Вінницький національний технічний університет
Факультет комп'ютерних систем і автоматики
Кафедра автоматики та інформаційно-вимірювальної техніки

Пояснювальна записка

до магістерської кваліфікаційної роботи

на тему: «Застосування методів машинного навчання для визначення авторства україномовного тексту»

Виконав: студент групи 1АКІТ-18м
спеціальності
151 – Автоматизація та
комп'ютерно-інтегровані технології

Стовбчатий М. М.

Керівник д.т.н., проф., зав.каф. АІІТ
Квєтний Р.Н.

Рецензент:_____

Вінниця – 2019

Анотація

В даній роботі розроблено новий узагальнений метод визначення авторства тексту, який базується на комбінування методів лінгвістики та машинного навчання, що дозволяє значно підвищити точність атрибуції авторства. Інформативність обраних чисельних параметрів синтаксичної структури речення україномовного тексту доведено та ефективність запропонованого методу підтверджено шляхом порівняльного експерименту для різних україномовних текстів 3-х авторів різними методами машинного навчання.

Annotation

In this work, developed a new generalized method for determining the authorship of text, which is based on a combination of methods of linguistics and machine learning, which greatly improves the accuracy of attribution of authorship. The informativeness of the chosen numerical parameters of the syntactic structure of the sentence of the Ukrainian-language text is proved and the effectiveness of the proposed method is confirmed by a comparative experiment for different Ukrainian-language texts of 3 authors by different methods of machine learning.

ЗМІСТ

ВСТУП

Актуальність даної роботи полягає у широкій предметній області застосування результатів лінгвістичної експертизи текстової інформації. Автоматизація процесів лінгвістичної експертизи, зокрема визначення авторства тексту, дозволить підвищити якість формування профілів учасників соціальних мереж і розбиття їх на категорії, виявлення плагіату, оперативного визначення недоброчесних або зловмисних дій користувачів інформаційних систем тощо. На відміну від англійської, російської, цілого ряду романських мов, рівень розвитку відповідних лінгвістичних моделей і технологічних засобів для української мови є недостатнім.

Мета роботи полягає в підвищенні якості визначення авторства україномовного тексту на основі методів і моделей комп'ютерної лінгвістики та машинного навчання, а також доступних програмних бібліотек і технологічних засобів.

Для досягнення поставленої мети необхідно розв'язати такі задачі дослідження:

а) Аналіз застосування методів машинного навчання в задачах комп'ютерної лінгвістики;

б) Огляд та порівняльний аналіз існуючих систем лінгвістичної експертизи;

в) Побудова лінгвістичної моделі та методу визначення авторства на основі аналізу зв'язків між лексичними одиницями україномовного тексту;

г) Дослідження можливостей мови програмування Python у сфері аналізу статистичних даних та машинного навчання;

д) Побудова програмного забезпечення та оцінка якості встановлення авторства тексту.

Об'єкт дослідження - процеси статистичного, синтаксичного та семантичного аналізу україномовних текстів.

Предмет дослідження – моделі, методи та засоби визначення авторства україномовних текстів.

Наукова новизна: Запропоновано новий метод визначення авторства україномовного тексту, який, на відміну від існуючих, базується на лінгвістичній моделі побудови графу зв’язків між лексичними одиницями речення тексту та застосуванні методів машинного навчання за новими формальними ознаками множини речень тексту, що дозволяє підвищити якість визначення авторства україномовного тексту.

Практична цінність роботи полягає у отриманні методики підготовки даних та навчання нейронної мережі за результатами удосконаленого статистичного аналізу тексту з метою визначення авторства україномовного тексту для обраної групи авторів.

Методи дослідження – метод синтаксичного аналізу тексту, методи машинного навчання, методи статистичного дослідження структури речень.

Апробація результатів та публікації. Результати роботи висвітлено в матеріалах конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2020)» та матеріалах доповідей XLVIII науково-технічної конференції підрозділів Вінницького національного технічного університету (2018) [1,2].

РОЗДІЛ 1 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ВИЗНАЧЕННЯ АВТОРСТВА ТЕКСТУ

1.1 Задача статистичної атрибуції тексту

Основна ідея статистичної чи обчислювальної атрибуції авторства полягає в тому, що вимірюючи деякі текстові особливості, ми можемо розрізнити тексти, написані різними авторів[3]. Перші спроби кількісної оцінки стилю письма відносяться до 19 століття, перше дослідження Менденхолла (1887) на п'єсах Шекспіра з подальшим статистичним дослідження в першій половині 20 століття. Пізніше детальне дослідження Мостеллера і Уолласа (1964)[4] про авторство "Документів федераліста" (серія із 146 політичних нарисів, написаних Джоном Джеєм, Олександром Гамільтоном та Джеймсом Медісон, на дванадцять з яких претендували і Гамільтон, і Медісон), безперечно, найбільше впливовий твір в авторській атрибуції[5]. Їх метод базувався на байєсівській статистиці та аналізі частот невеликого набору загальних слів (наприклад, "і", "до" тощо), що призвело до значних результатів дискримінації між авторами-кандидатами [6].

Насправді, роботами Мостеллера і Уоллеса (1964) започатковано нетрадиційні авторські дослідження атрибуції, на відміну від традиційних методів, заснованих на міркуваннях експерта людини. З тих пір потім і до кінця 1990-х років у дослідженні авторської атрибуції переважали спроби отримати функції для кількісного визначення стилю письма, цей напрямок досліджень відомий як «стилометрія» [7].

Отже, велику кількість параметрів, включаючи тривалість речення, довжину слова, частоту слів, частоту символів, склад та розмір словника було запропоновано застосувати як аргументи функції визначення авторства в стилометрії. Рудман (1998) підрахував, що було здійснено майже 1000 різних

способів у межах цього підходу. Запропоновані в цей період методології атрибуції авторів формальні обрахунки параметрів стилю нерідко базувались на комп'ютерних обчисленнях, проте практично не було спроб побудувати повністю автоматизовану систему.

У певних випадках застосування методів стилометрії були досягнуті вражаючі попередні результати. Це змусило багатьох лінгвістів вважати, що рішення проблеми визначення авторства незабаром буде закрито. Найбільш характерний приклад – метод CUSUM (або QSUM) який набув розголосу і був прийнятий в судах як експертний доказ.

Однак дослідницька спільнота жорстоко критикувала стилометрію і вважала цей підхід загалом недостовірним. Власне, головною проблемою того раннього періоду була відсутність об'єктивної оцінки запропонованих методів. У більшості випадків випробувальними текстами були літературні твори невідомого або спірного авторства (наприклад, справа федераліста), тому оцінка точності віднесення твору до автора навіть не була можливою. Основні методологічні обмеження цього періоду стосовно процедури оцінювання був наступним:

а) текстові дані були занадто довгими (зазвичай, навіть цілі книги) та, ймовірно, не однорідні стилістично;

б) кількість кандидатів-авторів була занадто мала (як правило, 2 або 3;.

в) накопичені дані з оцінки не контролювались за темою;

г) оцінка запропонованих методів була переважно інтуїтивно зрозумілою (як правило, на основі суб'єктивного візуального огляду експертів-лінгвістів);

д) порівняння різних методів було важким через відсутність відповідних баз даних.

З кінця 1990-х років у дослідженнях атрибуції авторства відбулися значні зміни. Поява величезної кількості електронних текстів, доступних через Інтернет-засоби масової інформації (електронні листи, блоги, онлайн-форуми тощо), призвела до зростання потреби в ефективній роботі з цією інформацією. Цей факт мав значний вплив в таких наукових сферах, як пошук інформації, машинне навчання та обробка природної мови (NLP). Розвиток цих напрямків значно вплинув на технології визначення авторства, зокрема [8]:

а) дослідження у сфері пошуку інформації розробили ефективні методи представлення та класифікація великих обсягів тексту;

б) потужні алгоритми машинного навчання стали доступними для обробки багатовимірних і розріджених даних, що дозволило отримати більш ефективні класифікатори. Окрім того, для класифікації різних груп даних були створені стандартні методики оцінювання ефективності навчання мережі за тими ж даними базових показників;

в)NLP-дослідження розробили інструменти, здатні більш повно аналізувати текст та надавати нові форми даним для представлення стилю (наприклад, функції, засновані на синтаксисі).

Не менш важливо, що велика кількість доступних електронних текстів виявила потенціал використання аналізу авторства в різних прикладних областях, включаючи розвідку (наприклад, приписування повідомлень чи текстів відомим терористам, пов'язуючи різні повідомлення за авторством) кримінальні розслідування (наприклад, виявлення зловмисників за листами, перевірка справжності записок про самогубство тощо). Це ж стосується цивільного законодавства, наприклад, спори щодо авторського права, комп'ютерної криміналістики, наприклад, визначення авторів вихідного коду шкідливого програмного забезпечення. Звичайно, залишились напрями традиційного

застосування атрибуції авторства до літературних досліджень, зокрема приписування анонімних чи спірних літературних творів відомим авторам.

Отже, останні 10-15 років можна розглядати як нову еру технологій аналізу авторства, в якій переважають зусилля з розробки практичних програм, що стосуються текстів у реальному світі. Значно більше досліджуються електронні листи, блоги, повідомлення на форумі, вихідний програмний код тощо, а не тексти – підстави для рішення спірних літературних питань. Важливо, що зараз увага акцентується на об'єктивній оцінці запропонованого методу шляхом порівняння різних методів на основі загально-доступних текстових корпусів. Крім того, факторами, що відіграють вирішальну роль у точності аналізу, розглядаються створені моделі, наприклад такі, як розмір навчального тексту кількість кандидатів-авторів та розповсюдження навчальних текстів для кандидатів-авторів [9].

У типовій задачі атрибуції авторства тексту невідомого авторства присвоюється один автор-кандидат з потенційного набору авторів-кандидатів, для яких доступні зразки текстів з беззаперечним авторством. З точки зору машинного навчання це може розглядатися як задача однозначної категоризації тексту на багатьох класах. Інакше цю задача також називають ідентифікацією авторства (або автора), як правило, дослідники з досвідом роботи в комп'ютерних науках. Декілька досліджень зосереджуються виключно на атрибуції авторства.

В той же час інші дослідники використовують дану задачу як просто ще один тестовий майданчик для відпрацювання методології категоризації тексту.

Поза основною проблемою можна визначити кілька інших завдань аналізу авторства, зокрема наступне:

а) підтвердження авторства, тобто, обґрунтувати, чи написав певний автор певний текст чи ні.

б) виявлення плагіату, тобто визначення ступеню подібності між двома текстами.

в) авторське профілювання або отримання його характеристик шляхом вилучення інформації про вік, освіту, стать тощо автора певного тексту.

г) виявлення стилістичних невідповідностей, можливих при спільному написанні певного тексту.

Отже, огляд досліджень в галузі атрибуції авторства тексту демонструє значні зміни у підходах протягом останніх двох десятиліть (попередній період повно переглядається Холмсом. Сучасні технології аналізу текстової інформації спираються на моделі машинного навчання та відповідні параметри обчислень, інструменти комп'ютерної лінгвістики та пошукові машини. Мовні чи літературні питання з відповідними оцінками лінгвістів-експертів вже не відіграють вирішальної ролі в задачі атрибуції авторства тексту.

1.2 Стиль-метричні особливості

Попередні дослідження щодо атрибуції авторства запропонували таксономії ознак для кількісної оцінки стилю написання, так звані маркери стилів, за різними ознаками та критеріями [12].

Особливості представлення стилістичних ознак в основному зосереджені на обчислювальних вимогах до їх вимірювання. По-перше, лексичні та символьні особливості розглядають текст як просту послідовність слів-лексем або символів відповідно. Зауважимо, що хоча і лексичні особливості складніші за синтаксичні характеристики, аналіз починається традиційно з них. Потім, синтаксичні та семантичні ознаки потребують більш глибокого мовного аналізу, одночасно специфічного для застосування функцій, які можна ідентифікувати лише в певних

текстових областях або мовах. Основні категорії відповідних функцій та необхідні інструменти і ресурси для їх вимірювання наведені в таблиці [13].

Крім того, різні методи вибору та вилучення функцій для формування найбільш відповідного набору функцій для конкретного корпусу є окремою задачею.

1.3 Профільні підходи

Один із способів поводження з доступними текстами, що точно відносяться до кожного автора – це об'єднати їх в один текстовий файл. Цей великий файл використовується для отримання властивостей авторського стилю. Тоді текст невідомого авторства порівнюється з кожним авторським файлом і оцінюється на основі міри відстані – найімовірнішим автором визначається такий, для якого міра відстані від його файлу до невідомого найменша.

Слід підкреслити, що окреме представлення кожного тексту не використовується як зразок, є лише одне представлення великого файлу для автора. Як результат – відмінності між навчальними текстами того ж автора не враховуються. Причому стиль-метричні параметри, витягнуті з об'єднаного файлу, можуть бути зовсім різними порівняно з кожним із оригінальних навчальних текстів. У типовій архітектурі профільного підходу (рисунок 1.1) через x позначається вектор функцій подання тексту, причому xA - профіль автора A, а xU – профіль тексту невідомого авторства [14].

Профільні підходи мають дуже простий навчальний процес. Власне, навчання як фаза якраз включає вилучення профілів для кандидатів-авторів. Далі, атрибуція моделі зазвичай базується на функції відстані, яка обчислює відмінності профілю тексту невідомого авторства та профілю кожного автора.

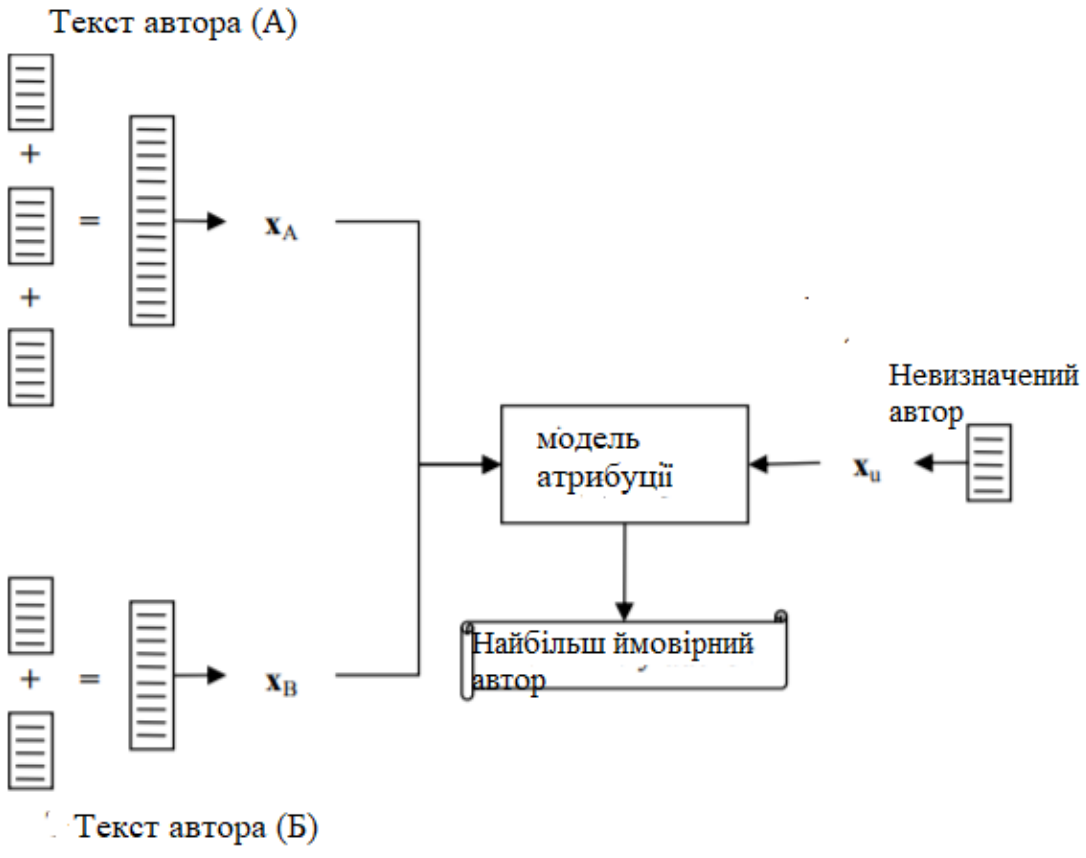

Рисунок 1.1 - Типова архітектура підходів, заснованих на екземплярах.

Нехай *PR(x)* є профілем тексту *x* і *d(PR (x), PR (y))* відстанню між профілем тексту *x* та профілем тексту *y*. Потім найімовірніше, автор невідомого тексту *x* задається:

$$author(x) = \arg\min d\big(PR(x), PR(x_a)\big), \tag{1.1}$$

де *a* - сукупність кандидатів-авторів, x_a – об'єднання всіх навчальних текстів для автора. Далі ми спочатку опишемо, як цей підхід можна реалізувати, використовуючи ймовірнісні та компресійні моделі, а потім метод CNG та його варіанти застосування.

1.4 Ймовірнісні моделі

Один з найбільш ранніх підходів до ідентифікації автора, який досі використовується в багатьох сучасних дослідження, полягає у використанні ймовірнісних моделей. Такі методи намагаються максимізувати ймовірність *P(x|a)* для тексту *x*, що належить кандидату в автори *a*. Потім модель атрибуції шукає автора, який максимізує наступний показник подібності:

$$author(x) = \arg\max log_2 \frac{P(x|a)}{(x|a)}, \tag{1.2}$$

де умовні ймовірності оцінюються шляхом конкатенації x_a всіх наявних навчальних текстів кандидату в автори a і об'єднання всієї решти текстів відповідно [15].

