensityAnalyzer*, що знаходиться в модулі *vader* бібліотеки *nlk*.

Ефективність певної моделі (E) будемо визначати наступним чином:

$$E = f(\vartheta, RMSE, v),$$

де ϑ – швидкість тренування, $RMSE$ – корінь середньоквадратичної похибки, що описує точність класифікації, v – обсяг даних. Показ-

ник швидкодії можна визначити за допомогою сторонніх модулів чи програмного забезпечення після імплементації моделей. Наприклад, модуль *time* чи програмні системи *Postman*, *JMeter*.

Для визначення точності прогнозу скористаємось спеціальними вибірками з даними щодо російського вторгнення в Україну (DS-Pr1nce, 2022) та виборів президента Сполучених Штатів у 2020 році (Dedhia, 2022). Вказані дані будуть поділені за принципом Парето (Алесу, 2010, с. 4) на дві групи у співвідношенні 80 до 20. 80 % використовуватимуться у якості вхідних даних для здійснення прогнозу. 20 % натомість слугуватимуть реальним значенням. Після класифікації для знаходження кореня середньоквадратичної похибки достатньо скористатися наступною формулою:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2},$$

де N – кількість класифікованих значень; y_i – реальне значення показника; \hat{y}_i – отримане значення показника. В якості обсягу даних розглядатимемо час, який витрачається користувачем для того, щоб підготувати модель до використання. При цьому, аби коректно порівняти дві моделі розглядатиметься обернений показник часу.

Правило порівняння ефективності. Задля можливості порівняння ефективності двох моделей введемо змінну C , що визначається наступною формулою:

```

1  def process_text(text, language):
2      stemmer = PorterStemmer()
3      lemmatizer = WordNetLemmatizer()
4      stop_words =
5          set(stopwords.words(
6              get_language_name(language)
7          ))
8      sentences = nltk.sent_tokenize(text)
9      sentences_processed = []
10     for sentence in sentences:
11         words_processed = []
12         words = nltk.word_tokenize(sentence)
13         words = [word for word in words if word not in stop_words]
14         for word in words:
15             word = stemmer.stem(word)
16             word = lemmatizer.lemmatize(word)
17             words_processed.append(word)
18         words =
19             [word for word in words_processed if word not in stop_words]
20         sentences_processed.append(" ".join(words))
21     text_processed = " ".join(sentences_processed)
22     return text_processed

```

Рис. 2. Код функції обробки

$$C_{AB} = \sum \text{normalize} \left(\frac{m_A}{m_B} \right),$$

де m_A – значення метрики для моделі A , m_B – значення метрики для моделі B .

Отримавши результати для кожної моделі, можна, використовуючи нечіткий параметр C , встановити яка модель є більш ефективною:

- $C_{AB} \geq 3.5$: модель A є більш ефективною за модель B ;
- $C_{AB} \leq 2.5$: модель A є менш ефективною за модель B ;
- $2.5 \leq C_{AB} \leq 3.5$: значення ефективності обох моделей є близьким один до одного.

Числові дані повинні бути зібрані для всіх метрик кожної моделі, тому використовується метод контрольованого експерименту. У наслідок цього було обрано стабільне та постійне середовище виконання – фізичний пристрій на базі *Ubuntu*. Для визначення швидкодії кожна модель імплементується за допомогою *Python 3*. Аби уникнути проблем при мануальному тестуванні та спростити процес налагодження прийнято рішення використати програмне середовище *Jupyter* та модуль *time* для оцінки швидкодії.

Як зазначалося раніше для визначення точності використовуються дві тестові вибірки. Ці дані розподіляються на дві частини у співвідношенні 80/20. Результат точності отриманий як співвідношення отриманого класу та реального для двох вибірок агрегується. Заключним показником для аналізу є швидкість підготовки даних. У цьому випадку показник залежить виключно від людини, що здійснює тестування.

Для цього експерименту можна визначити наступні помилки та невизначеності:

- при перевірці швидкості роботи варто очікувати похибки та помилки пов'язані з модулем оцінки часу виконання *time* та програмним середовищем *Jupyter*;
- при перевірці точності роботи моделей можуть виникнути проблеми із даними, що використовуються в якості тестової вибірки, оскільки вони прямо впливають на отриманий результат;
- при перевірці швидкості підготовки даних для аналізу можна виокремити дві основні проблеми: людський фактор та похибка інструменту вимірювання часу.

