Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра прикладної математики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Методи машинного навчання
розв'язання задачі ідентифікації текстів
(тема)

Виконав:
студент _2_ курсу, групи _ПМм-20-1
Подшиваленко Б.О.
(прізвище, ініціали)
Спеціальність
113 Прикладна математика
(код і повна назва спеціальності)
Тип програми _освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма
Прикладна математика
(повна назва освітньої програми)
Керівник ____доц. Гибкіна Н.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ПМ _____ ____Тевяшев А.Д._____
(підпис) (прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет _інформаційно-аналітичних технологій та менеджменту
Кафедра _прикладної математики
Рівень вищої освіти _другий (магістерський)
Спеціальність _113 Прикладна математика
(код і повна назва)
Тип програми _освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма _Прикладна математика
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри ПМ _____
(підпис)
“____” _____ 2021 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _Подшиваленку Борису Олександровичу
(прізвище, ім’я, по батькові)

1. Тема роботи _Методи машинного навчання розв’язання задачі ідентифікації текстів
затверджена наказом по університету від 05 листопада 2021 р. № 1641 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 10 грудня 2021 р.

3. Вихідні дані до роботи _набір текстів художніх творів англомовної літератури

4. Перелік питань, що потрібно опрацювати в роботі
1. Аналіз предметної області
2. Вибір і обґрунтування методу розв’язання
3. Програмна реалізація
4. Результати обчислювального експерименту

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій

1. Актуальність теми роботи
2. Постановка задачі
3. Аналіз предметної області
4. Метод чисельного аналізу
5. Результати обчислювального експерименту

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Підбір та вивчення технічної літератури за темою роботи	8 – 14 листопада 2021 р.	виконано
2	Вибір та обґрунтування методу	15 – 21 листопада 2021 р.	виконано
3	Розробка алгоритму і програми	22 – 28 листопада 2021 р.	виконано
4	Проведення аналітичних досліджень та розрахунків	29 листопада – 5 грудня 2021 р.	виконано
5	Робота над текстом пояснювальної записки	6 – 9 грудня 2021 р.	виконано
6	Представлення роботи на рецензію в ЕК	10 грудня 2021 р.	виконано

Дата видачі завдання _8_ листопада 2021 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Гибкіна Н.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 46 с., 2 табл., 5 рис., 1 дод., 10 джерел.

МАШИННЕ НАВЧАННЯ, ПОПЕРЕДНЯ ОБРОБКА ДАНИХ, МАСКИ, НЕЙРОННА МЕРЕЖА, ФОРМАТУВАННЯ, КЛАСИФІКАЦІЯ, ІДЕНТИФІКАЦІЯ АВТОРА, ТЕКСТ, ТОКЕНІЗАЦІЯ.

Об’єкт дослідження – художні твори зарубіжних авторів.

Мета роботи – дослідження методів машинного навчання для ідентифікації автора тексту.

Методи дослідження – методи машинного навчання.

У роботі проведений аналіз проблеми ідентифікації автора тексту. Обрано оптимальний метод машинного навчання для вирішення поставленої задачі – за допомогою нейронної мережі.

Досліджено можливість застосування обраних методів до ідентифікації текстів. Розроблено програмний продукт, який надає можливість визначити ймовірність належності тексту одному з авторів з заданого переліку. Результати тестування подані у вигляді графіків для відображення результатів ідентифікації.

ABSTRACT

Introductory note: 46 pages, 2 tables, 5 figures, 1 appendix, 10 sources.

MACHINE LEARNING, PRE-PROCESSING DATA, MASKS, NEURAL NETWORK, FORMATTING, CLASSIFICATION, AUTHOR`S IDENTIFICATION, TEXT, TOKENIZATION.

Object of research – artistic products of foreign authors.

Purpose of work – research of machine learning methods of identifying the author of the text.

Methods of research – machine learning methods.

The analysis of the problem of identification of the author of the text is carried out in the work. The optimal method of machine learning was chosen to solve the problem – using a neural network.

The possibility of applying the selected methods to the identification of texts is investigated. A software application has been developed that provides an opportunity to determine the probability that a text belongs to one of the authors from a given list. The test results are presented in the form of graphs to display the results of identification.

ЗМІСТ

ВСТУП

Актуальність теми. Удосконалення комп'ютерної техніки та комп'ютерних мереж, широке використання засобів розпізнавання графічної та текстової інформації, збільшення дискового простору інтернет-серверів, що підтримують відповідні ресурси наукових та художніх бібліотек, соціальних мереж, банків даних тощо, призводить до того, що обсяги інформації, поданої у електронному і відцифрованому вигляді, постійно збільшуються.

Текстова інформація накопичується в усіх сферах людської діяльності і стає загальнодоступною для ознайомлення, а часто й для вільного копіювання. Це спрощує можливості для деяких авторів щодо використання у своїх роботах (художніх творах, наукових та публіцистичних статтях тощо) фрагментів або, навіть, цілих текстів чужого авторства без посилань на джерела, з яких було запозичено інформацію, і, у свою чергу, ставить під сумнів авторство як оригінального так і скомпонованого на його основі тексту, а також викликає питання щодо доброчесності. З іншого боку, цінність запозиченої інформації зменшується, а сам процес ідентифікації автора стає майже неможливим.

Проблему вільного використання запозиченого тексту неможливо вирішити повністю через сучасні уявлення про доступність інформації та оприлюднення результатів досліджень та досягнень у загальнодоступних джерелах, часто, навіть, без обмеження доступу. Але є можливість підсилювати відповідальність авторів за рівень оригінальності їх робіт шляхом аналізування робіт, що подаються до опублікування у відкритому доступі, на схожість з вже доступними джерелами інформації, і робити висновки щодо можливості або неможливості належності цих робіт певному автору.

Обробка текстів великого обсягу є неможливою без використання комп'ютерної техніки, отже, постійний розвиток програмного забезпечення, зокрема, для роботи з великими даними (в тому числі, й текстовими), поширює можливості щодо вирішення проблеми перевірки авторства. Для вирішення цієї проблеми традиційно застосовують спеціалізовані математичні та комп'ютерні

методи аналізу текстів. В основі більшості комп'ютерних процедур ідентифікації авторства лежать математичні методи аналізу, їх точність залежить від низки параметрів, а саме: мова, стиль, жанр, обсяг, автор. В наш час на заміну класичним методам ідентифікації прийшли машинні методи аналізу текстів, які за наявності великого набору даних дають змогу з високою точністю встановлювати авторство або класифікувати тексти за можливими авторами. Отже, насущним є питання аналізу якості засобів машинного навчання для вирішення задач щодо встановлення авторів робіт та дослідження характеристик, які дають змогу отримувати прийнятні результати.

Мета і завдання кваліфікаційної роботи. Метою кваліфікаційної роботи є розробка програмного забезпечення для ідентифікації та класифікації автора тексту. Для досягнення поставленої мети необхідно виконати наступні завдання:

– провести огляд і аналіз сучасного стану задачі ідентифікації автора тексту;

– обрати оптимальний метод машинного навчання;

– провести навчання обраного алгоритму;

– розробити програмне забезпечення для вирішення задачі ідентифікації авторів текстів.

Об'єктом дослідження є тексти обраного жанру літератури.

Предметом дослідження є ідентифікація автора тексту методами машинного навчання.

Методи дослідження. У кваліфікаційній роботі використовуються технології попередньої обробки тексту, методи машинного навчання, нейронні мережі.

