Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

ДОСЛІДЖЕННЯ ТА ВИЯВЛЕННЯ НЕДОСТОВІРНИХ ВАКАНСІЙ ЗА ДОПОМОГОЮ МЕТОДІВ МАШИННОГО НАВЧАННЯ
(тема)

Виконав:
здобувач 2 року навчання,
групи ІНФм-24-1
Білоцерківська В.А.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Науковий керівник ст. викл. Кобилін І.О.
(посада, прізвище, ініціали)

Допускається до захисту

Завідувач кафедри інформатики _____ Кобилін О. А.
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту

Кафедра Інформатики

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Білоцерківській Вікторії Андріївні
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та виявлення недостовірних вакансій за допомогою методів машинного навчання

затверджена наказом університету від 14 листопада 2025 року № 1045Ст
2. Термін подання здобувачем роботи до екзаменаційної комісії 19 листопада 2025 р.
3. Вихідні дані до роботи методи обробки природної мови, методи класифікації текстів, літературні джерела щодо застосування алгоритмів машинного навчання, інструменти для попередньої обробки текстових даних, програмні засоби для реалізації моделей класифікації Python, Scikit-learn, NLTK, PyTorch, Transformers, програмні засоби для створення веб-застосунку FastAPI, React, TypeScript, набір даних "Fake Job Postings" з платформи Kaggle, допоміжні діаграми, графіки та статистичні матеріали, результати навчання та тестування моделей, синтетично згенеровані текстові вибірки для балансування класів.

4. Перелік питань, що потрібно опрацювати в роботі
1. Аналіз сучасних методів обробки природної мови.
2. Аналіз лінгвістичні особливості реальних і фейкових вакансій та визначити ключові ознаки, що впливають на результат класифікації.
3. Формування вибірки даних.
4. Реалізувати та порівняти різні моделі машинного навчання і вибрати найефективніший підхід для виявлення недостовірних вакансій.
5. Розробка програмного застосунку для автоматичного аналізу текстів вакансій та інтегрувати модель у клієнтсько-серверну архітектуру.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність проблеми виявлення недостовірних вакансій, об'єкт та мета дослідження, постановка задачі, блок-схема алгоритму аналізу тексту вакансії, діаграми попередньої обробки та векторизаці текстів, графічні ілюстрації статистики вибірки , приклад синтетично згенерованих текстів для балансування даних, схема архітектури веб-сервісу, інтерфейс головної сторінки застосунку з полем аналізу тексту, ілюстрація результатів класифікації з відсотковою оцінкою та підсвіченими підозрілими словами, сторінка статистики з графіками аналізу, приклад сторінки профілю користувача, підсумкові графіки порівняння ефективності моделей, висновки та перспективи подальшого розвитку системи.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	29.09.2025	
2	Аналіз завдання, підбір літератури	30.09.25-07.10.25	
3	Аналіз літератури з досліджуваної проблеми	08.10.25-14.10.25	
4	Особливості методів обробки природної мови	15.10.25-20.10.25	
5	Дослідження методів аналізу вакансій	21.10.25-27.10.25	
6	Програмна реалізація	28.10.25-05.11.25	
7	Обґрунтування отриманих результатів	06.11.25-11.11.25	
8	Оформлення пояснювальної записки	12.11.25-14.11.25	
9	Перевірка на нормоконтроль	19.11.25	
10	Перевірка на плагіат	21.11.25	
11	Рецензування	22.11.25	
12	Підготовка презентації та доповіді	23.11.25	
13	Занесення роботи в електронний архів	23.11.25	
14	Попередній захист кваліфікаційної роботи	01.12.25	

Дата видачі завдання 29 вересня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ ст. викл. Кобилін І. О.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи: 87 с., 6 табл., 11 рис., 28 джерел.

МАШИННЕ НАВЧАННЯ, ОБРОБКА ПРИРОДНОЇ МОВИ, ВЕБСЕРВІС, КЛАСИФІКАЦІЯ ТЕКСТІВ, БЛОК-СХЕМА АЛГОРИТМУ, NLP, PYTHON, REACT, TENSORFLOW, SCIKIT-LEARN, PANDAS.

Об’єктом дослідження є процес автоматизованого аналізу текстів вакансій з метою виявлення недостовірних або шахрайських оголошень.

Предметом дослідження є методи машинного навчання та обробки природної мови, що використовуються для класифікації вакансій за рівнем достовірності.

Метою дослідження є створення вебсервісу, який дозволяє автоматично оцінювати підозрілість вакансії за її текстом, використовуючи сучасні алгоритми машинного навчання.