Задля зменшення впливу вказаних проблем було вирішено виконати 10 замірів швидкодії для звичайної та модифікованої моделей, по одному заміру для визначення точності результатів для кожної тестової вибірки та по 5 замірів для визначення часу підготовки до

аналізу. Для кожної метрики після проведення відповідних дій передбачається агрегація результатів

Аналіз результатів. Перед тим, як безпосередньо порівняти ефективність роботи двох алгоритмів, наведемо результати їх тренування – графіки, що зображають порівняння втрат (*loss*) з кількістю епох. Для CNN маємо наступні результати (рис. 3). Як видно з вказаного рисунку найкращою кількістю епох є трохи більше 1-єї.

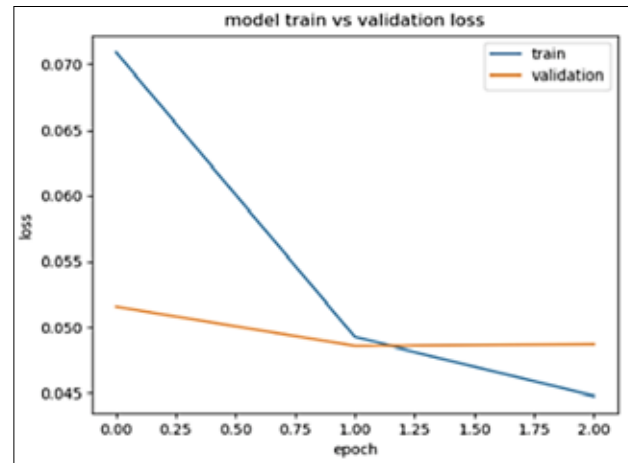


Рис. 3. Порівняння *loss* з кількістю епох для CNN

Перейдемо до наступної моделі – RNN (точніше LSTM). Для неї отримуємо наступний результат (рис. 4). Як бачимо у цьому разі оптимальна кількість епох рівна 1.

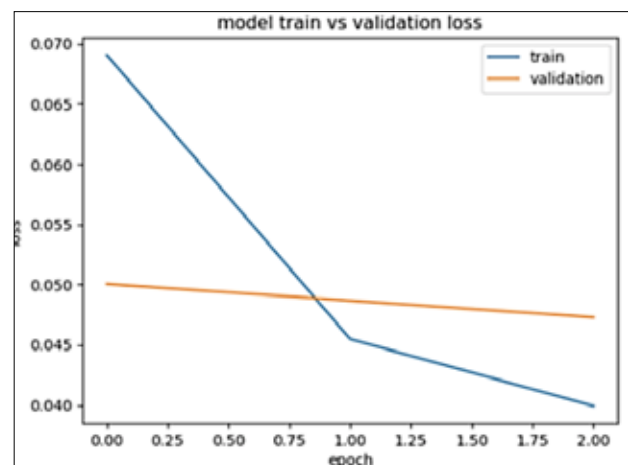


Рис. 4. Порівняння *loss* з кількістю епох для RNN

Перейдемо до визначення параметрів ефективності та почнемо із замірів швидкості тренування за допомогою бібліотеки *time*.

Результати наведено на рисунку 5. Знайдемо середнє значення для кожного з алгоритмів. Маємо для CNN значення часу близько 301 с, для RNN – 319 с.

Перейдемо до наступного показника – точність класифікації. Як зазначалося вище точність прогнозу вимірюється за допомогою показника *RMSE*. Маємо наступну точність отриманих результатів для обох вибірок: CNN (війна) – 93 %, CNN (вибори) – 94 %, RNN (війна) – 94 %, RNN (вибори) – 96 %.

Заключним показником є час підготовки даних для роботи архітектурних моделей. Результати замірів для даного показника наведені на рисунку 6.

Знайдемо середнє значення показника для кожного з алгоритмів. Маємо для базового алгоритму – 391 с, для модифікованого алгоритму – 391 с.