Публікації. Результати, отримані у кваліфікаційній роботі, було представлено на 25-му Міжнародному молодіжному форумі «Радіоелектроніка та молодь у XXI столітті» (м. Харків, 20-22 квітня 2021 р.) [9].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Проблема ідентифікації текстів та методи її розв’язання

Існує багато методів аналізу тексту. Їх можна поділити на дві великі групи – експертні і формальні. В основу експертних методів покладено аналіз тексту професійними лінгвістами [3]. Ці методи є трудомісткими, вони потребують багато часу і наявності певної кількості кваліфікованих спеціалістів. Аналіз великих наборів даних експертними методами не є виправданим і доцільним, оскільки на нього витрачається багато людино-годин, а точність аналізу безпосередньо залежатиме від мови, жанру та обсягу досліджуваних матеріалів, професійного рівня спеціалістів та їх обізнаності з тематикою дослідження. Отже, доцільним є використання формальних методів ідентифікації .

Основа формальних методів аналізу текстів – математичні методи. Завдяки розвитку обчислювальних машин задача аналізу вийшла на новий рівень за точністю і надійністю результатів. Наразі для аналізу використовуються методи дискретного аналізу, математичної статистики, дерева рішень, нейронні мережі, методи кластерного аналізу та інші.

Існуючі системи ідентифікації текстів, побудовані на основі формальних методів, різняться, власно, методами ідентифікації автора, засобами аналізу тексту, необхідним обсягом тексту та точністю. Нижче наведено деякі з них [10]:

а) система «Лінгвоаналізатор»:

– методи: ланцюги Маркова, інформаційна ентропія;

– засіб аналізу тексту: статистичний аналіз частот графем;

– мінімальний обсяг тексту: 40 тисяч символів;

– точність: 84-89%;

б) система «Авторовед»:

– методи: нейронні мережі, метод опорних векторів;

– засіб аналізу тексту: статистичний аналіз частот триграм;

– мінімальний обсяг тексту: 20 тисяч символів;

– точність: 95-98%;

в) система «СМАЛТ» («Статистичні методи аналізу тексту»):

– метод: математична статистика;

– засіб аналізу тексту: морфологічний і синтаксичний аналіз;

– мінімальний обсяг тексту: 40 тисяч символів;

– точність: 78-88%.

Застосування формальних методів базується на відображенні тексту у вектор параметрів. Ці параметри відображують характерні особливості тексту. З точки зору математики це можна подати як відображення текстового документу у точку n-вимірного простору. Таким чином, автор, тексти якого досліджуються, може бути представлений вектором параметрів, побудованим на основі цих та/або інших текстів даного автора [10].

За такої постановки задача ідентифікації авторів текстів зводиться до задачі встановлення схожості між аналізованим текстом та деяким «зразком» або між двома аналізованими текстами. Як міра «схожості» може використовуватися «відстань» між векторами ознак для текстів, що порівнюються (або між вектором ознак аналізованого тексту та еталонним вектором ознак для певного автора).

В основі формальних методів ідентифікації лежить уявлення про те, що зі зростанням обсягу тексту параметри, які характеризують авторський стиль, стають стійкими з імовірнісної точки зору. Це дозволяє встановлювати авторство за формальними характеристиками тексту, які стабільно повторюються. Тому більш висока якість ідентифікації досягається для текстів великого обсягу, і менш точний результат виходить для текстів невеликого обсягу.

Відкритим залишається питання вибору авторського інваріанту (набору формальних параметрів тексту). Часто практично вирішується обмежене коло задач ідентифікації для попередньо заданого набору текстів. Налаштування, тестування та демонстрація інструментів аналізу орієнтовані тільки на ці тексти, і немає жодної гарантії, що методи ефективно справлятимуться із подібною за-

дачею на інших даних. Інакше кажучи, для побудови універсального і незалежного від текстів авторського інваріанту потрібно шукати нові шляхи формування параметрів.

За допомогою факторного аналізу та аналізу головних компонент можна встановити внесок тієї чи іншої характеристики у процес розпізнавання автора; ієрархічний кластерний аналіз дозволить зробити об'єднання окремих характеристик у підгрупи, підгрупи у групи тощо. Чималу допомогу можна отримати від нейронних мереж, якщо спробувати навчити мережу на наборі прикладів, взявши в якості входів окремі характеристики, а потім оцінювати, який вплив має той чи інший вхід на систему виходів.

Недостатньо досліджено залежність якості класифікації різними методами від обсягів фрагментів та від числа класів (тобто можливих авторів). Також вищенаведені системи аналізу текстів не орієнтовані на комплексне дослідження та порівняння стилів текстів (для різних задач аналізу стилів текстів з використанням різних методів їх розв'язання, різних частотних ознак, різного текстового матеріалу тощо).

До проблем, що ускладнюють дослідження у сфері ідентифікації текстів, належить також проблема складання вибірки еталонних текстів. Бажано, щоб твори були підібрані так: тексти різних письменників максимально різнилися один від одного, а тексти одного письменника були максимально близькими. Але нерідкою є ситуація, коли автор у певні періоди своєї творчості змінює стиль викладання або пише у різних жанрах або працює у співавторстві. Ці факти створюють додаткові складнощі під час вирішення задачі встановлення авторства. Тому проблему ідентифікації авторства не можна вважати вирішеною, існуючі методи та алгоритми потребують доопрацювання та оновлення з урахуванням нових досягнень у тих розділах прикладної математики та інформаційних технологій, які можуть бути використані під час обробки текстових даних.

1.2 Методи машинного навчання ідентифікації тексту

Алгоритми машинного навчання можна описати як навчання цільової функції f, що якнайкраще співвідносить вхідні змінні X і вихідну змінну Y: $Y = f(X)$.

Найбільш поширеною задачею у машинному навчанні є передбачення значень Y для нових значень X. Це називається прогностичним моделюванням, і мета цієї задачі – зробити якомога точніше передбачення. В залежності від того, які значення Y можливі: дійсні або з деякої скінченної множини, виділяють задачі регресії та класифікації. Оскільки задача ідентифікації текстів полягає у визначенні одного з авторів із деякого відомого переліку, як найімовірнішого для цього тексту, то її можна розглядати як задачу класифікації.

Розглянемо декілька популярних алгоритмів прогнозування та класифікації, що використовуються в машинному навчанні і можуть бути корисними для розв’язання задачі ідентифікації [2].

1.2.1 Лінійна регресія

Лінійна регресія – мабуть, один із найбільш відомих та зрозумілих алгоритмів у статистиці та машинному навчанні. Прогностичне моделювання насамперед стосується мінімізації помилки моделі або, іншими словами, здійснення якомога більш точного прогнозування. Лінійна регресія реалізується рівнянням, яке описує таку пряму, що найбільш точно відбиває взаємозв’язок між вхідними змінними і вихідними змінними. Для складання цього рівняння потрібно визначити значення коефіцієнтів для вхідних змінних. Для оцінки регресійної моделі використовуються різні методи на кшталт лінійної алгебри чи методу найменших квадратів [2].

Мета лінійної регресії – моделювання взаємовідносин між набором ознак і ціллю з неперервним значенням. Отже, маємо простір об’єктів X та множину ві-

дповідей Y, $Y \in \mathbb{R}$. Кожен об'єкт представлено вектором ознак $\vec{x} = (x_1, x_2, ..., x_m)$. Значення цільової змінної відомі на об'єктах тренувального набору даних. Необхідно побудувати алгоритм $a: X \to Y$, який прогнозує значення цільової змінної y для будь-якого $\vec{x} \in X$. Рівняння лінійної моделі виглядає наступним чином:

$$a(\vec{x}) = \omega_0 + \sum_{i=1}^{m} \omega_j x_j , \qquad (1.1)$$

де ω_j, $j = \overline{1,m}$, – параметри моделі (коефіцієнти пояснювальних змінних);

ω_0 – коефіцієнт зсуву;

$x_1, x_2, ..., x_m$ – набір незалежних змінних.