Використано методи машинного навчання, методи обробки природної мови та статистичного аналізу даних.

Наукова новизна роботи полягає у створенні комплексного вебсервісу, який поєднує модель машинного навчання для аналізу вакансій у реальному часі.

Взаємозв’язок з іншими роботами полягає у використанні сучасних підходів з кібербезпеки та NLP для протидії шахрайству в цифровому середовищі.

Рекомендації щодо використання результатів передбачають інтеграцію системи у платформи пошуку роботи для автоматичної перевірки нових оголошень.

У результаті дослідження розроблено вебсервіс із бекендом на Python, що містить модель машинного навчання для аналізу вакансій, та фронтендом на React із зручним інтерфейсом користувача.

ABSTRACT

Explanatory note to the qualification work: 81 pages, 6 table, 11 figures, 28 sources.

MACHINE LEARNING, NATURAL LANGUAGE PROCESSING, WEBSERVICE, TEXT CLASSIFICATION, ALGORITHM BLOCK DIAGRAM, NLP, PYTHON, REACT, TENSORFLOW, SCIKIT-LEARN, PANDAS.

The object of the research is the process of automated analysis of job vacancy texts aimed at detecting unreliable or fraudulent postings.

The subject of the research is the machine learning and natural language processing methods used to classify vacancies by their level of authenticity.

The purpose of the research is to create a web service that automatically evaluates the suspiciousness of a vacancy based on its text, using modern machine learning algorithms.

Machine learning methods, natural language processing techniques, and statistical data analysis were applied.

The scientific novelty of the research lies in the development of a comprehensive web service that integrates a machine learning model for real-time vacancy analysis.

The relationship with other works is reflected in the use of modern approaches from cybersecurity and NLP to counter fraud in the digital environment.

The recommendations for the use of the results include integrating the system into job search platforms to automatically verify new postings.

As a result of the research, a web service was developed with a Python-based backend containing a machine learning model for vacancy analysis and a React-based frontend with a user-friendly interface.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ШІ – штучний інтелект

API – Application Programming Interface (програмний інтерфейс прикладного програмування)

CNN – Convolutional Neural Network

GPU – Graphics Processing Unit (графічний процесор)

GRU – Gated Recurrent Unit (вентильний механізм у рекурентних нейронних мережах)

GUI – Graphical User Interface (графічний інтерфейс користувача)

IDE – Integrated Development Environment (інтегроване середовище розробки)

LSTM – Long Short-Term Memory (довга короткострокова пам’ять)

NER – Named Entity Recognition (розпізнавання іменованих сутностей)

NLP – Natural Language Processing (обробка природної мови)

SVM – Support Vector Machine(метод опорних векторів)

ВСТУП

У сучасних умовах стрімкого розвитку цифрових технологій та глобалізації ринку праці значна частина процесів пошуку роботи та підбору персоналу відбувається онлайн. Сайти з працевлаштування, соціальні мережі та спеціалізовані вебплатформи стали основними інструментами для взаємодії між роботодавцями та здобувачами. Разом із позитивними тенденціями цифровізації виникла і нова проблема – поява великої кількості недостовірних або шахрайських вакансій, що становлять потенційну загрозу для користувачів.

Такі оголошення можуть використовуватися з метою збору персональних даних, фінансового шахрайства або поширення неправдивої інформації про умови працевлаштування. Саме тому виникає потреба у створенні інтелектуальних систем автоматичного аналізу вакансій, здатних виявляти потенційно небезпечні чи недостовірні пропозиції.

Актуальність роботи полягає у необхідності розробки ефективних програмних засобів, які б дозволяли автоматично аналізувати великі обсяги текстової інформації та своєчасно виявляти ознаки шахрайства у вакансіях. Використання методів машинного навчання для цієї задачі дає можливість суттєво підвищити точність аналізу, зменшити вплив людського фактора та оптимізувати процес модерації контенту на платформах з працевлаштування.

На сьогодні існують численні дослідження, спрямовані на виявлення фейкового контенту, однак більшість із них зосереджена на соціальних мережах, коментарях або новинах. Проблема недостовірних вакансій залишається менш вивченою. І хоча деякі компанії впроваджують автоматизовані фільтри для модерації контенту, більшість із них використовують прості евристичні правила, наприклад, перевірку ключових слів, які не здатні забезпечити високий рівень точності.