Знайшовши відповідні метрики для порівняння ефективності двох алгоритмів перейдемо до знаходження показника C_{AB} . У якості моделі *A* розглядатимемо CNN, а моделі *B* – RNN. Маємо $C_{AB} \approx 2.83$. Це дозволяє констатувати вищу ефективність RNN проти CNN, хоча ця відмінність неістотна. Даючи вигравш у швидкості тренування CNN програє у точності прогнозування. У випадку імплементації алгоритмів в рамках хмарної платформи, яка дозволяє значно пришвидшити швидкість роботи, отримана різниця в швидкодії між моделями буде неістотною. Однак варто зауважити, що точність може змінюватися в залежності від даних та їх обсягів.

Висновки. Дана робота мала на меті дослідження ефективності використання нейромережових архітектур типу CNN та RNN для виявлення фейкових новин. Задля цього була проаналізована предметна галузь пов'язана з неправдивою інформацією та виявлені ключові паттерни, які характеризують подібні дані. Окрім цього було проаналізовано теоретичне підґрунтя обох типів моделей. У ході чого виявлено, що стандартні моделі не можуть ефективно застосовуватися для текстової класифікації, тому було прийнято рішення для подальшого розгляду мати на увазі наступні положення:

- під CNN розуміти згорткову нейромережу з одношаровою згорткою і відповідним налаштуванням усіх її параметрів;
- під RNN розуміти налаштовану для аналізу тексту мережу LSTM, що має як довготривала, так і короткочасну пам'ять.

Наступним кроком стало визначення експериментального середовища, що включало себе:

- визначення функції ефективності;
- визначення правила порівняння ефективності двох моделей;
- план експерименту.

У якості тестових вибірок вирішено обрати дані, що стосується російського вторгнення на територію України та президентських виборів у Сполучених Штатах в 2020 році.

У результаті було виявлено, що на представлених даних модель CNN працює швидше за RNN, однак дає менш точний результат класифікації. З огляду на запропоноване правило порівняння ефективності, зазначені

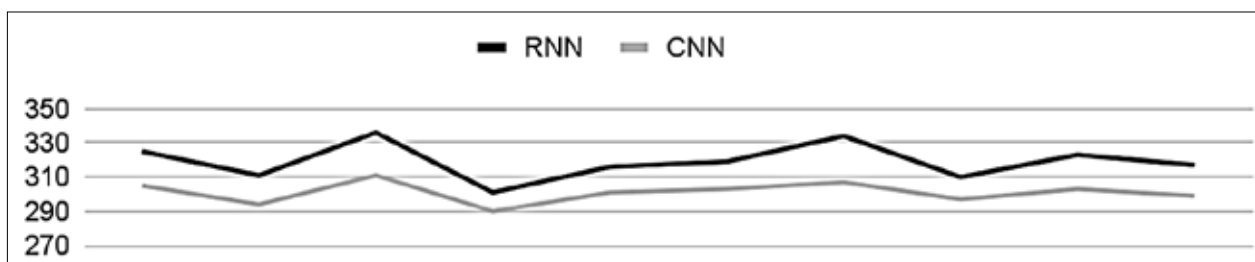


Рис. 5. Швидкість тренування нейромереж

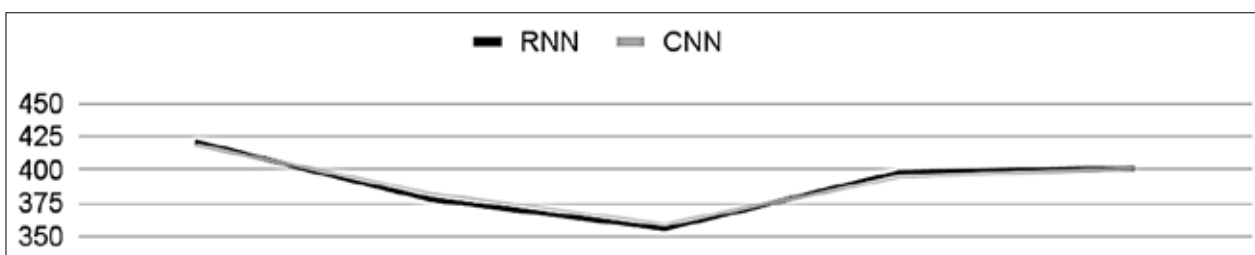


Рис. 6. Швидкість підготовки вхідних даних

імовірності похибок різного роду та шляхи подолання диспропорції між алгоритмами, отриманий вигравш ефективності можна вважати неістотним.