Варіанти таких імовірнісних класифікаторів (наприклад, наївні Bayes) уперше було досліджено у межах напряму тематичної категоризації тексту. Проте той же алгоритм Bayes, доповнений статистичними мовними моделями, досяг високих результатів в експериментах з атрибуцією авторства. В порівнянні з іншими популярними класифікаторами, наївний Bayes дозволяє досліджувати ланцюгові залежності спостережуваних змінних для збору контекстної інформації [16].

Більше того, у межах даного підходу можуть бути застосовані складні методи згладжування шляхом статистичного моделювання мови (найкращі результати для визначення авторства були отримані з використанням абсолютного згладжування). Описаний спосіб можна застосувати як до символьних ланцюгів, так і до послідовностей слів. Відомо, що Пенг у 2004 р. досяг найкращих результатів щодо визначення авторства, використовуючи рівень слів у

ймовірнісній моделі для конкретного корпусу. Однак цього не було підтверджено в інших дослідженнях.

1.5 CNG та варіанти

Профільний метод, що представляє особливий інтерес, відомий як підхід Common n-Grams (CNG), описаний Keselj у 2003 р. Цей метод використовує конкретне зображення профілю автора. Зокрема, профіль *PR(x)* тексту *x* складається з *L* найчастіших символів, що позначають *n*-грами цього тексту. Наступна відстань використовується для оцінки подібності між двома текстами *x* і *y* [17]:

$$d\big(PR(x), PR(y)\big) = \sum_{g \in P(x \cup P(y))} \left(\frac{2(f_x(g) - f_y(g))}{f_x(g) + f_y(g)}\right) * 2 \qquad (1.3)$$

де g - символьна *n*-грама, тоді як $fx\,(g)$ і $fy\,(g)$ - відносні частоти виникнення цієї *n*-грами в текстах *x* і *y*.

Отже, через відповідне слово цей метод обчислює несхожість між двома профілями, обчислюючи відносну різницю між їх загальними *n*-грамами. Усі *n*-грами двох профілів, які не є загальними, вносять постійну величину в значення відстані. Метод CNG має два важливі параметри, які слід налагодити: розмір профілю *L* і довжина символу *n*-грами *n*, тобто скільки довгих рядків складають профіль. Keselj та ін. (2003) повідомили про свої найкращі результати для $1000 \leq L \leq 5000$ та $3 \leq n \leq 5$. Цей основний підхід успішно застосовується в різних експериментах з ідентифікації авторства, включаючи конкурс авторського права на атрибуцію, організований у 2004 році. Важлива проблема в авторських

завданнях атрибуції виникає при розповсюдженні результатів, оскільки навчальний корпус над авторами-кандидатами неоднаковий [16].

Наприклад, особливо незвично у криміналістичних заявах мати лише декілька навчальних текстів для деяких кандидатів-авторів і дуже багато навчальних текстів для інших авторів [17]. Більше того, довжина цих зразків може не допускати їх сегментацію на кілька частин для збагачення навчальних прикладів певних авторів. В умовах машинного навчання це становить проблему дисбалансу в класах [18].

Більшість підходів з атрибуції авторства проводять експериментальні дослідження на основі збалансованих навчальних наборів (тобто рівної кількості зразків навчального тексту для кожного автора-кандидата), тому неможливо оцінити їх точність за умов класового дисбалансу. Лише кілька досліджень беруть до уваги цей фактор. Функція дистанції CNG добре працює [18], коли тренувальний корпус є відносно збалансованим, однак це не вдається, якщо розмір не менше одного профілю автора коротший за L [19].

Наприклад, якщо ми використовуємо $L = 4\ 000$ і $n = 3$, але доступні навчальні тексти певного автора-кандидата занадто короткі, то загальна кількість 3-грам, що може бути витягнута з текстів цих авторів ймовірно менша за 4000. Функція відстані сприяє такому автору-кандидату, оскільки відбудеться об'єднання профілю невідомого тексту та профілю цього автора [20]. У результаті виходить значно менша n-грама, тому відстань між невідомим текстом і цим автором буде можна оцінити як досить низьку порівняно з іншими авторами. Щоб подолати цю проблему, Францаско, Стамататос, Грицаліс і Кацікас (2006) запропонували іншу і простішу відстань, що називається спрощеним перетином профілю (SPI), яка просто підраховує суму загальних n-грам двох профілів, не враховуючи решти [21].

Застосування цього заходу до ідентифікація автора вихідного коду дала кращі результати, ніж оригінальна відстань CNG [22]. Зауважимо, що, на відміну від відстані CNG, SPI – це міра подібності, яка означає, що найбільше ймовірно, автором є автор-кандидат з найвищим значенням SPI. Проблема цієї відстані може виникнути, коли всі кандидати-автори, крім одного, мають дуже короткі тексти. Тоді показник SPI наддасть перевагу автору з довгими текстами, оскільки в цих текстах буде виявлено набагато більше спільних n-грам з невідомим текстом [23]. Ще один варіант обрахунку відстані CNG був запропонований Stamatatos (2007)

$$d\big(PR(x), PR(y)\big) = \sum_{g \in P(x \cup P(y))} \Big(\frac{2\big(f_x(g) - f_y(g)\big)}{f_x(g) + f_y(g)}\Big) * 2 * \Big(\frac{2(f_x(g) - f_N(g))}{f_x(g) + f_N(g)}\Big) * 2, \qquad (1.4)$$

де N – корпусна норма (об'єднання всіх доступних текстів усіх кандидатів у автори) та f_N (g) - відносна частота появи n-грами g у нормі корпусу. Потрібно відмітити, що ця функція не симетрична як оригінальна функція CNG [24]. Зокрема, перший аргумент *PR(x)* – профіль невідомого тексту, а другий аргумент – авторський профіль.

Отже, лише n-грами профілю невідомого тексту вносять до розрахункової суми. У результаті, проблеми, описані раніше з незбалансованими корпусами, значно зменшуються, оскільки відстань між невідомим текстом та авторами-кандидатами завжди базується на одній і тій же кількості доданків. Більше того, кожен доданок множиться на відносну відстань питомої n-грамової частоти від норми корпусу [25]. Отже, чим більше n-грам відхиляється від його «нормальної» частоти, тим більш точнішою є відстань. З іншого боку, якщо частота n-грам виявляється точно такою ж, як і «нормальна» частота, вона взагалі не впливає на значення відстані (коефіцієнт норми дорівнює нулю).

Експерименти, про які повідомляє Stamatatos (2007), показали, що запропонована функція відстані може краще працювати у випадках, коли були обмежені та незбалансовані корпуси, що доступні для навчання. Крім того, ця відстань є досить стабільною щодо параметра L. Однак у випадках, коли було доступно достатньо навчальних текстів, оригінальний метод CNG дає кращі результати.

1.6 Моделі на основі подібності

Основна ідея моделей на основі подібності – це обчислення парної подібності між невідомим текстом і всіма навчальними текстами з подальшою оцінкою найбільш ймовірного автору на основі алгоритму найближчого сусіда.

Найпомітніший метод для такого підходу був запропонований Burrows (2002) під назвою Delta. На першому кроці цей метод обчислює розподіл набору функціональних слів (спочатку 150 найпоширеніших слів). Тоді у кожному документі відхилення частоти кожного слова від норми обчислюється у виразі z-оцінка, орієнтовно вказуючи, чи використовується вона більше (позитивна z-оцінка) або менше (негативна z-оцінка) разів, ніж в середньому [26].

Нарешті, міра Delta, яка вказує на різницю між набором (навчальних) текстів, написаних одним і тим же автором, і невідомим текстом – це середнє значення абсолютної різниці між z-балами для всього функціонального слова, встановленого в навчальних текстах, і відповідними z-балами невідомого тексту [27]. Чим менша міра Delta, тим більше стилістична подібність між невідомим текстом і автором-кандидатом. Цей метод був переважно перевіреним на літературних текстах (англійські вірші та романи), даючи неабиякі результати (Берроуз, 2002; Гувер, 2004a). Було продемонстровано, що це дуже ефективний

метод атрибуції для текстів принаймні з 1500 слів. Для коротших текстів точність падає відповідно до довжини [28]. Однак навіть для досить коротких текстів правильний автор зазвичай був включений у перші п’ять позицій рейтингових авторів, що забезпечує засіб скорочення набору авторів-кандидатів. Теоретичне дослідження аспектів застосування методу Delta описав Аргамон (2008). Більш детально він показав, що міру Delta можна розглядати як відстань, зважену на осі класифікації найближчого сусіда, де невідомий текст присвоюється найближчій категорії замість найближчого навчального тексту [29].

Було також показано, що дистанційний рейтинг кандидата в автори, вироблений за мірою Delta [30], є еквівалентним ранжуванню ймовірності за припущенням, що частоти слів відповідають розподілу Лапласа. Цей погляд вказує на багато розширень та узагальнень методу Delta, наприклад, з використанням гаусових розподілів частот слів замість розподілу Лапласа тощо [30]. Детальне вивчення варіантів Delta від Burrows’ було представлено Hoover (2004). Він виявив, що при використанні більших наборів частоти слів (> 500) точність методу зростала [31]. Продуктивність також була покращена, коли було забезпечено усунення більшості особистих займенників у текстах [32]. Були також вивчені деякі інші варіанти методу Delta, але вони не були суттєвими з огляду на покращення оцінки в порівнянні з оригінальним методом [33].

Інший метод на основі подібності [34], що використовує моделі стиснення тексту для оцінки різниці між текстами, було описано Бенедетто. Навчальний етап цього методу просто включає стиснення кожного навчального тексту в окремих файлах за допомогою нестандартного алгоритму (GZIP). Для оцінки автора невідомого тексту [35] цей текст є з'єднується з кожним навчальним текстовим файлом, а потім кожен отриманий файл стискається таким самим алгоритмом [36].

Нехай $C(x)$ – бітовий розмір стиснення файлу x, тоді як $x + y$ – конкатенація текстових файлів x і y. Тоді різниця $C(x + y) - C(x)$ вказує на схожість навчального тексту x з невідомим текстом y. Нарешті, рішенням, що відповідає найближчому сусіду, оцінюється швидше за все автор. Цей метод піддався різкій критиці декількома дослідниками, що свідчить про багато слабких місць. По-перше, метод виглядає занадто повільним, оскільки йому доводиться застосовувати алгоритм стиснення стільки разів, скільки є навчальних текстів. Зауважимо, що для удосконаленого профільного підходу Хмельова та Техана [40], алгоритм стиснення викликається стільки разів, скільки є кандидатів-авторів. Отже, цей час роботи буде значно нижчий, ніж для профільного методу на основі стиснення Бенедетто.

Багаточисельні експерименти з ідентифікації авторства показали, що підхід на основі стиснення у межах методики на основі профілю зазвичай перевершує відповідний метод, заснований на екземплярі. Важливим фактором, що сприяє цьому підходу, є те, що метод пошуку 1-го найближчого сусіда чутливий до шуму [41]. Однак з цією проблемою можна поборотися за допомогою використання для вибору найближчих сусідів відомих методик більшості голосів або зваженої схеми голосування.

Також варто відзначити, що GZIP – це алгоритм стиснення на основі словника, який використовує розсувне вікно 32K для побудови словнику [42]. Це означає, що якщо навчальний текст достатньо довгий, то початок цього документа буде ігноровано, коли GZIP намагається стиснути конкатенацію цього файлу з невідомим текстом [42]. Порівняльні експерименти на різних корпусах показали, що RAR алгоритм стиснення переважає GZIP у більшості випадків [43]. Альтернативна міра відстані для підходу на основі стиснення була запропонована Килибрасом і Вітанієм у 2005 році. Виходячи з поняття складності Колмогорова,

було визначено нормовану відстань стиснення (NCD) між двома текстами x і y таким чином:

$$\mathrm{N}CD(x,y) = \frac{C(x+y) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}. \quad (1.5)$$

Автори використовували цю метрику відстані та алгоритм стиснення BZIP2, щоб класифікувати літературні твори російською мовою. Для випадку чотирьох різних авторів було отримано відмінні результати [44]. Автори навіть спробували порівняно згрупувати відповідні переклади цих текстів англійською мовою, що теж показало хороші результати.

РОЗДІЛ 2 ОГЛЯД ТЕХНОЛОГІЙ ТА ЗАСОБІВ МАШИННОГО НАВЧАННЯ ДЛЯ ОБРОБКИ ПРИРОДНОЇ МОВИ

2.1 Перцептрон

Багатошаровий перцептрон представляє собою нейронну мережу, кожний шар якої створений з елементів-нейронів (власне, їх математичних моделей). Ці елементи бувають трьох типів: сенсорні (вхідні, S), асоціативні («приховані» шари, A) і реагуючі (вихідні, R). Багатошаровим цей тип перцептронів називається не тому, що складається з декількох шарів, бо вхідній і вихідній шари можна не застосовувати в коді, а вміщує декілька (зазвичай, не більше двох-трьох) навчених (A) слоїв.

Модель нейрона (будемо називати його просто нейрон) – це елемент мережі, що вміщує декілька входів, кожний з яких має вагу. Нейрон, передаючи сигнал, помножує сигнали на вагу і сумує отримані величини, після чого передає результат до іншого нейрону або на вихід мережі. Тут багатошаровий перцептрон має розбіжності. Його функція є сигмоїдною, вона видає знання про зміни часу від 0 до 1. Якщо сигмоїд відноситься до декількох функцій, ми отримує вигляд логістичної функції.

Декілька шарів, які можуть вчитися (точніше, налаштовуватися), легко застосовувати для апроксимації дуже складних нелінійних функцій, які мають потрібну область застосування [50]. Типова архітектура багатошарового перцептрону схематично зображена на рис. 2.1.

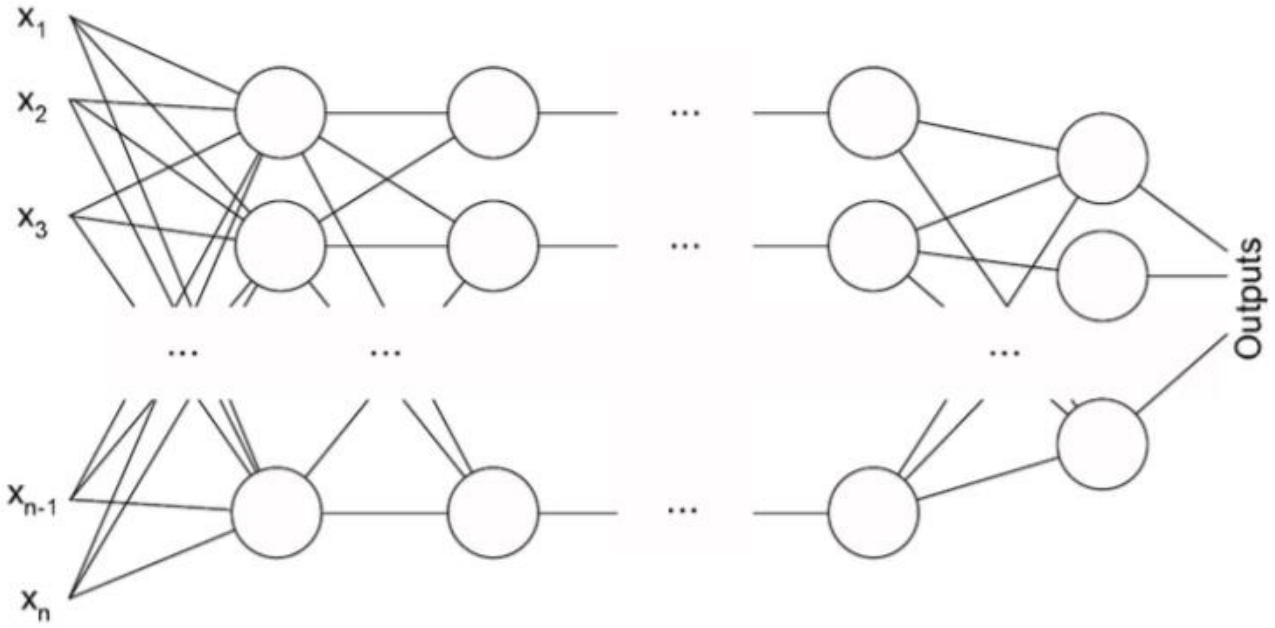

Рисунок 2.1 – Архітектура багатошарового перцептрону

2.2 Класифікатор

Як і в звичайних нейронних мережах, в загорткових мережах нейрони з повнозв’язних шарів (класифікаторів) мають зв’язок з усіма функціями активації з попереднього шару [55]. Активації одних і тих же шарів класифікації можна обчислити множенням матриць, що супроводжуються зміщенням. Різниця між класифікаційним шаром і шаром згортки полягає в тому, що нейрони шару згортки підключаються лише до локальної області на вході, і що нейрони цього шару можуть розділяти параметри як скалярний продукт [56], тому їх функціональна форма однакова. Більше того, допускаються різні комбінації між повнозв’язними та згортковими шарами.

Класифікатор повинен запам'ятати всі дані тренувань і зберігати їх для подальшого порівняння з даними тестового циклу. Це дуже дорогий ресурс, оскільки набори даних можуть бути в гігабайтах. Лінійний класифікатор (рисунок

2.2) дає оцінку класу як зважену суму всіх значень пікселів у трьох його кольорових каналах [57].