Метою навчання моделі (1.1) є пошук таких вагових коефіцієнтів, які б дозволили найбільш точно описати наявний зв'язок між пояснюючими змінними та цільовою змінною. Пошук оптимальних значень вагових коефіцієнтів відбувається за допомогою методу найменших квадратів або задля скорочення часу обчислень – ітераційними методами оптимізації, зокрема, методом градієнтного спуску та його модифікаціями. Надалі отриману модель можна використовувати для прогнозування відповідей нових значень пояснюючих змінних, які не входили до складу тренувального набору даних.

Обмеженням на застосування лінійної регресії є неперервність значень вихідної змінної. Якщо ж вихідна змінна є категоріальною чи бінарною, для її прогнозування використовують логістичну регресію.

1.2.2 Логістична регресія

Логістична регресія – ще один статистичний алгоритм, що використовується у машинному навчанні. Логістична регресія добре підходить для класифікації об'єктів, кожен з яких належить одному з двох класів (бінарна класифіка-

ції), але може бути узагальнена й на випадок багатьох класів (багатокласова класифікація) [2].

Формально задача побудови логістичної регресії зводиться до наступного. Розглядається простір об'єктів X, кожен з яких представлено вектором ознак $\vec{x} = (x_1, x_2, ..., x_m)$, і множина відповідей Y, $Y = \{0,1\}$. Значення залежної змінної відповідають наявним класам: $y = 1$ – додатній клас, $y = 0$ – від'ємний клас. Як і раніше, значення цільової змінної відомі на об'єктах тренувального набору даних, і за цими даними необхідно побудувати алгоритм $a: X \to Y$, що передбачатиме значення цільової змінної y для нових об'єктів $\vec{x} \in X$.

Модель логістичної регресії має вигляд:

$$a(\vec{x}) = \frac{1}{1 + e^{-\sum_{i=0}^{m} \omega_j x_j}},$$

де ω_j, $j = \overline{0,m}$, – коефіцієнти логістичної регресії;

$x_1, x_2, ..., x_m$ – набір незалежних змінних ($x_0 = 1$).

Наведена вище функція набуває значень у інтервалі від 0 до 1 і називається логістичною.

Модель логістичної регресії дозволяє не лише відносити об'єкт до того чи іншого класу, а прогнозує ймовірність такої належності: $p_+ = P(y = 1 | \vec{x}) \in [0;1]$. Обернена до неї функція $\text{logit}\, P(y = 1 | \vec{x}, \vec{\omega}) = \ln\left(\frac{p_+}{1 - p_+}\right) = \vec{\omega}^T \vec{x} \in \mathbb{R}$ відбиває лінійний зв'язок між ознаками і логарифмом відношення шансів належності та неналежності до обраного класу.

Логістична регресія схожа на лінійну тим, що для її побудови також потрібно визначати оптимальні значення вагових коефіцієнтів для вхідних змінних. Різниця полягає в тому, що вихідне значення перетворюється за допомогою нелінійної чи логістичної функції. Оцінювання коефіцієнтів логістичної ре-

гресії здійснюється за допомогою методу максимальної правдоподібності, а не методу найменших квадратів.

Використання ймовірностей належності об’єктів тому чи іншому класу, які дозволяє прогнозувати логістична регресія, корисно у випадках, коли потрібно мати більше обґрунтувань щодо прийняття остаточних висновків відносно класифікації. Модель логістичної регресії швидко навчається та добре підходить для завдань бінарної класифікації.

1.2.3 Лінійний дискримінантний аналіз

Логістична регресія, як правило, використовується, коли потрібно віднести зразок до одного з двох класів. Якщо класів більше, ніж два, краще використовувати алгоритм лінійного дискримінантного аналізу LDA (Linear discriminant analysis) [2]. Постановка задачі LDA досить проста. Спочатку визначаються статистичні властивості даних, розрахованих за кожним класом. Для кожної вхідної змінної такими статистичними характеристиками є середнє значення кожного класу та дисперсія. Прогнози формуються шляхом обчислення дискримінантного значення для кожного класу та вибору класу з найбільшим значенням. Лінійний дискримінанстний аналіз є простим та ефективним алгоритмом для задач класифікації.

Нехай маємо точки $\vec{x}^{(1)}, \vec{x}^{(2)}, ..., \vec{x}^{(n)}$ у m-вимірному просторі, n_1 з яких належать класу c_1 та n_2 – класу c_2. Для одиничного вектору $\vec{v}$ збудуємо проекції цих точок на напрямок даного вектору. Тоді скалярний добуток $\vec{v}^T \vec{x}^{(i)}$ співпадає з відстанню від початку координат до проекції точки $\vec{x}^{(i)}$ на напрямок вектору $\vec{v}$. Інакше кажучи, $\vec{v}^T \vec{x}^{(i)}$ є проекцією вектора $\vec{x}^{(i)}$ на простір меншої розмірності. Оцінимо ступінь розділення проекцій різних класів. Нехай μ_1 та μ_2 – середні значення першого та другого класів, а $\tilde{\mu}_1$ та $\tilde{\mu}_2$ середні значення проекцій першого та другого класів. Тоді

$$\tilde{\mu}_1 = \frac{1}{n_1}\sum\{\vec{v}^T\vec{x}^{(i)} \mid \vec{x}^{(i)} \in c_1\} = \vec{v}^T\left(\frac{1}{n_1}\sum\{\vec{x}^{(i)} \mid \vec{x}^{(i)} \in c_1\}\right) = \vec{v}^T\mu_1$$

та $\tilde{\mu}_2 = \vec{v}^T\mu_2$. Величина $|\tilde{\mu}_1 - \tilde{\mu}_2|$ може бути використовуватися як міра для розподілу класів.

1.2.4 Дерева рішень

Дерева рішень є одним з найпопулярніших алгоритмів машинного навчання, який використовується для розв'язання задач не лише класифікації, а й регресії. Дерева являють собою ієрархічну структуру, у якій прогнозування здійснюється за допомогою послідовності правил певного виду, кожному з яких задовольняє чи не задовольняє розглядуваний об'єкт [2].

Дерево рішень можна представити у вигляді двійкового дерева. Кожен вузол пов'язаний з певною вхідною змінною і є точкою розділу зразків за цією змінною (за умови, що зміна – число). Листові вузли – вихідна змінна, яка використовується для прогнозування. Прогнози формуються у результаті проходу зразка по дереву від кореня до листового вузла певним шляхом та виведення значення класу на цьому вузлі.

Алгоритм побудови дерева рішень у загальному вигляді включає декілька етапів. На кожному етапі дані розділяються за тією ознакою, яка призводить до найбільшого прирощення інформації [7]. Якщо на якомусь етапі усі дані у поточному вузлі мають однакові мітки, то вузол є листом. Інакше розподіл продовжується у кожному дочірньому вузлі до отримання листів.

Дерева швидко навчаються і роблять прогнози. Крім того, вони показують достатню точність на широкому колі задач і не вимагають особливої підготовки даних. Ще більш потужні результати дозволяє отримати поєднання дерев рішень у так звані випадкові ліси, які представляють собою ансамблі методів.

1.2.5 Інші класифікатори

Наївний Баєс – це простий, але дуже ефективний алгоритм машинного навчання, у основі якого лежать методи класичної теорії ймовірностей, а саме, теорема Баєса [2].

Модель складається з двох типів ймовірностей, які розраховуються за тренувальними даними: ймовірності кожного класу та умовні ймовірності для кожного класу. Після побудови ймовірнісної моделі її можна використовувати для прогнозування на нових даних. Даний алгоритм називається наївним, тому що припускає, що кожна вхідна змінна (тобто окремі ознаки) незалежна. Це сильне припущення, яке часто не відповідає реальним даним. Проте цей алгоритм дуже ефективний для цілого ряду складних задач на кшталт класифікації спаму або розпізнання рукописних цифр [1].