Тому виникає потреба у розробці системи, яка базується на глибокому аналізі тексту та навчанні на реальних даних, що дозволяє розпізнавати навіть приховані ознаки недостовірності. Використання алгоритмів машинного

навчання та нейронних мереж надає змогу виявляти закономірності, які неочевидні для людини, але мають статистичну значущість у контексті шахрайських вакансій.

У рамках даної роботи передбачається розробка вебсервісу, який об’єднує аналітичну частину модель машинного навчання з користувацьким інтерфейсом для взаємодії в реальному часі. Користувач може ввести текст вакансії, після чого система аналізує його та повертає оцінку достовірності у відсотках. Крім того, реалізовано модуль візуалізації статистики, який демонструє аналітичні графіки щодо розподілу підозрілих вакансій за часом або категоріями.

Для реалізації програмної частини використано мову програмування Python, бібліотеки Pandas, Scikit-learn, TensorFlow, PyTorch для побудови й навчання моделей, а також React для створення інтерактивного інтерфейсу користувача.

Серед методів, що застосовуються в роботі, – логістична регресія, метод опорних векторів (SVM), нейронні мережі та векторизація текстів за допомогою TF-IDF або Word2Vec.

Наукова цінність дослідження полягає у створенні комплексного підходу до виявлення недостовірних вакансій, який поєднує методи обробки текстів, машинного навчання та вебтехнологій. У результаті аналізу сучасних рішень встановлено, що існує потреба у створенні відкритих інструментів і моделей, які могли б використовуватися різними платформами з працевлаштування для автоматичного контролю достовірності контенту.

Отже, наукова задача, що розв’язується в роботі, полягає у створенні та дослідженні моделі машинного навчання для класифікації вакансій за ступенем достовірності та розробці вебсервісу для практичного застосування цієї моделі.

Реалізація поставленої задачі дозволить покращити якість інформаційних сервісів у сфері працевлаштування, підвищити безпеку користувачів і сприятиме розвитку систем штучного інтелекту, здатних аналізувати тексти в реальному часі.

1 ОГЛЯД ОСНОВНИХ МЕТОДІВ АНАЛІЗУ ТЕКСТОВИХ ДАНИХ

1.1 Огляд методів обробки природної мови

Обробка природної мови (NLP) – це галузь штучного інтелекту, що вивчає методи і алгоритми, за допомогою яких комп'ютери можуть аналізувати, розуміти та генерувати людську мову. У контексті дослідження недостовірних вакансій NLP є базовою складовою, оскільки саме завдяки їй можливо перетворити текст вакансії у структуру, придатну для машинного аналізу [1].

Основні завдання NLP включають:

– токенізацію – поділ тексту на окремі слова або фрази;

– лематизацію – зведення слова до початкової форми;

– видалення стоп-слів – виключення із тексту частих, але неінформативних слів («і», «в», «на», «що» тощо);

– аналіз частин мови (POS-tagging) – визначення граматичної ролі слова у реченні;

– виділення сутностей (NER) – розпізнавання назв організацій, імен, локацій тощо.

Одним із перших етапів NLP є попередня обробка тексту. Вона включає очищення даних від шуму – видалення спеціальних символів, HTML тегів, чисел, стоп-слів, а також приведення тексту до єдиного формату. Ці кроки є необхідними, щоб уникнути впливу несуттєвих елементів на модель. Після очищення застосовується токенізація – поділ тексту на окремі слова або фрази (токени). Далі проводиться лематизація або стемінг – зведення слів до їх початкової форми, що дозволяє моделі розглядати слова «працюю», «працював» і «працюватиме» як одне поняття. Такі процедури особливо важливі при роботі з українською або англійською мовами, які мають багату морфологію [2].

На наступному етапі важливу роль відіграє семантичний аналіз тексту, що дозволяє оцінити контекст і значення слів у реченні. Наприклад, система може розрізняти, коли слово «безкоштовно» використовується у позитивному

контексті «безкоштовне навчання під час роботи» або у підозрілому «безкоштовна вакансія без досвіду і договору». Для цього застосовуються моделі типу Word2Vec, GloVe або сучасні контекстні моделі на базі трансформерів (BERT, RoBERTa, DistilBERT). Вони дозволяють враховувати значення слова залежно від його оточення, що значно підвищує точність класифікації.

Ще одним важливим напрямом у NLP є аналіз тональності (Sentiment Analysis), який визначає емоційне забарвлення тексту. У фейкових вакансіях часто зустрічаються надмірно позитивні формулювання, перебільшення вигод або обіцянки швидкого прибутку. Виявлення таких мовних патернів допомагає ідентифікувати потенційно шахрайські оголошення.