Зазначений висновок відповідає світовій науковій практиці, яка рекомендує під час

аналізу текстової інформації, особливо, за наявності двох класів (фейкові та не фейкові дані у чинному випадку), використовувати будь-яку із запропонованих моделей, або їх конгрегації у разі наявності перевірки зображень на достовірність (Saran, 2021).

ЛІТЕРАТУРА:

1. Fake News. *Cambridge Dictionary*. URL: <https://dictionary.cambridge.org/dictionary/english/fake-news> (дата звернення: 05.04.2023).
2. Wesolowski K. Fake news further fogs Russia's war on Ukraine. *Deutsche Welle*. URL: <https://www.dw.com/en/fact-check-fake-news-thrives-amid-russia-ukraine-war/a-61477502> (дата звернення: 08.04.2023).
3. Twitter steps up Ukraine misinformation fight. *BBC News*. URL: <https://www.bbc.com/news/business-61512261> (дата звернення: 08.04.2023).
4. Hybrid Warfare Analytical Group. How russian disinformation works in Telegram. *Ukraine Crisis Media Center*. URL: <https://uacrisis.org/en/ros-dezinfo-telegram> (дата звернення: 08.04.2023).
5. Blocman A. Laws to combat manipulation of information finally adopted. *IRIS Merlin*. URL: <https://merlin.obs.coe.int/article/8446> (дата звернення: 09.04.2023).
6. Danzig J. How fake news caused Brexit. *UACES*. URL: <https://europe.ideason europe.eu/2017/11/14/fake-news-caused-brexite/> (дата звернення: 09.04.2023).
7. Про внесення змін до статті 259 Кримінального кодексу України щодо посилення відповідальності за завідомо неправдиве повідомлення про загрозу безпеці громадян : Закон України від 02.03.2021 р. № 1292-IX. URL: <https://zakon.rada.gov.ua/laws/show/1292-20> (дата звернення: 09.04.2023).
8. Classifying Fake News Detection Using SVM, Naive Bayes and LSTM / P. Jain та ін. 2022 *12th International Conference on Cloud Computing, Data Science & Engineering*, м. Noida, 27–28 січ. 2022 р. URL: <https://doi.org/10.1109/Confluence52989.2022.9734129> (дата звернення: 20.04.2023).
9. A Study on Fake News Detection Using Naive Bayes, SVM, Neural Networks and LSTM / P. S. Reddy та ін. *Journal of Advanced Research in Dynamical and Control Systems*. 2019. Т. 11, № 06. С. 942–947.
10. Pai A. CNN vs. RNN vs. ANN – Analyzing 3 Types of Neural Networks in Deep Learning. *Analytics Vidhya*. URL: <https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/> (дата звернення: 02.05.2023).
11. Stecanella B. Understanding TF-IDF: A Simple Introduction. *Monkey Learn*. URL: <https://monkeylearn.com/blog/what-is-tf-idf/> (дата звернення: 08.09.2023).
12. Pennington J., Socher R., Manning C. D. GloVe: Global Vectors for Word Representation. *Stanford Edu*. URL: <https://nlp.stanford.edu/projects/glove/> (дата звернення: 15.05.2023).
13. Risdal M. Getting Real about Fake News. *kaggle*. URL: <https://www.kaggle.com/datasets/mrisdal/fake-news> (дата звернення: 16.05.2023).
14. Lying words: Predicting deception from linguistic styles. / M. L. Newman та ін. *Personality and social psychology bulletin*. 2003. Т. 29, № 5. С. 665–675.
15. IBM Cloud Education. Natural Language Processing (NLP). *IBM Cloud Learn Hub*. URL: <https://www.ibm.com/cloud/learn/natural-language-processing> (дата звернення: 20.05.2023).
16. DS-Pr1nce. War in Ukraine: Russian social network discussions. *kaggle*. URL: <https://www.kaggle.com/datasets/ustyk5/war-in-ukraine-russian-social-network-discussions> (дата звернення: 25.05.2023).
17. Dedhia R. Fake News. *kaggle*. URL: <https://www.kaggle.com/datasets/ronikdedhia/fake-news> (дата звернення: 02.06.2023).
18. Alecu F. The Pareto Principle in the Modern Economy. *Economics of Knowledge*. 2010. Т. 2, № 3. С. 2–5.
19. Saran A. Text Classification – CNN with LSTM. *Medium*. URL: <https://anandsarank.medium.com/cnn-with-lstm-for-text-classification-53d18e5f7f5c> (дата звернення: 11.06.2023).