Ще один простий і дуже ефективний алгоритм класифікації – метод k найближчих сусідів [7]. Прогнозування класу для кожної нової точки здійснюється шляхом пошуку її найближчих сусідів у наборі даних і узагальнення значень вихідної змінної для цих екземплярів. Питання полягає у тому, як визначати схожість між екземплярами даних. Якщо всі ознаки мають один масштаб, то найпростіший спосіб полягає у використанні евклідової відстані між об’єктами, яка враховує відмінності між зразками за кожною вхідною змінною. Модель KNN (k-nearest neighbors) для класифікації кожного нового об’єкта щоразу опрацьовує весь набір тренувальних даних [7].

Метод k найближчих сусідів може вимагати багато пам’яті для зберігання всіх даних. Також навчальні дані можна оновлювати, щоб прогнози залишалися точними, враховуючи змінення у тенденціях, які відбивають дані. Метод k найближчих сусідів може мати проблеми при обробці даних великої розмірності (тобто з великою кількістю вхідних змінних), що негативно впливає на його ефективність під час розв’язання задачі класифікації. Це ситуація називається прокляттям розмірності і для її усунення рекомендується використовувати для класифікації лише найважливіші для прогнозу зміні.

1.3 Нейронні мережі як метод розв’язання задач машинного навчання

Штучні нейронні мережі (ШНМ) – це великий клас систем, архітектура яких має аналогію з побудовою нервової тканини з нейронів. ШНМ складається з набору «нейронів», поєднаних між собою. Кожен нейрон є елементарним перетворювачем вхідних сигналів у вихідні. Вихідні сигнали обчислюються як функція вхідних сигналів. Як правило, передаточні функції всіх нейронів у мережі фіксовані, а ваги є параметрами мережі і можуть змінюватися. Деякі входи нейронів позначені як зовнішні входи мережі, а деякі виходи – як зовнішні виходи мережі. Подаючи будь-які числа на входи мережі, ми отримаємо певний набір чисел на виходах мережі. Таким чином, робота нейромережі полягає у перетворенні вхідного вектора у вихідний вектор, причому це перетворення задається вагами мережі [7].

Для того, щоб мережа розв’язувала поставлену задачу, її треба «натренувати» на даних, для яких відомі значення вхідних параметрів, і правильні відповіді на них. Тренування полягає у підборі ваг міжнейронних зв’язків, що забезпечують найбільшу близкість відповідей мережі до відомих правильних відповідей. Нейронні мережі мають дуже широкий спектр застосування. Зокрема, їх застосовують для розв’язання задач класифікації, регресії та кластеризації.

Архітектура ШНМ дозволяє ефективно розпаралелювати процеси навчання та прогнозування. Безпосередньо підбираючи структуру та параметри ШНМ у кожній окремій задачі, можна досягти високої або хоча б прийнятної точності навіть у тих випадках, коли класичні класифікатори не справляються. Але підбір конфігурації ШНМ стикається з труднощами, оскільки заздалегідь не завжди можна сказати, на які параметри слід орієнтуватися. Їх обирають або спираючись на досвід аналогічних реалізацій або експериментально, що часто суттєво збільшує час, що витрачається на розробку. З цієї точки зору важливим етапом побудови ШНМ для конкретного набору даних є попередня обробка цих даних та відбір значущих ознак, за якими надалі й проводитиметься навчання та подальші передбачення.

Існує ряд експериментів з використанням нейронних мереж для класифікації текстів, але зазначається, що на навчання таких мереж витрачається дуже великий час [2]. Це пов’язано з тим, що для задач високої розмірності потрібна ШНМ з великою кількістю вузлів.

1.4 Змістовна та формальна постановка задачі

1.4.1 Змістовна постановка задачі

Метою даної кваліфікаційної роботи є розв’язання задачі ідентифікації текстів (зокрема, ідентифікування автора окремого тексту та аналізу двох текстів на предмет можливого спільного їх авторства) методами штучних нейронних мереж.

Особливістю художніх творів є характерний авторський стиль написання, але він може різнитися навіть для одного автора для творів, написаних у різний час або у різних стилях. Через це задача ідентифікації стає надзвичайно складною, оскільки для отримання якісних висновків щодо ідентифікації текстів необхідно враховувати всі найголовніші індивідуальні особливості окремих авторів. З урахуванням цього для розв’язання задачі встановлення авторства та ідентифікації текстів є доцільним використання модифікованих алгоритмів аналізу текстів.

У даній роботі розглядається використання методів машинного навчання, а саме методів нейронних мереж, які дають змогу ідентифікувати авторів текстів або визначити найбільш ймовірного автора з низки запропонованих. У якості узагальнюючої характеристики авторського стиля обирається певний числовий показник, який обчислюється на основі текстів (одного або декількох) автора. Цей показник стиля вважається індивідуальною особливістю автора. Порівняння показників для творів різних авторів або показника одного твору з еталонним показником автора, дає змогу визначити автора окремого тексту або

порівняти два тексти на предмет спільності їх автора.

Для тренування нейронної мережі, призначеної для розв'язання задачі класифікації, обрано масив художніх творів англомовної літератури.

1.4.2 Формальна постановка задачі

Процес ідентифікації авторів документів формально можна описати так. Під класифікацією текстового документа розумітимемо задачу автоматичного віднесення документу до однієї з заданих категорій. Кожна категорія представлена множиною текстів одного автора.

Розглядається множина текстів $T = \{t_1, ..., t_k\}$ і множина авторів $A = \{a_1, ..., a_n\}$. Для деякої підмножини текстів $T' \subseteq T$ автори відомі, тобто існує множина пар «текст-автор» $D = \{(t_i, a_i)\}_{i=1}^{\ell}$. Треба визначити, хто з множини A є справжнім автором текстів з множини $T'' = \{t_{|T'|+1}, ..., t_k\} \subseteq T$.

У даній постановці задачу ідентифікації автора можна розглядати як задачу класифікації з декількома класами. Таким чином множина A складається з певних класів та їх міток, D – тренувальний набір даних для навчання, T'' – об'єкти, що необхідно класифікувати.

Метою є побудова класифікатора, тобто знаходження деякої цільової функції $F : T \times A \to [0,1]$, яка співвідносить текст з множини T до його справжнього автора. Значення функції видається у вигляді ступені належності об'єкта класу.

У даній кваліфікаційній роботі для розв'язання поставленої задачі було обрано штучні нейронні мережі. Експерименти вітчизняних та зарубіжних дослідників показують, що на сьогоднішній день вони, при належному виборі вхідних параметрів та налаштуванні є найкращими у своєму класі [2]. У моделі використовується векторне представлення тексту, коли кожен текст представлений точкою у N-вимірному просторі. Елементами вектора мають бути харак-

теристики рівнів символів, слів, речень, структурні ознаки тексту.

Для визначення відмінностей стилів авторів пропонується наступна послідовність етапів.

Етап 1. Розбиття множини текстів на дві групи – тренувального та тестового наборів: перший використовується для навчання моделі класифікатора, другий – для перевірки точності ідентифікації автора за допомогою навченої моделі.

Етап 2. Формування моделі тексту шляхом вибору моделі представлення текстової інформації та виділення певних інформативних груп параметрів тексту. Відмінності в стилях авторів характеризується, головним чином, вживанням та частотою зустрічі певних ознак у тексті – вектором $(x_1,...,x_n)$;

Етап 3. Приведення значень ознак у єдиний діапазон за допомогою операцій нормування та шкалювання.

Етап 4. Коригування гіперпараметрів класифікатора з метою забезпечення високої роздільної здатності досліджуваних авторів шляхом навчання класифікатора на нормованих векторах ознак групи навчальних текстів та перевірки точності навченого класифікатора на векторах ознак тестової групи текстів. Початкове навчання класифікатора відбувається з параметрами за замовчуванням або за заданих значень параметрів.