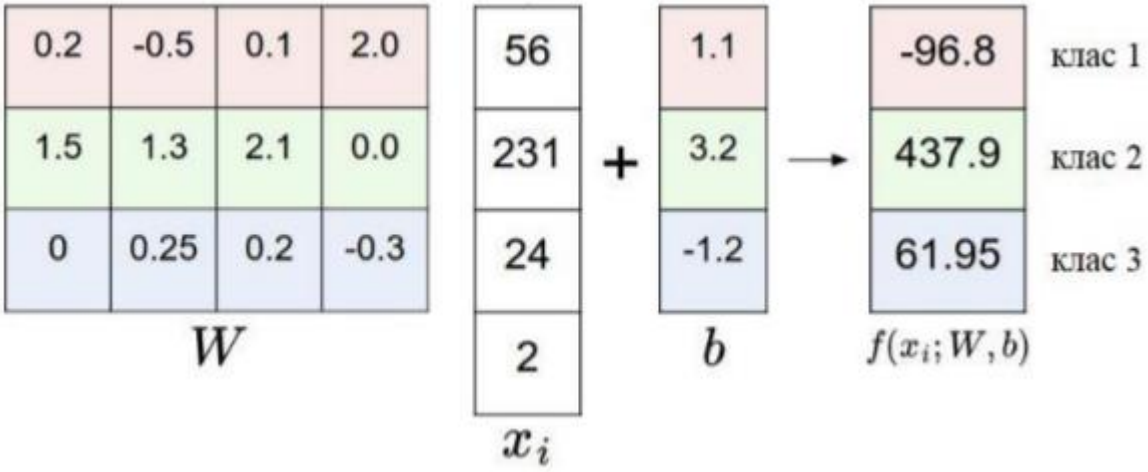

Рисунок 2.2 - Лінійний класифікатор

Залежно від того, які значення встановлені для цих шкал, функція класифікатора має можливість оцінювати позитивно чи негативно (залежно від ознаки кожної ваги) наявність певних кольорів у певних положеннях зображення. Наприклад, людина стверджує, що клас зображення «корабель» може бути більш імовірним, якщо на боках багато синього (що може відповідати воді).

Ще один спосіб організації класифікатора – за шаблоном. Особливість методу стосується ваг W і полягає в тому, що кожному рядку W відповідає шаблон (або іноді його ще називають прототипом) для одного з класів. Оцінка кожного класу зображень [60] отримується шляхом порівняння кожного шаблону по одному із зображенням, використовуючи внутрішню функцію (або точкову особливість), щоб знайти те, що найкраще працює. Використовуючи це порівняння, лінійний класифікатор відповідає шаблону.

2.3 Word2vec

Word2vec – це група споріднених моделей, які використовуються для створення ланцюгів зі слів. Ці моделі представляють собою неглибокі, двошарові нейронні мережі, які навчаються реконструювати мовні контексти слів. Word2vec приймає за свій вхід великий корпус тексту та створює векторний простір, як правило, розмірністю у декілька сотень, при цьому кожному унікальному слову в корпусі присвоюється відповідний вектор у просторі. Побудовані вектори розміщуються у векторному просторі так, що слова, які мають спільний контекст у корпусі, розташовуються близько один до одного у просторі [60].

Word2vec була створена та опублікована в 2013 році командою дослідників з Google під керівництвом Томаша Міколова та запатентована. Потім алгоритм був проаналізований та пояснений іншими дослідниками. Вбудовування векторів, створених за допомогою алгоритму Word2vec, має багато переваг порівняно з попередніми алгоритмами, наприклад, прихованим семантичним аналізом [61].

Word2vec може використовувати будь-яку з двох модельних архітектур, щоб створити розподілене подання слів: неперервний мішок слів (CBoW) або так званий Skip-gram. У архітектурі CBoW модель прогнозує поточне слово на основі поточного контексту навколишніх слів. Порядок контекстних слів не впливає на передбачення (припущення про сумнівні слова). У архітектурі Skip-gram модель використовує поточне слово для передбачення оточуючого вікна контекстних слів. Архітектура Skip-gram зважує на найближчі контекстні слова сильніше, ніж на більш віддалені контекстні слова. Згідно з приміткою авторів, CBoW швидша, а Skip-gram повільніша з моделей, проте демонструє кращі результати для нечастого слова.

Модель Word2vec може бути навчена за допомогою ієрархічної softmax або негативної вибірки. Щоб наблизити умовну ймовірність вірогідності журналу,

яку модель прагне максимально використовувати, ієрархічний метод softmax використовує дерево Хаффмана для зменшення обчислення. З іншого боку, негативний метод вибірки наближається до розв'язання проблеми максимізації, зводячи до мінімуму ймовірність журналу вибіркових негативних випадків. На думку авторів, ієрархічний softmax краще працює для рідкісних слів, тоді як негативна вибірка працює краще для частих слів і краще для векторів з низькими розмірами. Із збільшенням епох тренінгу ієрархічна softmax перестає бути корисною.

Запропоновано розширення word2vec для побудови векторів із цілих документів (а не окремих слів). Це розширення має назву para2vec або doc2vec і реалізовано в інструментах C, Python та Java / Scala, при цьому версії Java і Python також підтримують функції вбудовування документів у нові, невидимі документи.

Асгарі та Мофрад запропонували розширення векторів слів для n-грам у біологічних послідовностях (наприклад, ДНК, РНК та білків) для розв'язання задач біоінформатики. Названі біо-вектори (BioVec) для позначення біологічних послідовностей в цілому з білковими векторами (ProtVec) для білків (амінокислотні послідовності) та гени-вектори (GeneVec) для послідовностей з генів. Таке представлення може широко застосовуватися в аплікаціях машин навчання Proteomics та Genomics. Отримані результати свідчать про те, що BioVectors можуть характеризувати біологічні послідовності з точки зору біохімічних та біофізичних інтерпретацій основних моделей. Аналогічний варіант, dna2vec, показав, що існує кореляція між оцінкою подібності Needleman-Wunsch і схожістю косинуса векторів слів dna2vec.

Однією з найбільших проблем Word2Vec є те, як поводитися з невідомими або поза словниковими (OOV) словами та морфологічно схожими словами. Особливо це може бути проблемою в таких сферах, як медицина, де синоніми та споріднені слова можуть використовуватися залежно від бажаного стилю

рентгенолога, а слова, можливо, нечасто використовуються у великому корпусі. Якщо модель word2vec раніше не стикалася з конкретним словом, вона буде змушена використовувати випадковий вектор, який, як правило, є далеко не ідеальним відображенням.

IWE поєднує Word2vec з методикою семантичного відображення словника для вирішення основних проблем вилучення інформації з клінічних текстів, які включають неоднозначність стилю розповіді у вільному тексті, лексичні варіації, використання неграматичних та телеграфічних фаз, довільне впорядкування слів та часту появу скорочень і абревіатур. Особливий інтерес показала модель IWE, що навчалася на одному інституційному наборі даних та була успішно переведена на інший інституційний набір даних – це демонструє хорошу узагальненість підходу між медичними установами [63].

2.4 Загальний огляд NLTK

Natural Language Processing (далі – NLP) або обробка природної мови представляє собою спільний напрям досліджень в інформатиці та AI, присвячений формальним методам аналізу природних (людських) мов. NLP дозволяє застосовувати алгоритми машинного навчання для тексту й мови.

Наприклад, методи NLP використовуються, щоб створювати системи на кшталт розпізнавання мови, узагальнення документів, машинного перекладу, виявлення спаму, розпізнавання іменованих сутностей, відповідей на питання, автозавершення слів, інтелектуального введення тексту тощо.

Сьогодні у багатьох з користувачів є смартфони з розпізнаванням мови – в них застосовуються методи NLP для того, щоб розуміти нашу мову. Також багато

людей використовують ноутбуки з вбудованим в ОС розпізнаванням мови, зокрема:

а) Cortana – віртуальний помічник Windows;

б) Siri – це помічник для ОС від Apple;

в) Gmail – відомий поштовий сервіс вміє визначати спам, щоб він не потрапляв у папку «вхідні» поштової скриньки;

г) Dialogflow – платформа від Google, яка дозволяє створювати NLP-боти.

Наприклад, можна зробити бота для замовлення піци, якому не потрібен старомодний IVR, щоб прийняти замовлення клієнта.

Розглянемо основні підходи (поняття) NLP, що використовуються для обробки текстової інформації, а саме:

а) токенізація за реченнями.

б) токенізація за словами.

в) лематизації і стемінг тексту.

г) стоп-слова.

д) регулярні вирази.

е) мішок слів.

є) TF-IDF.

Вищезазначені підходи будемо аналізувати з огляду на можливості відомого лінгвістичного пакету NLTK [72].

а) Токенізація (іноді – сегментація) за реченнями – це процес поділу писемної мови на речення-компоненти. Ідея виглядає досить простою. В англійській і деяких інших мовах ми можемо виокремлювати речення кожний раз, коли знаходимо певний знак пунктуації – крапку.

Але навіть в англійській мові ця задача нетривіальна, так як крапка використовується і в скороченнях. Таблиця скорочень може сильно допомогти під час обробки тексту, щоб уникнути невірної розстановки кордонів речення. У

більшості випадків для цього використовуються бібліотеки, зокрема і в пакеті NLTK, так що користувач пакету може особливо не переживати про деталі реалізації.

б) Токенізацією (іноді – сегментацією) за словами називають процес поділу речень на слова-компоненти. В англійській і багатьох інших мовах, що використовують ту чи іншу версію латинського алфавіту, пропуск – це непоганий роздільник слів.

Проте, можуть виникнути проблеми, якщо ми будемо використовувати тільки пропуск – в англійській мові складові іменники пишуться по-різному, зокрема через пропуск. І тут знову нам допомагають бібліотеки, що входять до складу лінгвістичних пакетів. Зазвичай тексти містять різні граматичні форми одного і того ж слова, а також можуть зустрічатися однокореневі слова.

в) Лематизація і стемінг – це лінгвістичні процеси, які виконуються програмно за кількома відомими алгоритмами, що також підтримується пакетом NLTK. Не дивлячись на різні підходи, лематизація і стемінг мають на меті привести всі словоформи, що зустрічаються в тексті, до однієї, нормальної словникової форми.

г) Стоп-слова – це слова, які викидаються з тексту до / після комп’ютерної обробки тексту. Коли ми застосовуємо машинне навчання до текстів, такі слова можуть додати багато шуму, тому необхідно позбавлятися від нерелевантних (незначущих) слів.

Під стоп-словами зазвичай розуміють артиклі, вигуки, сполучники тощо, які не несуть смислового навантаження. При цьому треба розуміти, що не існує універсального списку стоп-слів, все залежить від конкретного випадку. У складі пакету NLTK є попередньо встановлений список стоп-слів. Перед першим використанням вам знадобиться його завантажити: nltk.download ("stopwords"). Після завантаження можна імпортувати пакет stopwords і переглянути ці слова.

д) Регулярний вираз (регулярка, regexp, regex) – це послідовність символів, яка визначає шаблон пошуку. Формально регулярний вираз в програмуванні – це рядок, що описує або збігається з множиною рядків, відповідно до набору спеціальних синтаксичних правил. Регулярні вислови використовуються в багатьох текстових редакторах та допоміжних інструментах для пошуку та зміни тексту на основі заданих шаблонів.

В лінгвістичному пакеті NLTK для роботи з регулярними виразами потрібно імпортувати бібліотеку *re,* скориставшись командою *import re*. Корпус слів англійської мови Words Corpus буде використовуватися в якості лінгвістичних даних, серед яких буде проводитися пошук. Попередня підготовка списку слів передбачає видалення власних імен.

е) Алгоритми машинного навчання не можуть безпосередньо працювати з «сирим» текстом, тому необхідно конвертувати текст в набори цифр (вектори). Цей процес має назву витягання ознак. Мішок слів (див. п.2.3) – це популярна і проста техніка вилучення ознак, яка використовується при роботі з текстом. Вона описує входження кожного слова в текст. Щоб використовувати модель, нам потрібно:

а) ідентифікувати словник відомих слів (токенів);

б) вибрати ступінь присутності відомих слів.

Будь-яка інформація про порядок або структуру слів ігнорується, ось чому такі структури називаються мішки слів. Ця модель намагається зрозуміти, чи зустрічається знайоме слово в документі, але не знає, де саме воно зустрічається. Інтуїція підказує, що схожі документи мають схожий вміст. Також, завдяки вмісту, ми можемо дізнатися дещо про сенс документа.

є) У частотних методів ранжування є проблема – слова з найбільшою частотністю мають, відповідно, найбільшу оцінку. Проте у цих словах (наприклад, стоп-словах) може бути не так багато інформаційного виграшу для

моделі, як в менш частих словах. Один із способів виправити ситуацію – знижувати оцінку слова, яке часто зустрічається у всіх подібних документах. Цей підхід відомий під назвою TF-IDF, причому власне TF-IDF (скорочення від term frequency - inverse document frequency) – це статистична міра для оцінки важливості слова в документі, який є частиною колекції або корпусу.

Ранжування слова за підходом TF-IDF зростає пропорційно частоті появи такого слова в документі, але це компенсується кількістю документів, що містять це слово.

РОЗДІЛ 3 РОЗРОБКА КЛАСИФІКАТОРА ДЛЯ ВИЗНАЧЕННЯ АВТОРСТВА ТЕКСТУ

3.1 Метод групового врахування аргументів

Автором методу групового обліку аргументів (МГОА) є академік НАН України О.Г. Івахненко [20]. Метод знайшов своє застосування в різних галузях знань, а саме з метою структурної і параметричної ідентифікації, прогнозування. МГОА заснований на самоорганізації моделей і, на відміну від регресійного аналізу, де подано структуру моделі, має на меті визначити структуру моделі оптимальної складності.

Побудова адекватного рівняння лінійної регресії та його вдосконалення потребує зростаючої ретроспективи (періоду статистичного огляду), що часто неможливо. Збільшення кількості факторів супроводжується «прокляттям виміру», сутність якого полягає в накопиченні загальної помилки. На основі використання аксіом замкненої системи можна довести існування заданої табличної залежності, яка не може бути апроксимована складом заданого набору структурних елементів.

Метод групового обліку аргументів реалізований у багатьох алгоритмах для вирішення різних завдань. Він включає параметричні алгоритми, алгоритми кластеризації, аналогові комплекси, ребінаризацію та ймовірнісні алгоритми. Підхід до самоорганізації, реалізований на МГОА, ґрунтується на виборі прогресивно складніших моделей та виборі найкращого рішення відповідно до мінімального значення зовнішнього критерію.

Базові моделі часто можуть бути не тільки поліномами ступеня більше 1, а й іншими нелінійними функціями. Переглядаючи різні рішення в індуктивному підході до моделювання, МГОА намагається мінімізувати роль впливу аналітика

на результати моделювання. Комп'ютер знаходить структуру моделі та закони, за якими функціонує об'єкт, отже використовується як радник для пошуку нових рішень у задачах штучного інтелекту. Особливості методу:

1. Область досліджень МГОА є ефективною, оскільки оптимальна складність структури моделі є адекватною рівню шумових перешкод у вибірці даних (спрощені моделі прогнозування є більш точними для вирішення проблем із зашумленими або «короткими» даними).

2. Оптимальну кількість шарів та нейронів у прихованих шарах, структуру моделі та інші значення параметрів нейронної мережі отримують автоматично, що спрощує застосування методу.

Автор МГОА запропонував використовувати принцип зовнішнього доповнення. Виходячи з теореми Вейерштрасса про те, що будь-яку безперервну функцію можна апроксимувати з будь-якою точністю через поліном певного ступеня, він запропонував таку схему.

3.1.1 Багаторядний метод групового врахування аргументів

Нехай початкові данні зосереджені в матриці $A = \left(X_1, X_{2,}, \dots, X_{n,}Y\right)$, де X_i, для всіх $\mathrm{i} = 1, n$ та Y – вектори-стовпчики розмірністю N, а Y – вихідна характеристика. Задача полягає в ідентифікації

$$A = \left(X_1, X_{2,}, \dots, X_n\right) \tag{3.1}$$

поліномом Колмогорова-Габора у загальному вигляді

$$Y = a_0 + \sum_{i=0}^{n} a_i x_i + \sum_{i,j=1}^{n} a_{ij} x_i x_j + \sum_{i,j,k=1}^{n} a_{ij} x_i x_j + \cdots \tag{3.2}$$

Деякі автори в своїх роботах наполягають, що поліном має мати трішки інший вигляд, щоб не містити симетричних опорних функцій. Тому Ю.П. Зайченко використовує поліном вигляду

$$Y = a_0 + \sum_{i=0}^{n} a_i x_i + \sum_{i=1}^{n} \sum_{j=i}^{i} a_{ij} x_i x_j + \sum_{i=1}^{n} \sum_{j=i}^{i} \sum_{k=1}^{j} a_{ij}\, x_i x_j x_k + \cdots \qquad (3.3)$$

а Є.В. Снитюк – ще більш обмежену модель

$$Y = a_0 + \sum_{i=0}^{n} a_i x_i + \sum_{i=1}^{n} \sum_{j=i}^{i} a_{ij} x_i x_j + \sum_{i=1}^{n} \sum_{j=i}^{i-1} \sum_{k=1}^{j-1} a_{ij}\, x_i x_j x_k + \cdots \qquad (3.4)$$

Пропоноване звуження суттєво обмежує кількість моделей, що будуються, однак суттєво деформує простір можливого рішення. Обмеження їх застосування розглянемо далі.