REFERENCES:

1. Fake News. (n.d.). Cambridge Dictionary. <https://dictionary.cambridge.org/dictionary/english/fake-news>
2. Wesolowski, K. (2022, April 17). Fake news further fogs Russia's war on Ukraine. dw.com. <https://www.dw.com/en/fact-check-fake-news-thrives-amid-russia-ukraine-war/a-61477502>
3. BBC News. (2022, May 19). Twitter steps up Ukraine misinformation fight. <https://www.bbc.com/news/business-61512261>

4. *How russian disinformation works in Telegram*. (n.d.). Uacrisis.org. <https://uacrisis.org/en/ros-dezinfo-telegram>
5. Blocman, A. (n.d.). Laws to combat manipulation of information finally adopted. IRIS Merlin. <https://merlin.obs.coe.int/article/8446>
6. Danzig, J. (2017, November 14). How fake news caused Brexit. UACES. <https://europe.ideason europe.eu/2017/11/14/fake-news-caused-brexiteu/2017/11/14/fake-news-caused-brexiteu>
7. Pro vnesennia zmin do statti 259 Kryminalnoho kodeksu Ukrainy shchodo posylennia vidpovidalnosti za zavidomo nepravdyve povidomlennia pro zahrozu bezpetsi hromadian, Zakon Ukrainy №. 1292-IX [On making changes to Article 259 of the Criminal Code of Ukraine regarding increased liability for knowingly false notification of a threat to the safety of citizens: Law of Ukraine No. 1292-IX] (2021). <https://zakon.rada.gov.ua/laws/show/1292-20#Text> [in Ukrainian].
8. Jain, P., Sharma, S., Monica, & Aggarwal, P. K. (2022). Classifying Fake News Detection Using SVM, Naive Bayes and LSTM. In *2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE. <https://doi.org/10.1109/confluence52989.2022.9734129>
9. Reddy, P. S. (2019). A Study on Fake News Detection Using Naive Bayes, SVM, Neural Networks and LSTM. *Journal of Advanced Research in Dynamical and Control Systems*, 11(6), 942–947.
10. Pai, A. (2020, February). *CNN vs. RNN vs. ANN – Analyzing 3 Types of Neural Networks in Deep Learning*. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning>
11. Stecanella, V. (n.d.). *Understanding TF-IDF: A Simple Introduction*. MonkeyLearn Blog. <https://monkeylearn.com/blog/what-is-tf-idf/>
12. Pennington, J., Socher, R., & Manning, C. D. (n.d.). *GloVe: Global Vectors for Word Representation*. Stanford Edu. <https://nlp.stanford.edu/projects/glove/>
13. Risdal, M. (n.d.). *Getting Real about Fake News*. kaggle. <https://www.kaggle.com/datasets/mrisdal/fake-news>
14. Newman, M. L. (2003). Lying words: Predicting deception from linguistic styles. *Personality and social psychology bulletin*, 29(5), 665–675.
15. IBM Cloud Education. (n.d.). *Natural Language Processing (NLP)*. IBM Cloud Learn Hub. <https://www.ibm.com/cloud/learn/natural-language-processing>
16. DS-Pr1nce. (n.d.). *War in Ukraine: Russian social network discussions*. kaggle. <https://www.kaggle.com/datasets/ustyk5/war-in-ukraine-russian-social-network-discussions>
17. Dedhia, R. (n.d.). *Fake News*. kaggle. <https://www.kaggle.com/datasets/ronikdedhia/fake-news>
18. Alecu, F. (2010). The Pareto Principle in the Modern Economy. *Economics of Knowledge*, 2(3), 2–5.
19. Saran, A. (n.d.). *Text Classification – CNN with LSTM*. Medium. <https://anandsarank.medium.com/cnn-with-lstm-for-text-classification-53d18e5f7f5c>