Етап 5. Змінення переліку характеристик та ознак, які характеризують документи, якщо перебором гіперпараметрів класифікатора досягти необхідної точності не вдалось.

Підсумком є навчений класифікатор, ваги зв’язків якого налаштовані так, щоб він був здатний розділити стилі авторів, на текстах яких проводилось навчання, при поданні на його входи підібраного набору ознак.

1.5 Постановка задач дослідження

Після аналізу поставленої задачі та вибору методу для її розв’язання сфо-

рмуємо задачі дослідження даної кваліфікаційної роботи:

– сформулювати задачу ідентифікації авторів текстів за характерними ознаками авторського стилю;

– розв’язати задачу ідентифікації авторів текстів обраним методом машинного навчання;

– розробити програмний продукт, у якому буде реалізований метод ідентифікації та отримання висновків щодо належності тексту певному авторові;

– провести обчислювальні експерименти і проаналізувати отримані результати.

2 ВИБІР ТА ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ'ЯЗАННЯ

2.1 Огляд класичних методів машинного навчання для розв'язання задачі ідентифікації текстів

Важливим етапом під час вирішення задачі класифікації текстів є вибір методу машинного навчання, який застосовуватиметься до векторного подання текстів. Методи класифікації об'єктів, засновані на навчанні, вперше введені на розгляд у 1960-ті роки. В даний час розроблено безліч методів машинного навчання, які застосовуються при розв'язанні широкого кола завдань. Більшість цих методів застосовуються для вирішення задач класифікації, отже, можуть бути використані і для класифікації текстових даних. Розглянемо основні методи машинного навчання, які найчастіше використовуються для задач класифікації текстів.

2.1.1 Баєсівський класифікатор

Метод Баєса заснований на аналізі спільних розподілів ознак тексту та категорій [2]. Документу $D = \{d_1, d_2, ..., d_n\}$ ставиться у відповідність найбільш ймовірна категорія за формулою

$$c^* = \arg\max P_{c \in C}(c \mid x_1 = d_1, ..., x_n = d_n). \quad (2.1)$$

У задачі ідентифікації текстів метод Баєса застосовується окремо для кожної категорії і окремо приймається рішення, чи належить документ цій категорії.

Апостеріорна ймовірність належності документа до категорії обчислюється за формулою Баєса, що пов'язує апріорну ймовірність з апостеріорною:

$$P(c \mid x_1 = d_1,..., x_n = d_n) = \frac{P(x_1 = d_1,..., x_n = d_n \mid c) \cdot P(c)}{P(x_1 = d_1,..., x_n = d_n)}. \quad (2.2)$$

Оскільки знаменник є величиною постійною, то надалі його можна виключити з розгляду. За таких умов отримаємо:

$$c^* = \arg\max P_{c \in C}(x_1 = d_1,..., x_n = d_n \mid c) \cdot P(c). \quad (2.3)$$

Для спрощення обчислень висувають припущення про незалежність змінних $x_1, x_2,..., x_n$. У такому випадку формула для визначення найбільш ймовірної категорії матиме вигляд:

$$c^* = \arg\max P(c)_{c \in C} \cdot \prod_{i=1...n} P(x_i = d_i \mid c). \quad (2.4)$$

Звісно, припущення про незалежність змінних $x_1, x_2,..., x_n$ є дуже вагомою додатковою умовою, тому метод Баєса іноді називають «наївним» – naive bayes classifier. Не зважаючи на це, метод Баєса дає високі показники у задачі класифікації текстів, зокрема, у класичній задачі визначення спама.

2.1.2 Метод k найближчих сусідів

Метод k найближчих сусідів відносить класифікований об’єкт до того класу, до якого належить більшість з k найближчих до цього об’єкту зразків навчального набору даних [7].

У порівнянні з іншими методами метод k найближчих сусідів не потребує фази навчання. Для пошуку класу документу d, цей документ порівнюється зі всіма документами з навчальної вибірки. Для кожного документа e з навчальної вибірки визначається відстань – косинус кута між векторами ознак [2]:

$$\rho(d,e) = \cos(d,e). \quad (2.5)$$

Далі з навчальної вибірки обирають k документів, найближчих до d (k є гіперпараметром, значення якого визначається окремо у кожному конкретному випадку). Для кожного класу обчислюється релевантність за формулою

$$s(c_j,d) = \sum_{e\in\{k\}\wedge c_j\in Rub(e)} \cos(d,e). \quad (2.6)$$

Класи з релевантністю вище деякого заданого порогу зіставляють документу. Параметр k зазвичай обирається з інтервалу від 1 до 100.

Даний метод показує високу ефективність, але вимагає досить великих обчислювальних витрат.

2.1.3 Класифікатор Роше

Класифікатор Роше – один з найпростіших методів класифікації. Для кожної категорії обчислюється зважений центроїд за формулою [5]:

$$\vec{g}_c = \frac{1}{|R_c|}\sum_{d\in R_c}\vec{d} - \gamma\frac{1}{|R_{c,k}|}\sum_{d\in R_{c,k}}\vec{d}, \quad (2.7)$$

де R_c – множина документів, які належать категорії;

$R_{c,k}$ – k документів, які не належать категорії;

γ – параметр, що вказує на відносну важливість негативних прикладів.

Після обчислення зважених центроїдів для кожної категорії класифікатор Роше визначає належність документа класу за допомогою обчислення відстані між векторами документу та центроїдом кожного класу. Отримана відстань порівнюється із заданим порогом. У якості функції відстані часто використовують

косинус між векторами.

Даний метод має корисну особливість: зважені центроїди можна швидко переобчислити при додаванні нових прикладів. Ця особливість корисна, наприклад, у задачі адаптивної фільтрації, коли користувач поступово вказує системі, які документи вибрано правильно, а які – ні. У відповідь система може уточнити результати з огляду на нові документи.

Існує безліч різних модифікацій цього методу. Завдяки дуже простій ідеї даний метод часто використовується як базовий метод для порівняння з іншими.

2.2 Застосування нейронних мереж для задачі ідентифікації тексту

У даний час розроблено велику кількість різних видів класифікаторів, для побудови яких використовуються як статистичні методи, так і методи машинного навчання. Необхідність використання в аналізі даних великої кількості різноманітних методів класифікації, обумовлена тим, що розв’язувані за її допомогою задачі можуть мати свої особливості, пов’язані, наприклад, з числом класів або з поданням вихідних даних – їх обсягом, розмірністю та якістю, що потребує вибору правильного класифікатора. Тому вибір класифікатора, що відповідає особливостям розв’язуваної задачі, є важливим фактором та запорукою отримання правильного рішення.

Різні види класифікаторів мають свої переваги та недоліки. Так, класифікатори, в яких використовуються статистичні методи, мають високу математичну обґрунтованість, але при цьому складні або незручні у використанні та вимагають знання ймовірнісного розподілу вихідних даних і оцінки його параметрів, а також мають фіксовану структуру моделі. Крім цього, статистичні методи оцінюють лише ймовірність належності об’єкта до класу, але не «пояснюють» чому.

Класифікатори, засновані на машинному навчанні, не вимагають оцінки

параметрів розподілу вихідних даних, а міра подібності в них формалізується за допомогою функції відстані. Такі класифікатори називаються метричними. Як правило, вони простіші у реалізації та використанні, ніж параметричні, а їх результати зручніші для інтерпретації та розуміння. Але при цьому метричні класифікатори є евристичними моделями – забезпечують рішення лише в обмеженій кількості практично значущих випадків, можуть дати неточне чи не єдине рішення. Тому використовувати їх результати потрібно обережно.