Відомо, що зі збільшенням порядку полінома точність наближення функції $F(x)$ збільшується, а потім зменшується. Іншими словами, збільшуючи максимальний ступінь многочлена, ми маємо таку форму залежності кореневої середньої квадратної похибки наближення від ступеня полінома (рис. 3.1). Якщо точність максимальна, то цей процес потрібно закінчити.

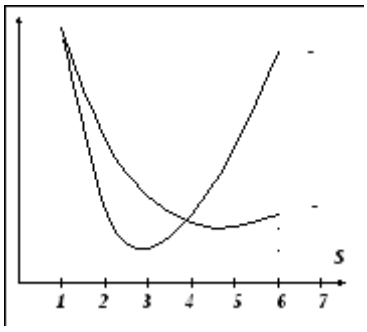

Рисунок 3.1 – Похибки апроксимації

Особливістю МГОА є те, що він може бути застосований у випадку невеликої кількості точок експерименту, навіть значно меншої за кількість членів

поліному. Це відбувається тому, що на кожному етапі моделювання наближення виконується не повним многочленом складності Y, а елементарною функцією підтримки.

Підтримуюча функція вибирається на першому етапі застосування МГОА.

Зазвичай в якості підтримуючих функцій використовуються наступні залежності:

а) мультиплікативна $y = a_0 + a_1 x_i x_j$;

б) адитивна $y = a_0 + a_1 x_i + a_2 x_j$;

в) повна першого порядку $y = a_0 + a_1 x_i + a_2 x_j + a_3 x_i x_j$;

г) повна другого порядку $y = a_0 + a_1 x_i + a_2 x_j + a_3 x_i^2 + a_4 x_j^2 + a_5 x_i x_j$.

Перша функція вимагає початкових даних принаймні трьох експериментів, для другої – 4, для третьої – 5, для четвертої – 7. Це справедливо тому, що метод визначення найменших квадратів використовується для отримання коефіцієнтів, при яких необхідно мати принаймні один ступінь свободи, щоб отримана залежність мала рівень довіри, відмінний від 0.

Позначимо $y_k = f(x_i, x_j)$, де f – одна із залежностей або схожа. На наступному етапі за допомогою методу найменших квадратів (МНК) визначають коефіцієнти p рівнянь**,** де p є залежним від підходу до побудови полінома. Зрозуміти формулу розрахунку p можна, виходячи з таких міркувань. Всі найбільш ймовірні пари індексів i та j утворюють матрицю (рис 3.2). Пари, які ми використовуємо при побудові k −функції, в залежності від методу, знаходяться:

a) у всій матриці, тоді Y визначається у адаптивні а $p = n^2$;

b) на основній діагоналі та вище, тоді Y визначається за залежністю повного порядку, а $p = \frac{n(n+1)}{2}$;

с) вище головної діагоналі, тоді Y визначається за залежністю повного другого порядку, а $\mathrm{p} = \frac{n(n-1)}{2}$.

		$j =$				
		1	2		n-1	n
$i =$	1	(1,1)	(1,2)	...	(1,n-1)	(1,n)
	2	(2,1)	(2,2)	…	(2,n-1)	(2,n)
	3	(3,1)	(3,2)	…	(3,n-1)	(3,n)
	…	…	…	…	…	…
	n	(n,1)	(n,2)	…	(n,n-1)	(n,n)

Рисунок 3.2 - Пари можливих індексів

Перший з цих підходів призводить до побудови $\mathrm{p} = \frac{n(n-1)}{2}$ «зайвих» моделей, адже при включенні в множину конкуруючих моделей $y_1 = f(x_1, x_2)$ та $y_1 = f(x_1, x_2)$ при застосуванні будь-якої підтримуючої функції будуть отримані дві рівні моделі з однаковими коефіцієнтами. Це не тільки зробить наші шанси увійти в наступне покоління різних моделей меншими (завжди вибираючи кращі та їх еквівалентні «близнюки»), але і майже помножить на 2 наші шанси у кількості розрахунків. З великим початковим розміром вибірки та $n \geq 10$ розрахунок непотрібних моделей за МНК може взяти великий відрізок часу, тож застосовувати $p = n^2$ нераціонально.

Різниця між підходами є у тому, що В.Є. Снитюк наполягає, обравши від початку максимальний порядок моделі надалі не нарощувати його шляхом заборони використання моделей з головної діагоналі матриці в таблиці 3.1. Наприклад, припустима за Ю.П. Зайченком модель $y_1 = f(x_1, x_2)$, додає до множини моделей x_1^2, повторне її застосування і x_1^4 тощо. Великий ріст порядку моделей, можливий при цьому процесі, може мати два результати – як позитивний, так і негативний з точки зору обраних критеріїв якості апроксимації.

При цьому різниця в кількості моделей, що розглядаються при підходах відрізняється несуттєво. Тож, обирати підхід зі зростанням максимального порядку полінома чи із забороною на таку дію – питання вибору експериментатора.

Після того, як усі залежності $y_k = f(x_i, x_j)$ для обраного рівня p ідентифіковані за МНК, до справи вступає зовнішній критерій, за яким із усієї множини відбирають найкращі моделі.

Щоб забезпечити роботу зовнішнього критерію, початковий зразок слід розділити на навчання та тестування. Залежно від обраного критерію якості, навчальний зразок повинен містити 40-60% рядків вихідної таблиці даних, але принаймні на один рядок, більший за порядок опорної функції.

Для негайного усунення «ненадійних» моделей застосовується відомий з дисперсійного аналізу критерій точності як обмеження.

3.2 Навчання штучної нейронної мережі

Найбільш відомі дослідження нейронних мереж (НМ) демонструють їхні унікальні можливості навчатись на даних про предметну область. Підвищення продуктивності НМ з розпізнавання образів відбувається за певними правилами, що враховують особливості предметної області.

Процес навчання може бути розглянутий як архітектурний, так і ітераційний – мета полягає у визначенні вагових коефіцієнтів синоптичних зав’язків для ефективного дослідження спеціальної задачі.

Під час навчання нейронної мережі, що використовується, потрібно виконати такі кроки для підвищення точності розпізнавання:

- У нейронній мережі задається початкова конфігурація з огляду на особливості задачі розпізнавання.

- На кожній ітерації змінюються вільні параметри нейронної мережі.

- Після зміни внутрішньої ваги нейрону вимірюють, що знаходиться на вихідних сигналах вже в наступній ітерації.

Правила навчання нейронної мережі називають алгоритмом навчання. Не існує універсального алгоритму навчання, придатного для всіх архітектурних конфігурацій НМ. Алгоритми навчання відрізняються один від одного способами створення синоптичних ваг нейронів.

Нейронна мережа автоматично встановлює ваги зв’язків для наявних навчальних множин. Властивість різних навчальних алгоритмів виявляється у їх більшій адаптивності у порівнянні із системами, які забезпечують розв’язання аналогічних задач розпізнавання, але формуються експертами.

Для того, щоб навчити модель, що використовується в НМ, потрібно застосувати таку функцію нейрона, яка є максимально ефективною в аналогічних задач розпізнавання. Всі відомі алгоритми навчання НМ базуються на правилах ітераційної зміни вагових коефіцієнтів.

Розглянемо три відомі загальні парадигми машинного навчання нейронних мереж:

а) контрольоване навчання (з вчителем).

б) неконтрольоване навчання (без участі вчителя, самонавчання).

в) змішане навчання.

У першому випадку НМ знаходиться в таких умовах розпізнавання, коли є відомими правильні відповіді (виходи НМ) для інших вхідних прикладів. Ваги налаштовують так, щоб отримати відповіді, що знаходяться якомога близько до відомих правильних відповідей.

Навчання без участі вчителя не вимагає інформації щодо правильних відповідей на попередньо відомих навчальних вибірках. У цьому випадку НМ розкриває внутрішню структуру даних та кореляцію між зразками у навчальних множинах, які дозволяють поділити нові зразки за категоріями.

При зміні навчальної вибірки частина зразків може використовуватися для навчання НМ шляхом зміни ваг, в той же час, коли інша частина зразків буде основою для оцінки якості розпізнавання.

В умовах контрольованого навчання велика кількість рішень отримується від НМ, де поточний вихід повинен порівнюватися з бажаним результатом. Ваги нейронів задаються початковими значеннями, але під час наступних ітерацій корегують для досягнення максимальної близькості між бажаними та поточними результатами. Іншими словами, нейрона мережа з контрольованим навчанням повинна спочатку пройти фазу навчання, яка може тривати досить довгий час. За протоколом обчислювальної системи, в залежності від потужності процесора, обсягу оперативної пам’яті та інших факторів фаза навчання НМ можна займати кілька годин і навіть діб.

Навчання має бути зафіксованим при досягненні нейромережею найкращого результату, зокрема як така множина значень ваг, що демонструють найбільше наближення до відомих результатів. Фактично це рівень бажаної статистичної точки у просторі заданих екземплярів навчальної вибірки.

Після навчання отримані ваги фіксуються для певного спеціального використання (розпізнавання). Відомі типи метрик дозволяють під час розпізнавання неперевірених екземплярів порівнювати їх з результатами навчальної вибірки та отримувати оцінку близькості цих екземплярів до відомих класів образів. В умовах задачі дослідження екземплярами образів є невідомі тексти, що потребують атрибуції, а класами – автори-кандидати.

Навчальні множини (вибірки) мають бути достатньо великими, щоб результати розпізнавання образів були релевантними. Крім того, бажано забезпечити можливість широко використовувати різноманітні дані, що не пов’язані кореляційними та іншими зв’язками.

Якщо визначено, що навчальний зразок ефективно використовується для одного варіанту образу, він встановлюється як еталон для цього прикладу і обов’язково застосовується в навчанні для наступного використання. Попередні менш ефективні приклади при навчанні в наступних ітераціях тоді забуваються. У результаті НМ вивчає всі можливі реалізації образів одного класу, знаючи найкращі вагомі коефіцієнти для загальної кількості відомих прикладів. Наприклад, у навчальних системах розпізнавання піксельних образів для десяти цифр, які представлені двадцятьма прикладами різних зображень кожної цифри, кожне зображення цифри «сім» не збігається з іншими. Для навчання мають бути відібрані найбільш оригінальні екземпляри цифри «сім» (так, як і інших цифр), причому їх кількість може бути і меншою за 20 – більш важливою для якості розпізнавання є оригінальність написання цифри.

Отже, головні компоненти образів для успішного їх розпізнавання мають бути представленими та використаними у навчальних даних. Ефективному розпізнаванню сприяє також відкидання неінформативних атрибутів вхідних даних, які тільки збільшують тривалість фази навчання. За необхідності здійснюється масштабування даних, що дозволяє нормалізувати їх за діапазоном можливих значень. Наприклад, попередня інформація про потоки даних, що надійшли від відповідних сенсорів, перетворюється в машинному форматі для стандартних комп'ютерів і є легко доступною для подальшого масштабування екземплярів навчальної вибірки НМ.

3.3 Застосування методів машинного навчання та методу МГОА для вирішення задачі дослідження

Основною метою роботи було поєднати методи лінгвістичного аналізу та машинного навчання нейронних мереж, чим перевірити результати і висновки бакалаврської роботи щодо інформативних ознак атрибуції авторства україномовного тексту. На основі співпраці з лабораторією комп'ютерної лінгвістики Київського Національного Університету імені Тараса Шевченка було отримано чисельні синтаксичні параметри, які описують стиль автора [3].

Для кращого розуміння підходу на рис. 3.3 представлено, як виглядає типова синтаксична структура речення у вигляді орієнтованого графа.

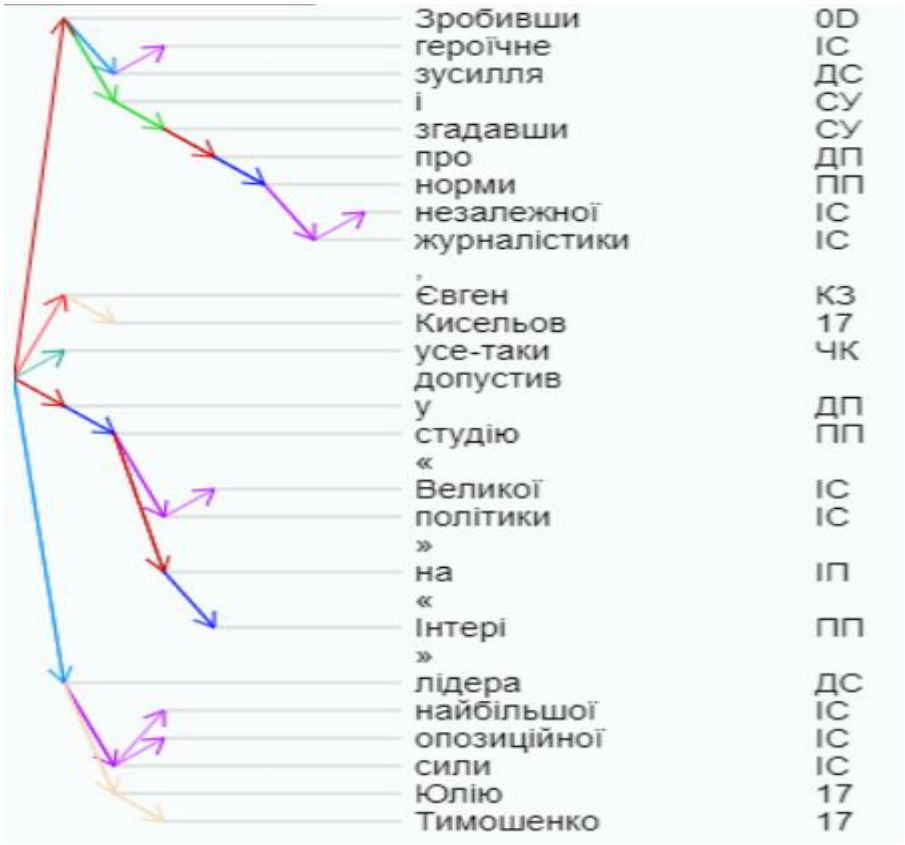

Рисунок 3.3 – Синтаксична структура речення у вигляді орієнтованого графа

Параметрами, які формально описують граф обрано:

- кількість вузлів у графі (словоформ) у реченні;
- кількість простих речень у складному;
- кількість рівнів у графі;
- максимальна кількість змін у шляху гілки графа;
- максимальна довжина дуги графа;
- загальна кількість вузлів у графі;
- середня кількість рівнів;
- середня кількість вузлів у рівні графа;
- співвідношення всіх вузлів речення, які не є термінальними (не є листями), до всіх вузлів цього речення;
- середня глибина гілки речення.

Для роботи було відібрано три автора – Микола Вінграновський, Іван Драч, Анатолій Мойсієнко. Як видно з рисунків (3.4 – 3.6), обрані параметри для цих авторів сильно відрізняються.

number of simple	the number of lev	width of branchin	the maximum number of changes	the maximum length	the total number c	average number c	the average numt	the ratio of all	average sentence
1.00	7.00	3.00	2.00	11.00	3.00	0.4	2.14	0.6	4.00
1.00	4.00	7.00	1.00	11.00	0.00	0.07	3.75	0.46	2.875
1.00	4.00	6.00	1.00	6.00	0.00	0.61	3.25	0.8	3.00
1.00	7.00	6.00	2.00	3.00	1.00	0.52	2.71	0.47	0.42
1.00	5.00	1.00	1.00	1.00	1.00	3,5	1.00	0.8	5.00
1.00	6.00	3.00	1.00	4.00	0.00	0.5	2.33	0.5	3,64
1.00	11.00	3.00	3.00	10.00	5.00	0.25	0.28	0.714	5.33
2.00	5.00	5.00	1.00	3.00	1.00	0.58	2.40	0.41	2.85
1.00	5.00	3.00	0.00	1.00	1.00	0.5	3,61	0.5	3,58
1.00	5.00	3.00	1.00	4.00	0.57	0.57	3,69	0.42	3,58
1.00	5.00	3.00	1.00	4.00	0.00	0.57	3,67	0.42	3.75
1.00	5.00	4.00	2.00	6.00	3.00	0.75	0.46	0.53	3.42
1.00	3.00	2.00	1.00	4.00	0.00	0.57	2.33	0.42	3,00
1.00	3.00	3.00	0.00	2.00	0.00	0.42	2.33	0.57	3,00
2.00	6.00	2.00	2.00	6.00	0.00	0.31	2.66	0.68	3,62
4.00	11.00	3.00	4.00	36.00	2.00	0.46	3.54	0.53	7.83
1.00	4.00	2.00	0.00	2.00	1.00	0.33	3,58	0.66	3,59

Рисунок 3.4 – Параметри М. Вінграновського

the number of	number of simple	the number of lev	width of branchin	the maximum nun	the maximum leng	the total number c	average number c	the average numt	the ratio of all	average sentence	branch depth
6.00	1.00	4.00	2.00	1.00	1.00	0.00	0.50	3.50	0.50	3.00	
20.00	2.00	6.00	2.00	2.00	10.00	3.00	0.50	3.52	0.50	3.56	
10.00	1.00	4.00	4.00	1.00	3.00	1.00	0.50	3.58	0.50	3.00	
15.00	1.00	10.00	2.00	1.00	3.00	2.00	2.66	3.50	0.73	3.59	
13.00	3.00	5.00	5.00	2.00	6.00	0.00	0.46	3.60	0.53	3.56	
10.00	1.00	5.00	3.00	1.00	2.00	1.00	0.30	2.00	0.70	4.00	
6.00	2.00	3.00	3.00	1.00	3.00	1.00	0.66	2.00	3.61	3.61	

Рисунок 3.5 – Параметри І. Драча

the number of	number of simple	the number of lev	width of branchin	the maximum nun	the maximum lenc	the total number c	average number c	the average numt	the ratio of all	average sentence
18.00	2.00	5.00	5.00	1.00	6.00	1.00	0.55	3,5	0.44	3,6
6.00	1.00	3.00	3.00	0.00	2.00	0.00	0.5	2.00	0.5	2.66
11.00	1.00	5.00	4.00	1.00	3.00	0.36	3,4	2.00	0.63	3,5
17.00	1.00	4.00	5.00	1.00	13.00	0.00	0.47	4.25	0.52	3,5
10.00	1.00	5.00	3.00	2.00	3.00	1.00	0.5	2.00	0.5	3,4
10.00	2.00	5.00	4.00	1.00	3.00	1.00	0.5	2.00	0.5	3,2
9.00	1.00	4.00	4.00	1.00	3.00	1.00	0.44	2.25	0.55	3,0
10.00	2.00	5.00	4.00	1.00	3.00	1.00	0.5	2.00	0.5	3,2
9.00	2.00	4.00	3.00	1.00	3.00	1.00	1.00	2.25	0.55	3,5
15.00	1.00	7.00	2.00	3.00	7.00	2.00	0.3	2.14	0.66	5,2

Рисунок 3.6 – Параметри А. Мойсеєнко

Для експерименту було відібрано синтаксичні параметри різних речень цих авторів у різних творах для більш чіткого розуміння стилю автора. На вхід програми атрибуції авторства подаються дані синтаксичних параметрів речень з творів автора. В цілому схема всього процесу зображена на (рисунку 3.7).

3.3.1 Вирішення задачі за допомогою бібліотеки машинного навчання Scikit-learn

Scikit-learn – це бібліотека Python, яка використовується для машинного навчання. Зокрема, це набір, як кажуть автори, простих і ефективних інструментів для аналізу даних і їх аналізу. Фреймворк побудований на основі декількох популярних пакетів Python, а саме NumPy, SciPy і matplotlib. Основною перевагою цієї бібліотеки є ліцензія BSD, під якою вона поширюється. Ця ліцензія дозволяє користувачу вирішувати, чи слід вносити зміни до початкового коду без будь-яких обмежень на комерційне використання.

Основною перевагою цього рішення є його доступність і простота – його легко використовувати навіть для початківців, а також відмінний вибір для більш простих завдань аналізу даних. Інші переваги:

а) scikit learn надає купу справді корисних утиліт для розбиття даних, обчислення загальної статистики та виконання навіть не дуже поширених матричних операцій.

Рисунок 3.7 - Схема програми

б) Scikit-learn має хорошу документацію та чистий, в основному послідовний API.

в) добре інтегрується з Numpy.

Мінуси:

а) модельний API, наданий Scikit-learn, не має великої гнучкості, а деякі моделі, такі як Random Forests, мають нестандартні або помилкові реалізації (зміна імпортів функцій у цьому випадку).

б) scikit learn не використовує апаратне прискорення, що робить його часом повільним, особливо для тренувальних моделей.

Незважаючи на мінуси дана бібліотека машинного навчання написана на мові програмування Python та легко з нею взаємодіє, тому я обрав її.

Для опрацювання великих об'ємів даних нам потрібно встановити бібліотеку NumPy – це бібліотека для мови програмування Python, яка забезпечує підтримку великих, багатовимірних масивів та матриць, а також велику колекцію математичних функцій високого рівня для роботи над цими масивами.

Також нам потрібна бібліотека SciPy це безкоштовна бібліотека Python з відкритим кодом, яка використовується для наукових та технічних обчислень. SciPy містить модулі для оптимізації, лінійної алгебри, інтеграції, інтерполяції, спеціальних функцій, FFT, обробки сигналів та зображень, вирішення ODE та інших завдань, що є загальними в науці та техніці. Також нам потрібна Pandas – це бібліотека програмного забезпечення, написана мовою програмування Python для маніпулювання та аналізу даних. Зокрема, він пропонує структури даних та операції для управління числовими таблицями та часовими рядами. Це безкоштовне програмне забезпечення, випущене за ліцензією BSD. Зрештою ми завантажуємо саму бібліотеку Scikit-learn.

Далі готуємо дані, у нас буде два масиви даних – за одним мережа буде навчатися, другий – це тестові дані, які вміщують тексти авторів у форматі синтаксичних параметрів.

Спочатку імпортуємо бібліотеки та вказуємо шлях до файлів з навчальними даними та тестовими.

```
import numpy as np
import scipy as sc
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.preprocessing import MinMaxScaler

train_data_filename = 'train_data.csv'
test_data_filename = 'test_data.csv'

names = ['Author',
         'number of nodes in sentence',
         'number of simple sentences in complex',
         'number of levels in graph',
         'width of branching at root',
         'maxnum of changes in path of branch',
         'max length of arc of graph',
         'total number of nodes in graph',
         'avg number of levels',
         'avg number of nodes in graph level',
         'ratio of all aterminal nodes to all nodes',
         'avg sentence branch depth']
```

Наступний крок – це описування формату перетворення текстових даних у дискретний формат:

```
author = {'Drach': 0,'Moisienko': 1, 'Vingranovsky': 2}
```

Записуємо дані про навчання та тестування з файлів у відповідні масиви:

```
train_dataset = pd.read_csv(train_data_filename,names=names)
test_dataset = pd.read_csv(test_data_filename,names=names)
train_dataset.Author = [author[item] for item in train_dataset.Author]
test_dataset.Author = [author[item] for item in test_dataset.Author]
train_array = train_dataset.values
test_array = test_dataset.values
```

Форматування вище створених масивів для отримання відповідних масивів вхідних та вихідних параметрів:

```
train_X = train_array[:,1:12]
test_X = test_array[:,1:12]
train_Y = train_array[:,0]
test_Y = test_array[:,0]
```

Далі нам потрібно нормалізувати дані:

```
scaler = MinMaxScaler(feature_range=(0, 1))
rescaled_train_X = scaler.fit_transform(train_X)
rescaled_test_X = scaler.fit_transform(test_X)
```

Далі підходимо до самого кода класифікатора, створюємо багатошаровий класифікатор:

```
mlp = MLPClassifier(hidden_layer_sizes=(150,100,50), max_iter=3000,activation =
'relu',solver='adam',random_state=1)
```

Далі наповнюємо модель даними на навчання:

```
mlp.fit(rescaled_train_X,train_Y

predict_train = mlp.predict(rescaled_train_X)
predict_test = mlp.predict(rescaled_test_X)
```

Далі показуємо результати по навчальним даним та даним для тренування:

```
print(confusion_matrix(train_Y,predict_train))
print(classification_report(train_Y,predict_train))

print(confusion_matrix(test_Y,predict_test))
print(classification_report(test_Y,predict_test))
```

Результати отримано дуже добрі – нейронна мережа одразу показала достовірність 95% (Рисунок 3.8). Схему програми представлено на рисунку 3.9.

```
C:\Users\fkca2\Desktop\nn_classifier\venv\Scripts\python.exe C:/Users/fkca2/Desktop/nn_classifier/nn_classifier.py
[[50  0  0]
 [ 0 50  0]
 [ 0  0 50]]
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00        50
         1.0       1.00      1.00      1.00        50
         2.0       1.00      1.00      1.00        50

    accuracy                           1.00       150
   macro avg       1.00      1.00      1.00       150
weighted avg       1.00      1.00      1.00       150

[[32  5  0]
 [ 0 37  0]
 [ 0  2 35]]
              precision    recall  f1-score   support

         0.0       1.00      0.86      0.93        37
         1.0       0.84      1.00      0.91        37
         2.0       1.00      0.95      0.97        37

    accuracy                           0.94       111
   macro avg       0.95      0.94      0.94       111
weighted avg       0.95      0.94      0.94       111


Process finished with exit code 0
```

Рисунок 3.8 – Результати нейронної мережі

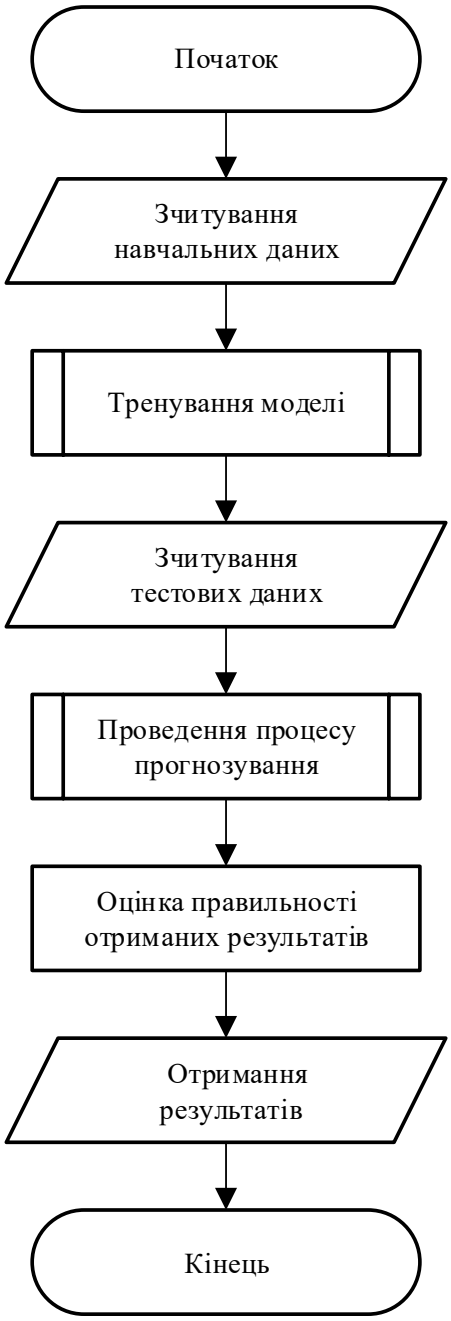

Рисунок 3.9 – Схема роботи програми з використанням нейронної мережі

3.1.2 Отримання результату на основі алгоритму МГУА

Основна ідея методу МГУА схожа з алгоритмами навчання нейронних мереж, тільки моделлю навчання є дерево, а не нейронна мережа. Роботу методу

порівнюють з тим, як рослини протягують коріння до води – якщо десь біля певного кореня є постійне джерело води, то цей корінь товстішає, а якщо в цьому місці сухо, то він тоншає і всихає. МГОА так само, якщо якийсь аргумент впливовіший, то він множиться на більше число, якщо малозначимий – на менше. Завдання на етапі навчання – визначити, який з параметрів на скільки впливає на кінцевий результат.

Для першого кроку також було взято два масиви, один з вхідними даними згідно рисунків 3.4 -3.6, та тестові дані, які були відібрані з загального масиву, так як на цей момент я не маю великої кількості параметрів авторів. Для тесту було відібрано 10 речень кожного автора (рисунок 3.10).

Мойсієнко 77	-10	9	1	4	4	1	3	1	0,44	2,25	0,55	3
Мойсієнко 78	-10	10	2	5	4	1	3	1	0,5	2	0,5	3,2
Мойсієнко 79	-10	12	2	4	3	1	3	1	1	2,25	0,55	3,5
Мойсієнко 80	-10	16	1	7	2	3	7	2	0,3	2,14	0,66	5,2
Мойсієнко 81	-10	15	1	7	2	3	7	2	0,3	2,14	0,66	5,2
Мойсієнко 82	-10	10	1	5	3	2	3	1	0,5	2	0,5	3,4
Мойсієнко 83	-10	10	2	5	4	1	3	1	0,5	2	0,5	3,2
Мойсієнко 84	-10	9	1	4	4	1	3	1	0,44	2,25	0,55	3
Мойсієнко 85	-10	10	2	5	4	1	3	1	0,5	2	0,5	3,2
Мойсієнко 86	-10	12	2	4	3	1	3	1	1	2,25	0,55	3,5
Вінграновський 79	-10	5	1	5	1	1	1	1	1/5	1	0.8	5
Вінграновський 80	-10	14	1	6	3	1	4	0	0.5	2.33	0.5	3.7
Вінграновський 81	-10	21	1	11	3	3	10	5	0.25	0.28	0.714	5.33
Вінграновський 82	-10	12	2	5	5	1	3	1	0.58	2,4	0.41	2.85
Вінграновський 83	-10	8	1	5	3	0	1	1	0.5	1.6	0.5	3.5
Вінграновський 84	-10	14	1	5	3	1	4	0.57	0.57	2.8	0.42	3.5
Вінграновський 85	-10	14	1	5	3	1	4	0	0.57	2.8	0.42	3.75
Вінграновський 86	-10	15	1	5	4	2	6	3	0.75	0.46	0.53	3.42
Вінграновський 87	-10	7	1	3	2	1	4	0	0.57	2,33	0.42	3
Вінграновський 88	-10	7	1	3	3	0	2	0	0.42	2,33	0.57	3
Драч 93	10	17	3	5	5	2	6	0	0.46	2.6	0.53	3.33
Драч 94	10	10	1	5	3	1	2	1	0.3	2	0.7	4
Драч 95	10	6	2	3	3	1	3	1	0.66	2	2.6	2.5
Драч 96	10	6	2	3	3	1	3	1	0.66	2	2.6	2.5
Драч 97	10	13	3	5	5	2	6	0	0.46	2.6	0.53	3.33
Драч 98	10	10	1	5	3	1	2	1	0.3	2	0.7	4
Драч 99	10	6	2	3	3	1	3	1	0.66	2	2.6	2.5
Драч 100	10	10	1	4	4	1	3	1	0.5	2.5	0.5	3
Драч 101	10	10	1	10	2	1	3	2	2.66	1.5	0.73	5.5
Драч 102	10	13	3	5	5	2	6	0	0.46	2.6	0.53	3.33

Рисунок 3.10 – Тестові дані для алгоритму МГУА

На виході отримані моделі показують нестабільність моделей на точках спостереження. Але тексти розбивались на речення, причому характеристики окремого речення утворювали вектор характеристик, що подавався як рядок в масиві (точка в багатовимірному просторі ознак). Це означає, що окремі речення мають різні властивості. Це і зрозуміло, бо окреме речення має свій зміст.

В загальному результати також отримано гарні. Словник ознак підібрано вдало. Межа інформативної достатності масиву вхідних даних перевищується за результатами випробування моделей-класифікаторів, кількість правильно класифікованих точок спостереження достатня, щоб правильно класифікувати тексти в цілому (рисунок 3.11).

Мойсієнко	Результат	Вінграновський	Результат	Драч	Результат
-0,838841777	0	-9,452837995	1	-4,63644683	1
5,43485192	1	-9,215371344	1	-6,535968421	1
9,490017091	1	-8,965253295	1	-10,21520101	1
9,679178561	1	-7,173696854	1	-9,081437844	1
9,674322794	1	-7,465227715	1	-9,081437844	1
10,92877734	1	-9,912205063	1	-6,132787341	1
5,43485192	1	-9,215371344	1	-6,535968421	1
-0,838841777	0	-9,452837995	1	-4,63644683	1
5,43485192	1	-9,215371344	1	-6,535968421	1
9,490017091	1	-8,965253295	1	-10,21520101	1
-9,585205982	1	7,504215039	1	-9,77265416	1
-3,732990179	1	8,566666043	1	-5,916235136	1
2,602581617	0	6,380756829	1	-3,462892186	1
-7,71584839	1	4,776869905	1	-7,263811705	1
-7,320355897	1	9,86998084	1	-9,02613012	1
-5,362515209	1	-8,244848369	0	-7,092175545	1
-5,592998979	1	8,144605935	1	-7,092175545	1
-8,537733392	1	8,820832673	1	-7,48792009	1
-9,559387173	1	-8,82085707	0	-6,814692161	1
-6,291477883	1	9,770904524	1	-9,928450064	1
-9,506860578	1	-8,51349922	1	10,20464318	1
-8,337961264	1	-8,815593061	1	9,37328861	1
-9,816400328	1	-8,593943169	1	10,20725501	1
-9,816400328	1	-8,593943169	1	10,20725501	1
-9,602853	1	-9,773967609	1	10,20464318	1
-8,337961264	1	-8,815593061	1	9,37328861	1
-9,816400328	1	-8,593943169	1	10,20725501	1
0,333346283	1	-9,178381398	1	-5,787957436	0
-9,344067875	1	-9,181156558	1	9,886701053	1
-9,602853	1	-9,773967609	1	10,20464318	1
	90%		93%		97%

Рисунок 3.11 - Результати методу МГОА

Якщо застосувати критерій сукупної оцінки (віднесення тексту до класу за результатами розпізнавання більшості його точок) - то кожен автор визначений безпомилково. Схема програми (рисунок 3.12)

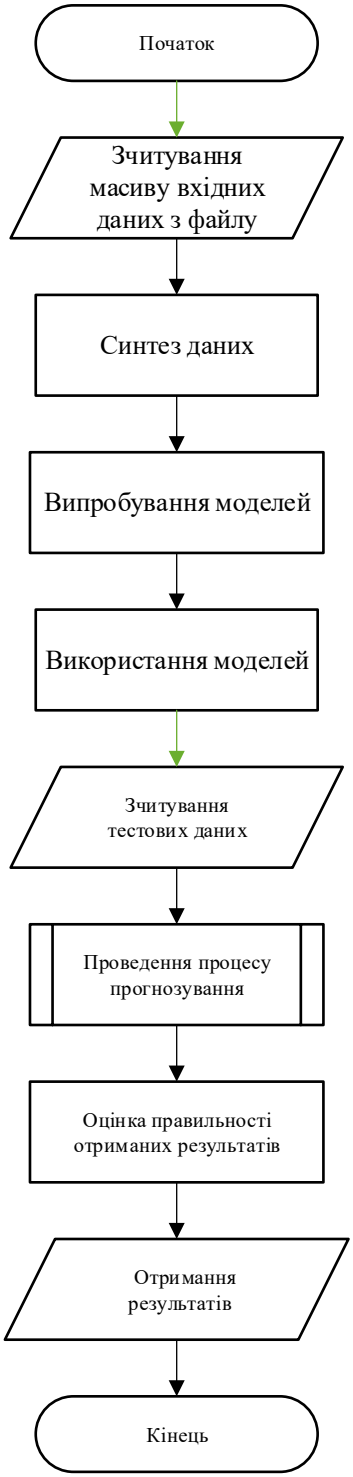

Рисунок 3.12 – Схема роботи програми з використанням алгоритму МГУА

Отже, збіг позитивних результатів машинного навчання за допомогою нейронної мережі та методу МГОА демонструє інформативність обраних

формальних ознак синтаксичної структури речення україномовного тексту для атрибуції авторства та підтверджує ефективність запропонованого методу визначення авторства україномовного тексту.