Певним компромісом між параметричними та метричними методами є використання для розв'язання задач класифікації нейронних мереж (НМ). Насправді, НМ є непараметричними моделями, що не вимагають припущень про імовірнісний розподіл даних, але при цьому і не використовують міри відстаней. Це робить їх універсальними класифікаторами, дозволяючи отримувати результати навіть у випадках, коли параметричні та метричні класифікатори не дають прийнятного рішення [6].

Слід зазначити, що задача класифікації для НМ не є основною (як, наприклад, для дерев рішень або алгоритму k найближчих сусідів). Основною задачею для НМ є прогнозування. Проте, використовуючи спеціальні способи представлення даних, можна адаптувати НМ до роботи з категоріальними даними, тобто отримувати на вхід та формувати на виході категоріальні значення.

Можна виділити цілу низку переваг нейронних мереж з точки зору для використання їх як класифікаторів:

– робота нейронних мереж після завершення процесу навчання майже не вимагає втручання користувача, що робить їх достатньо зручними для використання;

– нейронні мережі дозволяють апроксимувати будь-яку неперервну функцію з прийнятною точністю, тому коло їх застосувань дуже широке;

– завдяки нелінійності математичних моделей, що реалізуються у нейронних мережах, можливо ефективно розв'язувати задачі класифікації лінійно нероздільних класів.

Зауважимо, що не існує спеціальних нейромережевих архітектур для

розв’язання задачі класифікації, зокрема, текстових даних. Найчастіше у цьому випадку використовуються НМ, які є мережами прямого поширення. У таких мережах на вхідні нейрони яких подаються значення ознак об’єкта, який необхідно класифікувати. Результатом роботи мережі на виході буде згенеровано мітку класу, до якого належить об’єкт, або ймовірність такої належності.

Для організації таких класифікаторів зазвичай використовують багатошарові персептрони [8]. У таких мережах елементи вектора ознак надходять на вхідні нейрони та розподіляються на всі нейрони першого прихованого шару мережі, і в результаті розмірність змінюється. Наступні шари, таким чином, поділяють об’єкти на класи у просторі ознак вищої розмірності. Наприклад, якщо розмірність вектора ознак вихідних даних дорівнює 4, і прихований шар містить 6 нейронів, то вихідний шар розбиває об’єкти на класи у 6-мірному просторі. Це дозволяє зробити процес більш ефективним. Правильно підібравши конфігурацію і параметри НМ, можна отримати добрі результати класифікації навіть у тих випадках, коли класифікатори інших типів, що працюють тільки в розмірності навчальних даних, не забезпечують прийнятних результатів. Недоліком є те, що конфігурація мережі, що найкраще апроксимує функцію розділення класів у просторі ознак, наперед невідома. Тому доводиться підбирати її експериментально, або використовувати досвід аналогічних реалізацій. Якщо розподіл класів такий, що для їх визначення потрібна складна функція, розмірність НМ може бути неприйнятно великою. У цьому випадку проблему можна вирішити шляхом попередньої обробки вихідних даних.

Тому дуже важливим кроком під час розробки класифікаційної моделі на основі НМ є попередня обробка та очищення даних. Об’єкти предметної області можуть описуватися великою кількістю ознак. Не всі вони дозволяють надійно розрізняти об’єкти різних класів. Не бажано також використовувати ознаки, значення яких є випадковими і не відображають закономірностей розподілу об’єктів за класами. Крім цього, важливу роль відіграє вибір кількості використовуваних ознак. З одного боку, чим більше ознак застосовується при побудові класифікатора, тим більше інформації використовується для розподілу класів.

Але при цьому зростають обчислювальні витрати і вимоги до розміру НМ. З іншого боку, зниження кількості ознак погіршує роздільність класів.

Ще одним важливим етапом попередньої обробки навчальних даних є нормалізація значень ознак до діапазону $(0,1)$. Нормалізація необхідна, оскільки ознаки, за якими здійснюється класифікація, мають різну фізичну природу ті їх значення можуть різнитися на декілька порядків. Крім цього, перед побудовою класифікатора на основі НМ слід провести профайлінг даних з метою оцінки їх якості, та за необхідністю застосувати до них засоби очищення даних таких як: заповнення пропусків, зменшення аномальних значень та викидів, виключення дублікатів та протиріч.

2.3 Алгоритм розв’язання задачі ідентифікації автора тексту

Для того, щоб побудувати ефективно працюючий класифікатор, що підходить для розв’язання задачі ідентифікації, необхідно мати якісні вихідні дані. Жоден класифікатор, заснований, зокрема, на нейронних мережах, не зможе забезпечити потрібну якість моделі, якщо набір прикладів не буде достатньо повним і репрезентативним для задачі. Отже, під час розробки нейронної мережі, яка вирішуватиме поставлену задачу, значної уваги слід приділити не лише підбору параметрів та тренуванню мережі, а й попередній обробки начального масиву даних.

За таких умов побудова класифікатора на основі нейронної мережі полягає у реалізації наступних етапів.

Етап 1. Підготовка даних:

1) скласти базу даних із прикладів, характерних для даної задачі;

2) розбити всю сукупність даних на навчальну та тестову множини.

Етап 2. Передобробка даних:

1) провести вибір ознак, найбільш значущих виходячи з постановки конкретної задачі класифікації;

2) здійснити трансформацію та, за необхідності, очищення даних (нормалізацію, виключення дублікатів та протиріч, зменшення викидів тощо); в результаті бажано отримати простір множини прикладів, що лінійно розділяються за класами;

3) вибрати систему кодування вихідних значень.

Етап 3. Конструювання, навчання та оцінка якості мережі:

1) вибрати топологію мережі (кількість шарів, число нейронів у шарах тощо);

2) вибрати активаційну функцію нейронів;

3) вибрати алгоритм навчання мережі;

4) оцінити якість роботи мережі на основі валідаційної множини або іншого критерію, оптимізувати архітектуру;

5) зупинитися на варіанті мережі, який забезпечує найкращу здатність до узагальнення та оцінити якість роботи на тестовій множині.

Етап 4. Використання та діагностика:

1) з'ясувати рівень впливу різних факторів на результат роботи мережі;

2) переконатись, що натренована нейронна мережа забезпечує необхідну точність класифікації;

3) при необхідності повернутися на етап 2, змінивши спосіб подання прикладів або змінивши базу даних;

4) використовувати мережу для розв'язання поставленої задачі на нових даних.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Високорівнева мова програмування Python

Python – це найпопулярніша високорівнева мова програмування з динамічною семантикою. Вона досить проста для роботи та читання, її використовують для зниження вартості розробки та обслуговування програм. По суті, машинне навчання – це технологія, яка допомагає програмам на основі штучного інтелекту навчатися та видавати результати автоматично, без людського втручання. Мова Python найкраще підходить для виконання таких задач, тому що вона є досить зрозумілою у порівнянні з іншими мовами. Більше того, вона має відмінну продуктивність при обробці даних.

Простота та широке коло застосувань. Одна з основних причин, чому Python використовується для машинного навчання, полягає в тому, що вона має безліч фреймворків, які спрощують процес написання коду і скорочують час на розробку. Багато бібліотек та фреймворків Python використовуються в машинному навчанні. У наукових розрахунках використовується Numpy, у прикладних обчисленнях – SciPy, у аналізі даних – SciKit-Learn. Ці бібліотеки працюють у таких фреймворках, як TensorFlow, CNTK та Apache Spark. Також існує фреймворк, розроблений спеціально для машинного навчання – це PyTorch.

Зрозумілість. Python добре підходить для машинного навчання, тому що самі алгоритми машинного навчання складні для розуміння. При роботі з Python розробнику не потрібно приділяти багато уваги безпосередньо написанню коду, усю увагу він може зосередити на вирішенні складніших завдань, пов’язаних з машинним навчанням. Простий синтаксис мови Python допомагає розробнику тестувати складні алгоритми з мінімальною втратою часу на їх реалізацію.