4 ЕКОНОМІЧНИЙ РОЗДІЛ

4.1 Технологічний аудит розробленого методу визначення авторства україномовного тексту

Як було зазначено у попередніх розділах роботи, неабиякої актуальності набуло питання суттєвого покращення результатів лінгвістичних експертиз текстової інформації. Це можна зробити шляхом автоматизації процесів лінгвістичної експертизи (тобто визначення авторства тексту), що дозволить підвищити якість формування профілів учасників соціальних мереж і роз-биття їх на категорії, виявлення плагіату, оперативного визначення недоброчесних або зловмисних дій користувачів інформаційних систем тощо. Проведений нами аналіз показав, що рівень розвитку відповідних лінгвістичних моделей і технологічних засобів для української мови на сьогодні є недостатнім.

Тому перед нами було поставлене завдання підвищити якість визначення авторства україномовного тексту на основі методів і моделей комп'ютерної лінгвістики та машинного навчання, а також використання доступних програмних бібліотек і технологічних засобів.

Для досягнення поставленої мети нами було: проаналізовано застосування методів машинного навчання в задачах комп'ютерної лінгвістики; зроблено порівняльний аналіз існуючих систем лінгвістичної експертизи; побудовано лінгвістичну модель та розроблено метод визначення авторства на основі аналізу зв'язків між лексичними одиницями україномовного тексту; досліджено можливості мови програмування Python у сфері аналізу статистичних даних та машинного навчання; побудовано програмне забезпечення, за допомогою якого здійснюється встановлення авторства україномовного тексту.

Підсумком виконаної нами магістерської кваліфікаційної роботи є роз-робка, за результатами удосконаленого статистичного аналізу тексту, мето-дики підготовки даних та навчання нейронної мережі, що дає можливість більш достовірно та швидко визначати авторство україномовного тексту для обраної групи авторів.

Для встановлення технічного рівня та комерційного потенціалу отриманих в ході проведених досліджень результатів проведемо їх технологічний аудит. Для проведення аудиту були (з дозволу наукового керівника д.т.н., професора Бісікала О.В.) запрошені три експерти, які хоча ще і не закінчили вищий навчальний заклад, але мають значний досвід у розв’язанні цієї проблеми і є фахівцями з цього питання. Це студенти групи 1АКІТ-18. Захарчук Олександр, Комар Андрій та Віталій Маслов.

При проведенні технологічного аудиту експерти користувалися рекомендаціями Державного комітету України з питань науки, інновацій та інформатики , наведеними в таблиці 4.1.

Таблиця 4.1 – Критерії для проведення технологічного аудиту розробки та їх бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-тер.	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни Аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів

Продовження таблиці 4.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Крит ерій	0	1	2	3	4
Ринкові перспективи					
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх Штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Закінчення таблиці 4.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-тер.	0	1	2	3	4
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки пові-домлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати проведеного технологічного аудиту зведено в таблицю 4.2.

Таблиця 4.2 – Результати технологічного аудиту нашої розробки

Критерії	Прізвище, ініціали експерта		
	Захарчук О.	Комар А.	Маслов В.
	Бали, виставлені експертами:		
1	3	3	3
2	4	4	4
3	4	4	4
4	4	4	4
5	4	4	4

6	4	4	4
7	3	4	4
8	3	3	4
9	3	3	3
10	3	3	3
11	3	3	3
12	3	3	3
Сума балів	$СБ_1$ =41	$СБ_2$ = 43	$СБ_3$ = 43

Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_1^3 СБ_i}{3} = \frac{41+42+43}{3} = \frac{126}{3} = 42.$

На підставі рекомендацій, наведених в (див. табл. 4.3), можна зробити висновок, що розроблений нами метод визначення авторства україномовного тексту має технічний рівень та комерційний потенціал, який характеризується як «високий».

Таблиця 4.3 – Технічні рівні та комерційний потенціал розробки

Середньоарифметична сума балів $\overline{СБ}$, розрахована на основі висновків експертів	Технічний рівень та комерційний потенціал розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Такий високий рівень нашої розробки пояснюється тим, що нами запропоновано такий новий метод визначення авторства україномовного тексту, який, на відміну від існуючих, базується на лінгвістичній моделі побудови графу

зв’язків між лексичними одиницями тексту та застосуванні методів машинного навчання за новими формальними ознаками тексту, що дозволяє підвищити якість визначення авторства україномовного тексту.

4.2 Розрахунок витрат на проведення досліджень

Для визначення витрат на проведення наукових досліджень та розробку методу визначення авторства україномовного тексту розрахуємо такі основні статті:

а). Основна заробітна плата $З_о$ дослідників:

$$З_о = \frac{М}{Т_р} \cdot t \text{ [грн]}, \qquad (4.1)$$

де М – місячний посадовий оклад конкретного виконавця, (грн.)

Величини окладів знаходиться в межах (4173…20000) (грн.) за місяць;

$Т_р$ – число робочих днів в місяці; приймемо $Т_р = 20$ днів;

t – число робочих днів роботи виконавців роботи.

Зроблені розрахунки зведемо до таблиці 4.4:

Таблиця 4.4 – Основна заробітна плата виконавців роботи

Найменування посади виконавця	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів (або годин) роботи	Витрати на оплату праці, грн	Примітка
1. Керівник магістерської роботи – д.т.н., професор	20000	1000	25 годин	≈4167	Робочий день – 6 годин
2. Студент-магістрант	2000	100	60	6000	
3. Консультант з ЕЧ	13000	650	2,5 годин	≈271	Робочий день – 6 годин
4. Інші працівники	7000	350	2	700	
Всього				$З_о$=11138 грн	

б). Додаткова заробітна плата $З_д$ виконавців:

$$З_д = (0{,}1 \ldots 0{,}12) \cdot З_о. \quad (4.2)$$

Для нашого випадку будемо мати:

$$З_д = 0{,}1 \times 11138 \approx 1114 \text{ [грн]}.$$

в). Нарахування на заробітну плату $Н_{зп}$ виконавців:

$$Н_{зп} = (З_о + З_д) \cdot \frac{\beta}{100}, \quad (4.3)$$

де β – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, %. β = 22%. Тоді:

$$Н_{зп}=(11138+1114) \times 0{,}22 = 2695{,}44 \approx 2696 \text{ [грн]}.$$

г). Амортизація А обладнання, комп'ютерів та приміщень розраховується за формулою :

$$А = \frac{Ц \cdot Н_а}{100} \cdot \frac{Т}{12} \text{ [грн]}, \quad (4.4)$$

де Ц – загальна балансова вартість основних засобів, які використовувалися;

$Н_а$ – річна норма амортизаційних відрахувань. Спрощено можна прийняти,

Найменування основних засобів	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн.
1. Обладнання, комп'ютер, принтер тощо	32500	22	3 (при 50% використанні)	893,75
2. Приміщення кафедри та факультету	11500	4,5	3 (при 50% використанні)	64,69
Всього				**А = 958,44≈959**

Т – термін, використання основних засобів, місяці.

Зроблені розрахунки зведемо у таблицю 4.5.

Таблиця 4.5 – Розрахунок амортизаційних відрахувань

д). Витрати на матеріали М розраховуються за формулою:

$$M = \sum_1^n H_i \cdot Ц_i \cdot K_i - \sum_1^n B_i \cdot Ц_в \text{ [грн.]}, \quad (4.5)$$

де $Н_і$ – витрати матеріалу *і*-го найменування, кг; $Ц_і$ – вартість матеріалу *і*-го найменування, грн./кг; $К_і$ – коефіцієнт транспортних витрат, $К_і$ = (1,1…1,15); $В_і$ – маса відходів матеріалу *і*-го найменування, кг; $Ц_в$ – ціна відходів матеріалу *і*-го найменування, грн/кг; n – кількість видів матеріалів.

Витрати на комплектуючі К розраховуються за формулою:

$$K = \sum_1^n H_i \cdot Ц_i \cdot K_i \text{ [грн.]}, \quad (4.6)$$

де $Н_і$ – кількість комплектуючих *і*-го виду, шт.; Ці – ціна комплектуючих *і*-го виду, грн; $К_і$ – коефіцієнт транспортних витрат, $К_і$ = (1,1…1,15); n – кількість видів комплектуючих.

Загальна вартість основних матеріалів та комплектуючих, які були використані під час проведення досліджень та розробки методу визначення авторства україномовного тексту, складає приблизно 975 грн.

е). Витрати на силову електроенергію $В_е$ розраховуються за формулою:

$$B_e = \frac{B \cdot П \cdot Ф \cdot K_п}{K_д}, \quad (4.7)$$

де В – вартість 1 кВт-год. електроенергії, в 2019 р. В ≈ 2,2 грн/кВт;

П – установлена потужність обладнання, кВт; П = 1,0 кВт;

Ф – фактична кількість годин роботи обладнання, годин.

Приймемо, що Ф = 170 годин;

$К_п$ – коефіцієнт використання потужності; $К_п$ < 1 = 0,81.

$К_д$ – коефіцієнт корисної дії, $К_д$ = 0,6.

Тоді витрати на електроенергію складуть:

$$B_e = \frac{B \cdot П \cdot Ф \cdot K_п}{K_д} = \frac{2{,}2 \cdot 1{,}0 \cdot 170 \cdot 0{,}81}{0{,}6} \approx 504 \text{ [грн.]}$$

ж). Інші витрати $В_{ін}$ можна прийняти як (100…300)% від суми основної заробітної плати виконавців, тобто:

$$В_{ін} = (1..3) \times З_о. \quad (4.8)$$

Для нашого випадку: $В_{ін}$ = 2,0 × 11138 = 22276 (грн).

и). Сума всіх попередніх статей витрат дає витрати на виконання цього етапу роботи – В.

В = 11138+ 1114 +2696 +959 + 975 + 504+ 22276 = 39662 (грн).

к) Загальні витрати ЗВ на проведення подальших досліджень та завершення роботи розраховуються за формулою:

$$ЗВ = \frac{В}{\beta}, \quad (4.9)$$

де β – коефіцієнт, який характеризує етап виконання роботи на шляху до її можливого впровадження. Якщо розробка знаходиться: на стадії науково-дослідних робіт, то $\beta \approx 0{,}1$; на стадії технічного проектування, то $\beta \approx 0{,}2$; на стадії розробки конструкторської документації, то $\beta \approx 0{,}3$; на стадії розробки технологій, то $\beta \approx 0{,}4$; на стадії розробки дослідного зразка, то $\beta \approx 0{,}5$; на стадії розробки промислового зразка, $\beta \approx 0{,}7$; на стадії впровадження, то $\beta \approx \approx 0{,}8$-$0{,}9$.

Оскільки наша розробка потребує ще певного доопрацювання, то для нашого випадку доцільно прийняти, що $\beta \approx 0{,}55$.

Тоді: $ЗВ = \frac{39662}{0{,}55} = 72112{,}27$ грн або приблизно 73 тис. (грн.)

Тобто прогнозовані витрати на проведення подальших досліджень та завершення нашої розробки становлять приблизно 73 тис. грн.

4.3 Розрахунок економічного ефекту від можливої комерціалізації розробленого методу визначення авторства україномовного тексту

Немає сумнівів, що розроблений нами метод визначення авторства україномовного тексту буде користуватися значним попитом на ринку, оскільки в умовах надшвидкого розвитку соціальних мереж все більше і більше осіб (як фізичних, так і юридичних) будуть зацікавлені у встановлені авторства того чи іншого україномовного тексту.

Оскільки розроблений нами метод визначення авторства україномовного тексту має значно кращі функціональні можливості та вищу ефективність, ніж аналогічні методи (оскільки використовує комбінації лінгвістичних досліджень та нейронних мереж), то нашу розробку можна буде реалізовувати на ринку дещо дорожче, ніж аналоги. Якщо середня ціна подібних розробок становить сьогодні приблизно 4 тис. грн, то нашу розробку можна буде продавати приблизно за 4,5 тис. грн, тобто на 0,5 тис. грн дорожче.

Аналіз місткості ринку аналогічної продукції, проведений нами, показав, що на цей час в Україні кількість реальних та потенційних користувачів подібних розробок складає щороку приблизно 200 осіб і їх кількість буде постійно зростати. Розроблений нами метод можна буде реалізовувати на ринку з 1 січня 2021 року (оскільки він потребує ще деякого доопрацювання), а результати від його впровадження будуть виявлятися протягом 2021-го, 2022-го та 2023-го років. Прогноз зростання попиту на нашу розробку складає по роках:

1-й рік після впровадження (2021 р.) – приблизно +100 шт.;

2-й рік після впровадження (2022 р.) – приблизно +200 шт.;

3-й рік після впровадження (2023 р.) – приблизно +300 шт.

Можливе збільшення чистого прибутку $\Delta П_i$, що його може отримати потенційний інвестор від фінансування та впровадження розробленого нами методу визначення авторства україномовного тексту становитиме:

$$\Delta П_i = \sum_1^n (\Delta Ц_o \cdot N + Ц_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\upsilon}{100}), \qquad (4.10)$$

де $\Delta Ц_o$ – покращення основного якісного показника від впровадження результатів розробки у цьому році. Таким показником є зміна ціни нової розробки, грн. Для нашого випадку це буде $\Delta Ц_o$ = (4,5 - 4) = 0,5 тис. (грн);

N – основний кількісний показник, який визначає обсяг діяльності у цьому році до впровадження результатів розробки; N = 200 шт.;

ΔN – покращення основного кількісного показника від впровадження результатів розробки. Таке покращення по роках становитиме, відповідно: +100 (2021 р.), +200 (2022 р.) та +300 (2023 р.) шт.;

$Ц_o$ – основний якісний показник, який визначає обсяг діяльності (тобто ціну розробки) у році після впровадження розробки, грн; $Ц_o$ = 4,5 тис. (грн);

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки; n = 3 роки;

λ – коефіцієнт, який враховує сплату податку на додану вартість; $\lambda = 0{,}8333$;

ρ – коефіцієнт, який враховує рентабельність продукту. Рекомендується приймати ρ = (0,2...0,5); візьмемо ρ = 0,5;

υ – ставка податку на прибуток. У 2019 році υ = 18%.

Величина чистого прибутку $\Delta П_1$ для потенційного інвестора протягом першого року від можливого впровадження нашої розробки (2021 р.) складе:

$$\Delta П_1 = [0{,}5 \cdot 200 + 4{,}5 \cdot 100] \cdot 0{,}8333 \cdot 0{,}5 \cdot (1 - \frac{18}{100}) = 187{,}91 \approx 188 \text{ тис. грн.}$$

Величина чистого прибутку $\Delta П_2$ для потенційного інвестора від можливого впровадження нашої розробки протягом другого (2022 р.) року складе:

$$\Delta П_2 = [0{,}5 \cdot 200 + 4{,}5 \cdot 200] \cdot 0{,}8333 \cdot 0{,}5 \cdot (1 - \frac{18}{100}) = 341{,}65 \approx 342 \text{ тис. грн.}$$

Величина чистого прибутку $\Delta П_3$ для потенційного інвестора від можливого впровадження нашої розробки протягом третього (2023 р.) року складе:

$\Delta П_З = [0{,}5 \cdot 200 + 4{,}5 \cdot 300] \cdot 0{,}8333 \cdot 0{,}5 \cdot (1 - \frac{18}{100}) = 495{,}39 \approx 496$ тис. грн.

Теперішня вартість інвестицій PV, що мають бути вкладені в нашу розробку: PV = (2…5) × ЗВ.

Для нашого випадку PV = 1,5 × 73 = 109,5 ≈ 110 тис. грн.

Абсолютний ефект вкладених інвестицій $Е_{абс}$ може становити:

$$Е_{абс} = ПП - PV, \quad (4.11)$$

де ПП – приведена вартість всіх можливих чистих прибутків від можливого впровадження нашої розробки, грн;

PV – теперішня вартість інвестицій PV = 110 тис. грн.

Приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$ПП = \sum_1^m \frac{\Delta П_i}{(1+\tau)^t}, \quad (4.12)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої роботи, грн;

т – період часу, протягом якого виявляються результати впровадженої роботи, роки. Для нашого випадку т = 3 роки;

τ – ставка дисконтування. Керуючись даними статистики за 2019 рік та Програмою дій Уряду України на наступні 5 років, приймемо ставку дисконтування τ = 0,04 (4%);

t – період часу від моменту отримання прибутків до початку впровадження розробки.

Тоді приведена вартість всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження нашої розробки, складе:

$$ПП = \frac{188}{(1+0,04)^2} + \frac{342}{(1+0,04)^3} + \frac{496}{(1+0,04)^4} \approx 174 + 304 + 424 = 902 \text{ тис. грн.}$$

Абсолютний ефект від можливого впровадження нашої розробки (при прогнозованому ринку збуту) за три роки складе:

$Е_{абс}$ = 902 – 110 = 792 тис. грн. або 264 тис. грн щорічно.

Далі розрахуємо внутрішню норму дохідності $Е_в$ вкладених коштів:

$$E_в = \sqrt[T_ж]{1 + \frac{E_{абс}}{PV}} - 1, \qquad (4.13)$$

де $Е_{абс}$ – абсолютний ефект вкладених коштів; $Е_{абс}$ = 792 тис. грн;

PV –теперішня вартість початкових інвестицій PV = 110 тис. грн;

$Т_ж$ – життєвий цикл розробки, роки. $Т_ж$ = 5 роки.

Для нашого випадку будемо мати:

$$E_в = \sqrt[4]{1 + \frac{792}{110}} - 1 = \sqrt[4]{1 + 7,2} - 1 = \sqrt[4]{8,2} - 1 = 1,69 - 1 = 0,69 = 69,0\%.$$

Далі визначимо ту мінімальну дохідність, нижче за яку потенційний інвестор не буде вкладати кошти в нашу розробку. Мінімальна дохідність або мінімальна (бар'єрна) ставка дисконтування $\tau_{мін}$ визначається за формулою:

$$\tau = d + f, \qquad (4.14)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2019 році в Україні d = (0,10...0,15);

f –показник, що характеризує ризикованість вкладень; f = (0,05...0,5), але може бути і значно більше.

Для нашого випадку отримаємо:

$\tau_{мін} = 0,15 + 0,5 = 0,65$ або $\tau_{мін} = 65\%$.

Оскільки величина $E_в = 69,0\% > \tau_{мін} = 65\%$, то потенційний інвестор може бути зацікавлений у фінансуванні нашої розробки.

Далі розраховуємо термін окупності коштів, вкладених у розробку та впровадження нашого нового методу визначення авторства україномовного тексту. Термін окупності $T_{ок}$ можна розрахувати за формулою:

$$T_{ок} = \frac{1}{E_в}. \quad (4.15)$$

Для нашого випадку термін окупності $T_{ок}$ коштів складе:

$$T_{ок} = \frac{1}{0,69} \approx 1,45 \text{ років},$$

що свідчить про потенційну доцільність комерціалізації нашої розробки.

Результати виконаної економічної частини магістерської кваліфікаційної роботи зведено у таблицю:

Показники	Задані у ТЗ	Досягнуті у магістерській кваліфікаційній роботі	Висновок
1. Витрати на розробку методу визначення авторства україномовного тексту	Не більше 80 тис. грн	73 тис. грн.	Досягнуто
2. Абсолютний ефект від впровадження розробленого методу визначення авторства україномовного тексту тис. грн	не менше 250 тис. грн за рік	264 тис. грн щорічно	Виконано
3. Внутрішня норма дохідності інвестицій, %	не менше 50%	69,0%	Досягнуто
4. Термін окупності, роки	до 3-х років	1,45 років	Виконано

Таким чином, основні техніко-економічні показники розробленого методу визначення авторства україномовного тексту, визначені у технічному завданні, виконані.

ВИСНОВКИ

В магістерській кваліфікаційній роботі було розв’язано такі задачі дослідження:

- Аналіз застосування методів машинного навчання в задачах комп’ютерної лінгвістики.
- Огляд та порівняльний аналіз існуючих систем лінгвістичної експертизи.
- Побудова лінгвістичної моделі та методу визначення авторства на основі аналізу зв’язків між лексичними одиницями україномовного тексту.
- Дослідження можливостей мови програмування Python у сфері аналізу статистичних даних та машинного навчання.
- Побудова програмного забезпечення та оцінка якості встановлення авторства тексту.

Внаслідок дослідження запропоновано новий метод визначення авторства україномовного тексту, який, на відміну від існуючих, базується на лінгвістичній моделі побудови графу зв’язків між лексичними одиницями речення тексту та застосуванні методів машинного навчання за новими формальними ознаками множини речень тексту, що дозволяє підвищити якість визначення авторства україномовного тексту.

Збіг позитивних результатів машинного навчання за допомогою нейронної мережі та методу МГОА демонструє інформативність обраних формальних ознак синтаксичної структури речення україномовного тексту для атрибуції авторства та підтверджує ефективність запропонованого методу визначення авторства україномовного тексту.

ЛІТЕРАТУРА

1. Стовбчатий М.М Застосування методів машинного навчання для визначення авторства україномовного тексту / М.М. Стовбчатий.Матеріали доповідей XLVIII науково-технічної конференції підрозділів. Вінницького національного технічного університету, 22–23 березня. – Вінниця : ВНТУ, 2019.

2. Стовбчатий, М.М., Бісікало О. В., «Застосування методів машинного навчання для визначення авторства україномовного тексту, » в Матеріали конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2020)», Вінниця, 2020. [Електронний ресурс]. Режим доступу: https://conferences.vntu.edu.ua/index.php/mn/mn2020/schedConf/presentations. Дата звернення: Груд. 2019

3. Розпізнавання мови [Електронний ресурс]:–Режим доступу: http://cybermova.com/speech – Назва з екрану.

4. Романов А. С. Методика и программный комплекс для идентификации автора неизвестного текста / А.С. Романов: Автореф. дис.канд. техн. наук. Томск, - 2010.- 26 с

5. Программный комплекс СМАЛТ / А.А Рогов, Г.Б Гурин, А. А Котов, Ю. В Сидоров, Т. Г Суровцева. Электронные библиотеки: перспективные методы и технологии, электротронные коллекции: Труды X Всерос . науч. конф. «RCDL'2008». Дубна, 2008. С. 155-160.

6. Марков А.А. Об одном применении статистического метода / А.А. Марков. Известия Императорской Академии наук. Сер. 6. 1916. Т. 10, № 4. С. 239-242.

7. Хмелев Д. В. Классификация и разметка текстов с использованием методов сжатия данных / Д.В Хмелев. Все о сжатии данных, изображений и видео. 2003. URL: http: //compression.ru/download/articles/classif/intro.html

8. Шевелёв А. Г. Разработка и исследование алгоритмов сравнения стилей текстовых произведений / А.Г. Шевелев.:Автореф. дис. ... канд. техн. наук. Томск, -2006.- 18 с.

9. Кукушкина О. В. Определение авторства текста с внедрением буквенной и грамматической информации / О.В. Кукушкина, А.А. Поликарпов, Д.В. Хмелев. Проблемы передачи информации.М .: Наука, 2001. Т. 37, № 2. С. 96-108.

10. Мартыненко Г. Я. Основы стилеметриы / Г.Я Мартыненко.Л:ЛГУ,- 1988. -170 с.

11. Рогов А.А. Автоматизированная система обработки и анализа литературных текстов СМАЛТ/. А.А Рогов, Ю.В Сидоров, А.В. Король.Труды и материалы II Междунар. конгресса исследователей русского языка «Русский язык: исторические судьбы и современность». М: МГУ, 2004. С. 485-486.

12. Система антиплагіат [Електронний ресурс]:–Режим доступу: https://nsu.antiplagiat.ru – Назва з екрану.

13. Система авторовед [Електронний ресурс]: – Режим доступу: http://www.swsys.ru/index.php?page=article&id=3703 – Назва з екрану.

14. Шевелев А. Г. Методы автоматической класификации текстов на естественном язике: / А.Г Шевелев.Учеб. помощь. Томского: ТМЛ-Пресс, 2007. 144 с.

15. Романов А. С.Идентификация автора текста с помощью аппарата опорных векторов / А.С Романов, Р.В.Мещеряков .Компьютерная лингвистика и интеллектуальные технологии: по мате-риалам ежегодной Международной конференции «Диалог-2009». М.: РГГУ, 2009. Вып. 8, №15. С. 432–437.

16. Бісікало О. В. Метод визначення ключових слів англомовного тексту на основі DKPRO CORE / Бісікало О.В, Яхимович О.В. – "Технологічний аудит та резерви виробництва". – Том 1, № 2(21). – 2015. – С. 12 – 14.

17. Stamatatos E. Computer-Based Authorship Attribution without Lexical Measures / Stamatatos E, Fakotakis N, Kokkinakis N., 2001. – 234 с.

18. About WordNet [Електронний ресурс]: – Режим доступу: http://stevenloria.com/tutorial-wordnet-textblob. – Назва з екрану.

19. Tutorial: What is WordNet? [Електронний ресурс]: – Режим доступу: https://wordnet.princeton.edu. – Назва з екрану.

20. Метод групового врахування аргументів [Електронний ресурс] – Режим доступу до ресурсу: https://studfile.net/preview/4494701/page:8/.

21. Осовский С. Нейронные сети для обработки информации / Пер. с польск. – М.: Финансы и статистика, 2002. – 344 с.

21. Хайкин С. Нейронные сети: полный курс, 2-е изд.: Пер. с англ. – М.:”Вильямс”, 2006.

22. Комп’ютерне моделювання систем та процесів. Методи обчислень. Частина 1 : навчальний посібник / Квєтний Р. Н., Богач І. В., Бойко О. Р., Софина О. Ю., Шушура О.М.; за заг. ред. Р.Н. Квєтного. – Вінниця: ВНТУ, 2012. – 193 с.

23. Робот и искусственный интеллект [Електронний ресурс] – Режим доступу: http://robot.paccbet.ru/docs/robot_ai.php

24. Tom M. Mitchell, Machine Learning, – McGraw-Hill Science, 2007 – 422 s: ISBN 0070428077

25. Wendell Wallach, Teaching Robots Right from Wrong, – New York: Oxford, 2009, – 288s: ISBN 978-0-19-537404-9

26. Christopher M. Bishop, Pattern Recognition and Machine Learning, – NY: Springer Science+Business Media, 2006 – 758 s: ISBN-10: 0-387-31073-8

27. Работа будущего – обучение роботов взаимодействию с людьми [Електронний ресурс] – Режим доступу: https://geektimes.ru/company/robohunter/blog/252576/

28. Как создать искусственный интеллект? История вторая. Алгоритмы

интеллектуального поиска и хранения информации [Електронний ресурс] – Режим доступу: https://habrahabr.ru/post/258399/

29. Ограничения формального обучения, или Почему роботы не умеют танцеват [Електронний ресурс] – Режим доступу: https://geektimes.ru/post/280662/

30. Осваиваем техническое зрение на примере Bioloid SYTEM и HaViMo2.0 [Електронний ресурс] – Режим доступу: https://habrahabr.ru/company/robotgeeks/blog/251781/

31. Методика синтеза стратегии движения автономного мобильного робота на основе эволюционных процес сов [Електронний ресурс] – Режим доступу:http://cyberleninka.ru/article/n/metodika-sinteza-strategii-dvizheniya-avtonomnogo-mobilnogo-robota-na-osnove-evolyutsionnyh-protsessov

32. Удивительно, но метод проб и ошибок уже актуален и для роботов сов [Електронний ресурс] – Режим доступу: http://robotforum.ru/novosti-texnogologij/udivitelno-no-metod-prob-i-oshibok-uzhe-aktualen-i-dlya-robotov-video.html

33. Робототехники изучают новые методы обучения роботов [Електронний ресурс] – Режим доступу: http://www.engnews.ru/news/robototehniki_izuchayut_novye_metody_obucheniya_robotov.html

34. M. Dorigo, Robot Shaping: Developing Situated Agents through Learning, International Computer Science Institute, – Berkely: Technical Report, 2007. – 626 s. – ISBN 685-83-4595-309-8

35. Leslie Kaelbling, An Adaptable Mobile Robot, in F. Varela and P. Bourgine (eds) , Toward a Practice of Autonomous Systems, –L: MIT Press, 2012 – 156 s. – ISBN 978-83-7338-965-8.

36. Соколов Д.О., Применение двухэтапного генетического программирования для построения автомата, управляющего моделью танка в

игре «Robocode». Бакалаврская работа. – СПбГУ ИТМО, 2009 – 152s

37. Mitchell M., An Introduction to Genetic Algorithms. – L: MIT Press, 2006 – 352s ISBN 978-83-7338-309-8.

38. Хайкин С. Нейронные сети. Полный курс. – М.: Вильямс, 2006– 352s ISBN 978-83-8765-523-8.

39. Розенфельд А. Распознавание и обработка изображений с помощью вычислительных машин: пер. с англ. под ред. Д.С.Лебедева / Розенфельд А. – М.: Мир, 2012 – 352 s ISBN 978-83-7338-309-8..

40. Филиппов С.А. LEGO роботы в обучении мехатронике и автоматизации. - СПб.: Изд-во «Политехника-сервис», 2010– 352 s ISBN 978-83-7338-309-8.

41. Хюккель М. Оператор нахождения контуров на кодированных изображениях –. М.: Мир, 2013 – 352 s ISBN 978-83-7338-309-8..

42. Яковлев О.Б. Тепловидение и его применение в системах охраны и обеспечения безопасности – Алгоритм безопасности, 2008.- – 352 s ISBN 978-83-7643-309-8.

43. Обучение нейронной сети [Електронний ресурс] – Режим доступу: http://neuronus.com/nn/38-theory/240-algoritmy-obucheniya-iskusstvennykh-nejronnykh-setej.html

44. Промышленный робот учится методом проб и ошибок [Електронний ресурс] – Режим доступу: https://geektimes.ru/post/272926/

45. A Browne, Neural Network Perspectives on Cognition and Adaptive Robotics, – CRC Press, 2007 – 370 s ISBN-10 0750304553

46. Программируем роботов — бесплатный робосимулятор V-REP. Первые шаги [Електронний ресурс] – Режим доступу: https://habrahabr.ru/company/makeitlab/blog/253357/

47. V-REP User Manual [Електронний ресурс] – Режим доступу:

http://www.coppeliarobotics.com/helpFiles/index.html

48. V-REP [Електронний ресурс] – Режим доступу: http://www.coppeliarobotics.com/

49. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. – 2-е изд., стереотип. – М.: Горячая линия-Телеком, 2002. – 382 с.

50. Ясницкий Л.Н. Введение в искусственный интеллект. – М.: ИЦ «Академия», 2005. – 176 с.

51. Шалыто А.А. Методы аппаратной и программной реализации алгоритмов. – СПб.: Наука, 2005. – 780 с.

52. Хайкин С. Нейронные сети: полный курс, 2-е изд.: Пер. с англ. – М.:”Вильямс”, 2006.

53. Осовский С. Нейронные сети для обработки информации / Пер. с польск. – М.: Финансы и статистика, 2002. – 344 с.

54. Рутковская Д., Пилиньский Л., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер. с польск. – М.: Горячая линия-Телеком, 2007. – 452 с.

55. Ткаченко Р.О. Моделювання методами нейронних мереж: навч.-метод. посібник / Р.О. Ткаченко, П.Р. Ткаченко, Н.О. Мельник; ЛІБС УБС НБУ. – Львів, 2010. – 114 с.

56. Ткаченко Р.О., Юрчак І.Ю., Цимбал Ю.В. Неітераційне навчання нейронних мереж прямого поширення // Вісник Держ. ун-ту “Львівська політехніка”: Комп’ютерна інженерія та інформаційні технології. – – 1999. – № 380. – С. 109–115.

57. Цмоць І.Г. Інформаційні технології та спеціалізовані засоби обробки сигналів і зображень у реальному часі. – Львів: УАД, 2005. – 227с.

58. Адаптивні методи та їх використання у робототехніці [Електронний ресурс] – Режим доступу:

http://www.nbuv.gov.ua/old_jrn/natural/Tp/2011_1/Stat2.htm

59. Методы робастного, нейро-нечеткого и адаптивного управления / Под ред. Н.Д. Егупова. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. – 744 с.

60. Пью А. Техническое зрение роботов // М.: Машиностроение, 2007. — 320 с.

61. Paul Martin and Ulrich Nehmzow, Programming g by Teaching: Neural Network Control in the Manchester Mobile Robot, Intern. Conf. Intelligent Autonomous Vehicless, – Helsinki, June 2005 – 982 s ISBN 978-83-7338-309-8.

62. M.E. Pollack, “Intelligent Technology for an Aging Population: The Use of AI to Assist Elders with Cognitive Impairment,” – AI Magazine, 2005– 635 s ISBN 978-83-7338-309-8.

63. Wasserman, Stanley, and Katherine Faust. Social Network Analysis: Methods and Applications. – Cambridge University Press, 1994 – 552s ISBN 978-83-7338-309-8.

64. Гренандер У. Лекции по теории образов: пер. с англ. под ред. Ю.Журавлёва / Гренандер У – . М.: Мир, 2009– 302 s ISBN 325-83-3657-309-8..

65. Кислицкий М.И. Планетоходы, космическая робототехника и наземные роверы, – СПб.: Ленэкспо, 2010. – 968 s ISBN 978-83-6875-309-8.

66. Кощавцев Н.Ф. Состояние и перспективы развития техники ночного – Прикладная физика, 2009 – 458 s ISBN 345-83-356-309-8..