Величезна підтримка. Ще одна перевага Python – це велика підтримка та якісна документація. Існує безліч корисних ресурсів, на яких програміст може отримати допомогу та консультацію, перебуваючи на будь-якому етапі розроб-

ки та тестування програми.

Гнучкість. Ще одна перевага Python у машинному навчанні полягає у його гнучкості, наприклад, розробник має вибір між об’єктно-орієнтованим підходом і скриптами. Python допомагає поєднувати різні типи даних. Більш того, Python особливо зручний для тих розробників, які більшу частину коду пишуть за допомогою інтегрованого середовища розробки.

Наведені вище фактори пояснюють, чому Python так активно використовують у сфері машинного навчання.

3.2 Опис програми

Для реалізації задачі ідентифікації текстів методами машинного навчання у ході виконання кваліфікаційної роботи було написано програмний продукт мовою програмування Python. Ця мова підходить для вирішення трудомістких задач, і, зокрема, задачі кваліфікаційної роботи, завдяки своїй гнучкості, простоті у написанні коду і великої кількості бібліотек для аналізу та роботи з великими обсягами даних [4].

Після запуску програмного додатку перед користувачем відображається вікно з можливістю запустити програму. В програму імпортуються данні для навчання нейронної мережі у вигляді таблиці (автор, книга). Далі дані проходять попередню обробку, видаляються зайві знаки, всі слова приводяться до одного формату, та проходять токенізацію (присвоєння кожному слову порядкового номера). Ці токени складають масив з векторів за кожним твором авторів, що аналізуються. Навчання класифікатора проходить у декілька етапів. Для перевірки якості навчання у програму подається тестовий набір даних, який проходить таку саму попередню обробку. Шляхом розбиття текстів на класи за авторами класифікатор зіставляє тестовому тексту передбачуваного автора, з множини тих, що представлені у навчальній вибірці.

Структура мережі:

– двоспрямована модель на основі трансформеру BERT від Google;

– вхідний шар – BertEmbeddings;

– розмір батча – 4;

– довжина послідовності – 512;

– виключення (dropout = 0,2);

– слоїв уваги BertEncoder – 12;

– вузлів – 768;

– кількість виходів – 12;

– кількість параметрів – 110M;

– кількість класів – 3;

– алгоритм оптимізації – Adam;

– кількість епох – 7.

Лістинг програми наведено у додатку А.

4 РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНОГО ЕКСПЕРИМЕНТУ

В кваліфікаційній роботі було проведено серію експериментів, пов’язаних навчанням класифікатора для розв’язання задачі ідентифікації текстів, авторство яких не визначено. У якості вихідних даних було використано навчальний набір, що містить твори зарубіжної художньої літератури. Для аналізу було обрано вибірку творів наступних письменників: Мері Шеллі (MWS), Говард Лавкрафт (HPL), Аллан Едгар По (EAP). Даний вибір пояснюється великою кількістю написаних цими авторами творів, що необхідно для забезпечення формування тренувального набору достатнього обсягу. Вихідні дані аналізованих текстів наведені на рисунку 4.1. Дані представлені у вигляді таблиці з порядковим номером рядка, його ідентифікаційним номером, самим текстом, та автором.

	id	text	author
0	id26305	This process, however, afforded me no means of...	EAP
1	id17569	It never once occurred to me that the fumbling...	HPL
2	id11008	In his left hand was a gold snuff box, from wh...	EAP
3	id27763	How lovely is spring As we looked from Windsor...	MWS
4	id12958	Finding nothing else, not even gold, the Super...	HPL
...	...	...	...
19574	id17718	I could have fancied, while I looked at it, th...	EAP
19575	id08973	The lids clenched themselves together as if in...	EAP
19576	id05267	Mais il faut agir that is to say, a Frenchman ...	EAP
19577	id17513	For an item of news like this, it strikes us i...	EAP
19578	id00393	He laid a gnarled claw on my shoulder, and it ...	HPL

19579 rows × 3 columns

Рисунок 4.1 – Вихідні дані аналізованих текстів

Перед початком роботи з даними їх було попередньо оброблено, зокрема, видалено пунктуаційні знаки та стоп-слова, проведено лематизацію. До регіст-

ру побудований класифікатор не чутливий, тому було вирішено не навантажувати його зайвими обробками. Після попередньої обробки текст виглядає наступним чином.

	id	text	author
0	id26305	This process however afford me no mean of asc...	EAP
1	id17569	never once occur to me that the fumble might ...	HPL
2	id11008	his leave hand be a gold snuff box from which...	EAP
3	id27763	How lovely be spring As we look from Windsor ...	MWS
4	id12958	Finding nothing else not even gold the Superi...	HPL
...	...	...	...
19574	id17718	could have fancy while look at it that some e...	EAP
19575	id08973	The lids clench themselves together as if in ...	EAP
19576	id05267	Mais il faut agir that be to say a Frenchman ...	EAP
19577	id17513	For an item of news like this it strike us it...	EAP
19578	id00393	He lay a gnarl claw on my shoulder and it see...	HPL

19579 rows × 3 columns

Рисунок 4.2 – Вихідні дані аналізованих текстів

На рисунку 4.3 наведено графік розподілу цільових міток, який дозволяє оцінити рівномірність наповнення тренувального набору об’єктами різних класів.

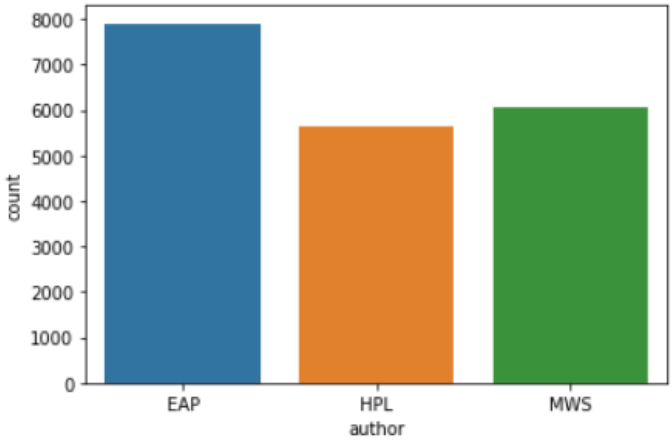

Рисунок 4.3 – Графік розподілу цільових міток

Виконаємо розподіл перевірочного набору даних та створимо спеціальний клас набору даних для тренування і тестування без цільових міток. Оскільки тексти у наборі даних мають різну довжину, ми будемо використовувати заповнення, щоб зробити всі повідомлення однакової довжини.

У якості алгоритму оптимізації нашої моделі будемо використовувати Adam. Цей оптимізатор для оновлення ваг використовує ковзні середні градієнтів і других моментів градієнтів.

Отже, ми визначили архітектуру моделі, вказали оптимізатор і функцію втрат та підготували дані. Після встановлення функції для оцінки експериментальним шляхом було визначено кількість епох. На рисунку 4.4 та 4.5 наведено результати тренування побудованої нейронної мережі. Зокрема, рис. 4.4 відображує результати функції втрат на тренувальному і тестовому наборі за кожною епохою, а на рис. 4.5 показана точність побудованої нейронної мережі.