67. Система навигации мобильного робота [Електронний ресурс] – Режим доступу: http://www.bestreferat.ru/referat-216810.html

68. Техническое зрение роботов [Електронний ресурс] – Режим доступу: http://ua.bankreferatov.ru/referats/6E01DAE0CAD74339C32568B600285359/vision.DOC.html&Key=940337

69. Глазами робота: что такое «машинное зрение» [Електронний ресурс] – Режим доступу: http://www.popmech.ru/technologies/238704-glazami-robota-chto-

takoe-mashinnoe-zrenie/

70. Применение нейронных сетей в робототехнике: перспективы и преимущества [Електронний ресурс] – Режим доступу: https://geektimes.ru/post/239543/

71. Нейронная сеть - обучение ИНС с помощью алгоритма обратного распространения [Електронний ресурс] – Режим доступу: http://robocraft.ru/blog/algorithm/560.html

72. Steven Bird, Ewan Klein, Edward Loper. Introduction to Natural Language Processing. – 2001-2007. – University of Pennsylvania.

ДОДАТКИ

Додаток А (обов’язковий)

Технічне завдання

ЗАТВЕРДЖЕНО

Зав. кафедри АІІТ

_____ Квєтний Р. Н.

«__»_____2019 р

ТЕХНІЧНЕ ЗАВДАННЯ

на магістерську кваліфікаційну роботу «Застосування методів машинного навчання для визначення авторства україномовного тексту»

08-02.МКР.013.00.000 ТЗ

Керівник роботи:

д.т.н., проф., зав. каф. АІІТ Квєтний Роман Наумович.

«___» _____ 2019 р.

Виконавець:

ст. гр. 1АКІТ-18м Стовбчатий Максим Михайлович

«___» _____ 2019 р.

Вінниця 2019

1. Назва та галузь застосування

Система для атрибуції автора україномовного тексту

2. Підстава для розробки

Розробку системи здійснювати на підставі наказу по університету № 254 від 02.10.2019 р. та завдання до магістерської кваліфікаційної роботи складеного та затвердженого кафедрою «Автоматизації та інтелектуальних інформаційних технологій.

3. Мета та призначення розробки

Мета роботи полягає в підвищенні якості визначення авторства україномовного тексту на основі методів і моделей комп'ютерної лінгвістики та машинного навчання, а також доступних програмних бібліотек і технологічних засобів.

4. Джерела розробки

1. Стовбчатий М.М Застосування методів машинного навчання для визначення авторства україномовного тексту / М.М. Стовбчатий.Матеріали доповідей XLVIII науково-технічної конференції підрозділів. Вінницького національного технічного університету, 22–23 березня. – Вінниця : ВНТУ, 2019. Режим доступу до ресурсу: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2019.

2. Метод групового врахування аргументів [Електронний ресурс] – Режим доступу до ресурсу: https://studfile.net/preview/4494701/page:8/.

3. Лінгвістичний портал Mova.info / Морфний сегментатор українського тексту. [Електронний ресурс] – Режим доступу до ресурсу: http://www.mova.info/Page2.aspx?l1=101. – Назва з екрану

5. Показники призначення

Система для атрибуції автора україномовного тексту має працювати на основі даних синтаксичного аналізу речень лабораторії комп'ютерної лінгвістики Київського Національного Університету імені Тараса Шевченка.

Вихідні дані для роботи програми:

- Текст у вигляді множини чисельних показників синтаксичного аналізу речень, з яких він складається.

Результати роботи програми:

- результати моделювання показають загальну достовірність визначення авторства 95%
- результати метода МГВА визначення автора «Драч Іван» показують достовірність 97%
- результати метода МГВА визначення автора «Мойсієнко» показують достовірність 90%
- результати метода МГВА визначення автора «Вінграновський» показують достовірність 95%

6. Економічні показники

- прогнозовані витрати на розробку –73 000 грн.;
- абсолютна ефективність розробки – 264 000грн.;
- термін окупності витрат для виробника – до 3-х років.

7. Стадії розробки

1. Розділ 1 «Аналіз методів та засобів визначення авторства тексту» має бути виконаний до 15.10.2019.

2. Розділ 2 «Огляд технологій та засобів машинного навчання для обробки природної мови» має бути виконаний до 25.10.2019.

3. Розділ 3 «Застосування методів машинного навчання та методу МГУА для вирішення задачі дослідження» має бути виконаний до 29.11.2019.

5. Економічний розділ має бути виконаний до 10.12.2019.

8. Порядок контролю та приймання

1. Рубіжний контроль. Провести до 15.11.2019.

2. Попередній захист магістерської кваліфікаційної роботи. Провести до 04.12.2019.

3. Захист магістерської кваліфікаційної роботи. Провести в період з 18.12.2019 до 19.12.2019.

Додаток Б (обов'язковий)

Графічна частина

Зав. Кафедри АІІТ _____ д-р техн. наук, професор
Квєтний Р. Н.
(підпис) (наукова ступінь, вчене звання, ініціали та прізвище)

Науковий керівник _____ д-р техн. наук, професор
Квєтний Р. Н.
(підпис) (наукова ступінь, вчене звання, ініціали та прізвище)

Тех. Контроль _____ .....
.....
(підпис) (наукова ступінь, вчене звання, ініціали та прізвище)

Нормоконтроль _____ .....
.....
(підпис) (наукова ступінь, вчене звання, ініціали та прізвище)

Рецензент _____ _____
_____
(підпис) (наукова ступінь, вчене звання, ініціали та прізвище)

Студент гр. 1АКІТ-18м _____ Стовбчатий М.М
(підпис) (ініціали та прізвище)

Продовження додатка Б

(A)	кількість вузлів	кількість простих	кількість рівнів	ширина гілкування	зміни шляху	дуга графа	вузли графа
Drach	6	1	4	2	1	1	0
Drach	17	2	6	2	2	10	3
Drach	10	1	4	4	1	3	1
Drach	15	1	10	2	1	3	2
Drach	17	3	5	5	2	6	0
Drach	10	1	5	3	1	2	1
Drach	6	2	3	3	1	3	1
Drach	20	2	6	2	2	10	3
Drach	10	1	4	4	1	3	1
Drach	15	1	10	2	1	3	2
Drach	9	3	5	5	2	6	0
Drach	17	1	5	3	1	2	1
Drach	6	2	3	3	1	3	1

Рисунок Б.1 – вхідні данні «Драч Іван»

Продовження додатка Б

(A)	кількість вузлів	кількість простих	кількість рівнів	ширина гілкування	зміни шляху	дуга графа	вузлів графа
Moisienko	18	2	5	5	1	6	1
Moisienko	6	1	3	3	0	2	0
Moisienko	11	1	5	4	1	3	0.36
Moisienko	12	1	4	5	1	13	0
Moisienko	10	1	5	3	2	3	1
Moisienko	16	2	5	4	1	3	1
Moisienko	9	1	4	4	1	3	1
Moisienko	10	2	5	4	1	3	1
Moisienko	9	2	4	3	1	3	1
Moisienko	15	1	7	2	3	7	2
Moisienko	6	1	3	3	0	2	0
Moisienko	11	1	5	4	1	3	0.36
Moisienko	17	1	4	5	1	13	0
Moisienko	12	1	5	3	2	3	1
Moisienko	10	2	5	4	1	3	1

Рисунок Б.2 – вхідні данні «Мойсеєнко»

Продовження додатка Б

Vingranovsky	15	1	7	3	2	11	3
Vingranovsky	15	1	4	7	1	11	0
Vingranovsky	13	1	4	6	1	6	0
Vingranovsky	19	1	7	6	2	3	1
Vingranovsky	5	1	5	1	1	1	1
Vingranovsky	14	1	6	3	1	4	0
Vingranovsky	21	1	11	3	3	10	5
Vingranovsky	12	2	5	5	1	3	1
Vingranovsky	8	1	5	3	0	1	1
Vingranovsky	14	1	5	3	1	4	0.57
Vingranovsky	14	1	5	3	1	4	0
Vingranovsky	15	1	5	4	2	6	3
Vingranovsky	7	1	3	2	1	4	0
Vingranovsky	7	1	3	3	0	2	0
Vingranovsky	16	2	6	2	2	6	0
Vingranovsky	39	4	11	3	4	36	2
Vingranovsky	6	1	4	2	0	2	1
Vingranovsky	10	1	9	2	0	0	3

Рисунок Б.3 – параметри автора «Вінграновський»

Продовження додатку Б

```
C:\Users\fkca2\Desktop\nn_classifier\venv\Scripts\python.exe C:/Users/fkca2/Desktop/nn_classifier/nn_classifier.py
[[50  0  0]
 [ 0 50  0]
 [ 0  0 50]]
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00        50
         1.0       1.00      1.00      1.00        50
         2.0       1.00      1.00      1.00        50

    accuracy                           1.00       150
   macro avg       1.00      1.00      1.00       150
weighted avg       1.00      1.00      1.00       150

[[32  5  0]
 [ 0 37  0]
 [ 0  2 35]]
              precision    recall  f1-score   support

         0.0       1.00      0.86      0.93        37
         1.0       0.84      1.00      0.91        37
         2.0       1.00      0.95      0.97        37

    accuracy                           0.94       111
   macro avg       0.95      0.94      0.94       111
weighted avg       0.95      0.94      0.94       111


Process finished with exit code 0
```

Рисунок Б.4 – результати навчання Scikit-learn

Продовження додатку Б

Мойсієнко	Результат	Вінграновський	Результат	Драч	Результат
-0,838841777	0	-9,452837995	1	-4,63644683	1
5,43485192	1	-9,215371344	1	-6,535968421	1
9,490017091	1	-8,965253295	1	-10,21520101	1
9,679178561	1	-7,173696854	1	-9,081437844	1
9,674322794	1	-7,465227715	1	-9,081437844	1
10,92877734	1	-9,912205063	1	-6,132787341	1
5,43485192	1	-9,215371344	1	-6,535968421	1
-0,838841777	0	-9,452837995	1	-4,63644683	1
5,43485192	1	-9,215371344	1	-6,535968421	1
9,490017091	1	-8,965253295	1	-10,21520101	1
-9,585205982	1	7,504215039	1	-9,77265416	1
-3,732990179	1	8,566666043	1	-5,916235136	1
2,602581617	0	6,380756829	1	-3,462892186	1
-7,71584839	1	4,776869905	1	-7,263811705	1
-7,320355897	1	9,86998084	1	-9,02613012	1
-5,362515209	1	-8,244848369	0	-7,092175545	1
-5,592998979	1	8,144605935	1	-7,092175545	1
-8,537733392	1	8,820832673	1	-7,48792009	1
-9,559387173	1	-8,82085707	0	-6,814692161	1
-6,291477883	1	9,770904524	1	-9,928450064	1
-9,506860578	1	-8,51349922	1	10,20464318	1
-8,337961264	1	-8,815593061	1	9,37328861	1
-9,816400328	1	-8,593943169	1	10,20725501	1
-9,816400328	1	-8,593943169	1	10,20725501	1
-9,602853	1	-9,773967609	1	10,20464318	1
-8,337961264	1	-8,815593061	1	9,37328861	1
-9,816400328	1	-8,593943169	1	10,20725501	1
0,333346283	1	-9,178381398	1	-5,787957436	0
-9,344067875	1	-9,181156558	1	9,886701053	1
-9,602853	1	-9,773967609	1	10,20464318	1
	90%		93%		97%

Рисунок Б.5 – Результат алгоритму МГОА

Продовження додатку Б

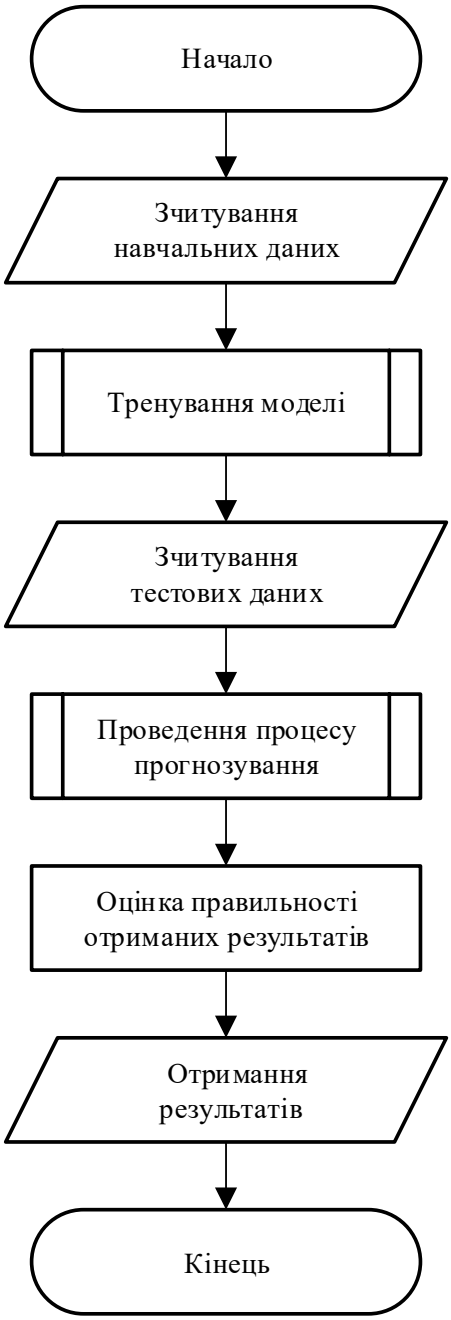

Рисунок Б.6 – Блок схема алгоритму з використання нейронної мережі

Продовження додатку Б

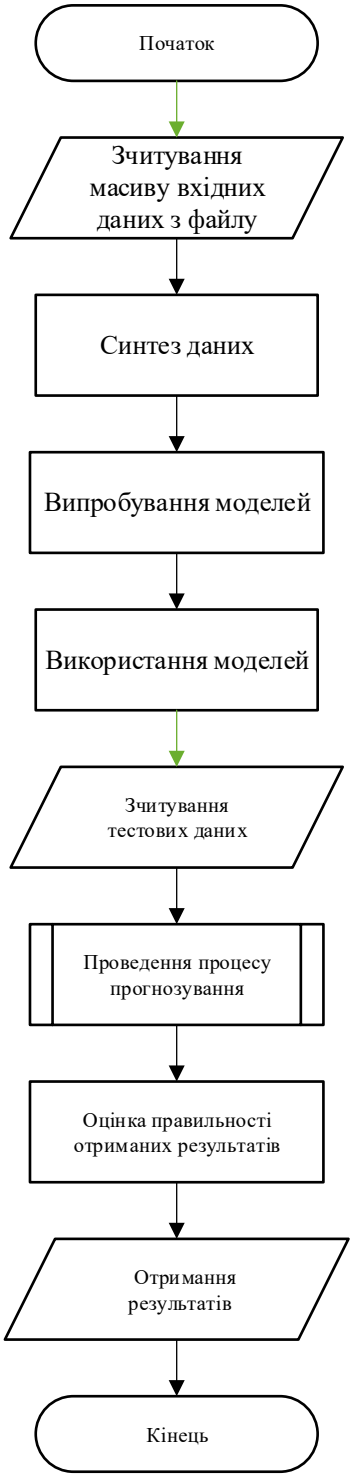

Рисунок Б.7 – Блок схема алгоритму з використанням алгоритму МГУА

Продовження додатку Б

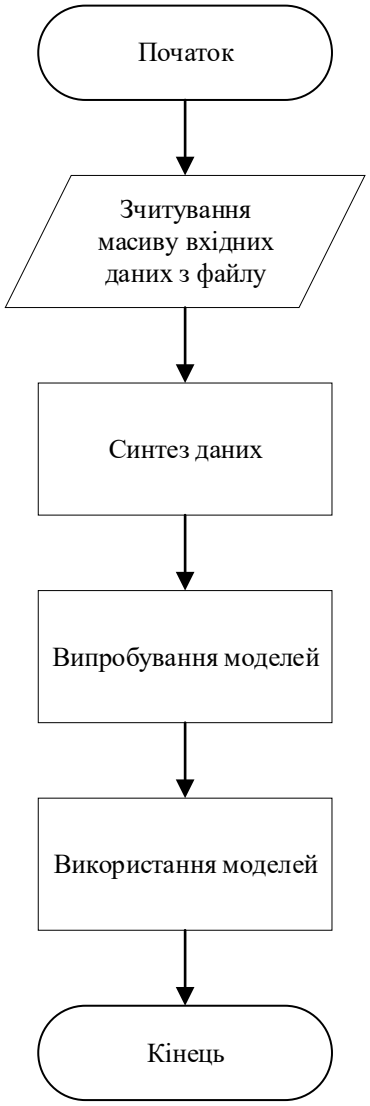

Рисунок Б.8 – Схема алгоритму МГОА

Додаток В – Код класифікатора

```
# Import required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import sklearn

# Import necessary modules
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from math import sqrt
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import r2_score,classification_report,confusion_matrix
from data_preprocessing import *

# Create multilayer perceptron.
mlp = MLPClassifier(hidden_layer_sizes=(150,100,50), max_iter=3000,activation =
'relu',solver='adam',random_state=1)

# This code fits the model to the training data
mlp.fit(rescaled_train_X,train_Y)

# Use above trained model to generate predictions on the training and test dataset,
respectively.
predict_train = mlp.predict(rescaled_train_X)
predict_test = mlp.predict(rescaled_test_X)

# Print the confusion matrix and the confusion report results on the training data
print(confusion_matrix(train_Y,predict_train))
print(classification_report(train_Y,predict_train))

# Print the confusion matrix and the confusion report results on the testing data
print(confusion_matrix(test_Y,predict_test))
print(classification_report(test_Y,predict_test))
```