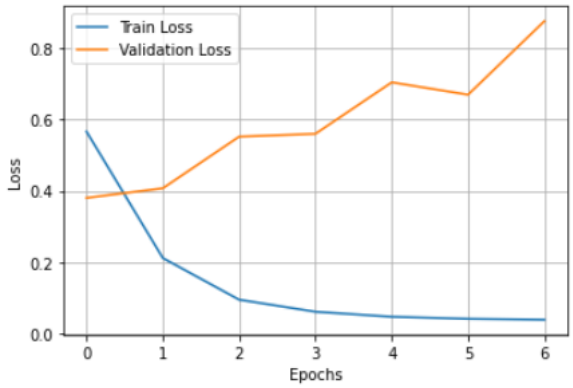

Рисунок 4.4 – Порівняння втрат навчального та перевірочного наборів

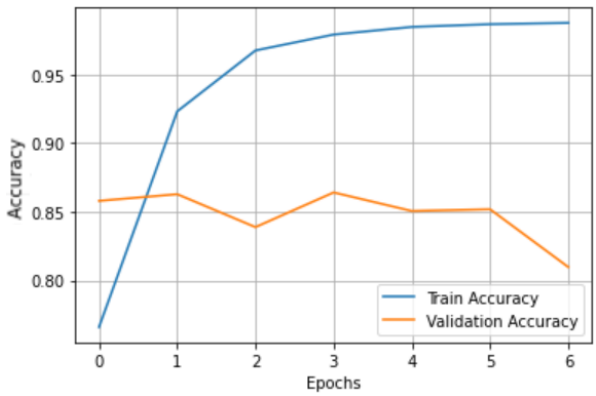

Рисунок 4.5 – Порівняння точності навчального та перевірочного наборів

У таблиці 4.1 наведено значення витрат для навчання та перевірки, а також точність на кожній епосі. Точність обчислюється шляхом зіставлення результатів передбачення дійсним авторам текстів, інакше кажучи, ми порівнюємо, скільки авторів модель ідентифікувала правильно з усього перевірочного набору.

З таблиці видно, що витрати на навчання зменшуються із зростанням номеру епохи, а точність моделі зростає. Отриману після 7 епох точність можна вважати прийнятною, отже, результати свідчать, що модель навчилась добре розв’язувати задачу ідентифікації текстів обраних авторів, а налаштування класифікатора можна вважати оптимальними. Отже, даний класифікатор можна використовувати для передбачення автору тексту.

Таблиця 4.1 – Результати тренування на кожній епосі

Набір / Епоха	Втрати		Точність	
	Тренувальний	Перевірочний	Тренувальний	Перевірочний
1	0,5661	0,3799	0,7660	0,8580
2	0,2111	0,4069	0,9233	0,8629
3	0,0943	0,5518	0,9676	0,8389
4	0,0604	0,5597	0,9793	0,8641
5	0,0464	0,7042	0,9848	0,8506
6	0,0407	0,6695	0,9868	0,8519
7	0,0381	0,8758	0,9877	0,8098

Для перевірки якості роботи моделі у програму було подано 8000 «невідомих» фрагментів текстів розглянутих авторів, частина результатів ідентифікації наведена у таблиці 4.2. Зокрема, у таблиці наведені ймовірності, з якими мережа відносить «невідомий» їй текст кожному з можливих авторів. Класифікатор визначає автора тексту за більшим зі значень ймовірності.

Таблиця 4.2 – Результати ідентифікації текстів невідомих авторів

Текст	EAP	HPL	MWS
1	0,9984	0,0013	0,0003
2	0,0006	0,9989	0,0005
3	0,0001	0,0001	0,9998
4	0,9989	0,0003	0,0008
5	0,0001	0,9998	0,0001
6	0,0272	0,0001	0,9727
7	0,9981	0,0002	0,0017
8	0,0004	0,9995	0,0001
9	0,0001	0,0001	0,9998
10	0,9979	0,0018	0,0003
11	0,0004	0,9995	0,0001
12	0,0037	0,0003	0,9960
13	0,9992	0,0005	0,0003
14	0,0891	0,9082	0,0027
15	0,0001	0,0001	0,9998
16	0,9992	0,0003	0,0005
17	0,0001	0,9998	0,0001
18	0,0914	0,3138	0,5948
19	0,9950	0,0007	0,0043
20	0,0001	0,9997	0,0002

Як видно з таблиці, даний метод є перспективним для подальших досліджень, оскільки дає доволі високу точність ідентифікації.

ВИСНОВКИ

У кваліфікаційній роботі було досліджено задачу ідентифікації авторів текстів. На основі проведеного аналізу сучасних методів машинного навчання для вирішення задачі ідентифікації автора текстів було обрано метод розв’язання поставленої задачі.

У ході виконання роботи було створено програмну реалізацію метода класифікації для ідентифікації автора тексту, наданого в відцифрованому вигляді, за допомогою штучних нейронних мереж. Програма була написана за допомогою мови Python з використанням нових бібліотек, створених для аналізу текстів. Програма містить наступні модулі: завантаження тексту, обробки тексту, токенізації, навчання моделі, класифікації за ознаками, модуль збору результатів тренування, модуль оцінки класифікації, модуль складання таблиці передбачень для текстів невідомих авторів та інші допоміжні модулі. Система має гнучку архітектуру, що дає можливість швидко змінювати параметри класифікації або особливо налаштувати класифікатор для нестандартних наборів даних. У якості текстів були взяти твори зарубіжних авторів. Було проведено багато експериментів для аналізу роботи класифікатора, у результаті яких було встановлено залежності точності навчання від кількості вхідних даних та кількості епох навчання.

Аналіз результатів показав, що при належному налаштуванні класифікатора реалізований метод дає прийнятні результати за умови наявності достатньо великих масивів вхідних даних, що вигідно відрізняє його від інших методів. Обмеженням методу є низька швидкість навчання класифікатора, це прямо пов’язано з кількістю даних, що аналізуються. Для збільшення показників точності можна запропонувати використовувати комбінацію з декількох моделей та усереднення їх результатів класифікації.

Розроблену програму можна використовувати для вирішення задач ідентифікації текстів таких як: встановлення авторів неідентифікованих текстів, перевірка на плагіат, сортування та інші задачі аналізу великих обсягів тексту.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Автоматическая обработка текстов на естественном языке и компьютерная лингвистика // Е. И. Большакова, Э. С. Клышинский, Д. В. Ландэ [и др.]. Москва : МИЭМ, 2011. 272 с.

2. Агеев М. С. Методы автоматической рубрикации текстов, основанные на машинном обучении знаниях экспертов : автореф. дис. на соискание уч. степени канд. физ.-мат. наук : [спец.] 05.13.11 “Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей” / Московский государственный ун-т им. М. В. Ломоносова. Москва, 2004. 136 с.

3. Батура Т. В. Формальные методы определения авторства текстов // Информационные технологии. 2012. Т. 10, № 4. С. 81–94.

4. Бенгфорт Б, Билбро Р, Охеда Т. Прикладной анализ текстовых данных на Python. Машинное обучение и создание приложений обработки естественного языка. Санкт-Петербург : Питер, 2019. 368 с.

5. Борисов Л. А. Орлов Ю. Н. Осминин К. П. Идентификация автора текста по распределению частот буквосочетаний // Прикладная информатика. 2013. Т. 26. № 2. С. 95-108.

6. Гольдберг Й. Нейросетевые методы в обработке естественного языка. Москва : ДМК Пресс, 2019. 282 с.

7. Жерон О. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем. Санкт-Петербург : ООО «Альфа-книга», 2018. 688 с.

8. Макмахан Б., Рао Д. Знакомство с PyTorch: глубокое обучение при обработке естественного языка. Санкт-Петербург : Питер, 2020. 256 с.

9. Подшиваленко Б. О. Застосування методів статистичного аналізу для розв’язання задачі ідентифікації текстів // 25-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті» : зб. матеріалів форуму (м. Харків, 20-22 квітня 2021 р.). Т. 7. Харків : ХНУРЕ, 2021. С. 65-66.

10. Романов А. С. Методика и программный комплекс для идентифика-

ции автора неизвестного текста автореф. дис. на соискание уч. степени канд. техн. наук : [спец.] 05.13.18 “Математическое моделирование, численные методы и комплексы программ” / Томский государственный ун-т систем управления и радиоэлектроники. Томск, 2010. 26 с.