Також значну роль відіграє розпізнавання сутностей (Named Entity Recognition, NER) – процес виявлення у тексті назв компаній, адрес, телефонів, імен чи сум грошей. Це дозволяє перевіряти, чи справді вакансія містить конкретну контактну інформацію, чи ж автор навмисно уникає деталей, щоб приховати особу. У реальних системах виявлення фейкових вакансій часто поєднують NER з аналізом шаблонів, щоб знайти невідповідності між зазначеними даними, наприклад, коли «роботодавець» не має жодних згадок у відкритих джерелах.

У сучасних підходах обробки природної мови активно використовуються нейронні мережі, зокрема рекурентні (RNN, LSTM, GRU) та трансформерні моделі. Вони здатні запам’ятовувати послідовності слів і враховувати контекст навіть у довгих текстах, що робить їх надзвичайно ефективними для класифікації вакансій за рівнем достовірності. Модель може навчатися розрізняти реальні описи посад від шахрайських, аналізуючи тисячі прикладів і виявляючи найменші відмінності у формулюваннях.

Використання NLP дозволяє також будувати статистичні моделі частоти вживання певних слів або словосполучень, що характерні для фейкових вакансій. Наприклад, слова «вклади», «швидкий заробіток», «без досвіду», «гарантований дохід» можуть вказувати на потенційну підозрілість. Такі лінгвістичні

особливості можна використовувати як додаткові ознаки під час навчання моделей машинного навчання.

Загалом, методи обробки природної мови створюють основу для побудови інтелектуальних систем, здатних не лише класифікувати текстові оголошення, а й пояснювати, які саме мовні конструкції чи слова вплинули на оцінку достовірності. Це робить аналіз прозорішим та більш зрозумілим для користувача.

У контексті вебсервісу для виявлення фейкових вакансій, NLP виступає ключовим компонентом, який перетворює звичайний текст у набір інформативних показників, що дозволяють штучному інтелекту робити точні висновки про його надійність.

У сучасних системах для реалізації NLP-процесів широко використовуються бібліотеки NLTK, spaCy, Stanza, transformers. Вони дозволяють швидко підготувати текст до аналізу, провести морфологічний та синтаксичний розбір, а також виконати семантичне моделювання.

Для задачі виявлення недостовірних вакансій такі методи особливо корисні, оскільки шахрайські тексти часто мають спільні мовні ознаки – повторювані шаблони, неправдиві обіцянки, емоційно забарвлену лексику «високі доходи без досвіду», «легка робота з дому», «виплати щодня». NLP дозволяє виділити ці патерни та перетворити їх на ознаки для подальшої класифікації.

Крім цього, NLP використовується для аналізу тональності, що дає змогу визначати емоційне забарвлення тексту. Наприклад, позитивна тональність із надмірним ентузіазмом може бути характерною для фейкових вакансій [3].

Таким чином, NLP є невід’ємною основою для побудови систем автоматичного розпізнавання текстових шахрайських оголошень, оскільки забезпечує структуризацію вхідних даних і виділення значущих характеристик.

1.2 Порівняння методів векторизації текстів

1.2.1 Метод Bag-of-Words

Один з найпростіших і найстаріших методів – метод Bag of Words (BoW). Його суть полягає у представленні кожного документа у вигляді вектора, де кожен елемент відповідає кількості поява певного слова у тексті. Таким чином, формується матриця розміром $m \times n$, де m – кількість документів, а n – кількість унікальних слів у корпусі.

Наприклад, якщо словник містить 10000 унікальних слів, то кожен текст буде представлений у вигляді вектора з 10000 компонентами, де більшість значень дорівнюватиме нулю. Основною перевагою BoW є простота реалізації та інтерпретації.

Проте цей метод має низку недоліків:

– втрачається порядок слів і контекст;

– збільшується розмірність простору;

– погано узагальнює значення нових слів.

Незважаючи на ці обмеження, BoW часто використовується як базовий метод для попередніх експериментів або для невеликих наборів даних.

1.2.2 Метод TF-IDF

Метод TF-IDF є вдосконаленням підходу BoW. Його головна ідея полягає у тому, щоб оцінювати не просто кількість поява слова, а його інформаційну значущість. Для цього враховується два показники:

– TF (Term Frequency) – частота появи слова в документі;

– IDF (Inverse Document Frequency) – зворотна частота появи слова в усіх документах корпусу.

Таким чином, ваги рідкісних, але важливих слів збільшуються, а поширені слова, на кшталт «і», «в», «з», мають меншу вагу.

Формула обчислення ваги має вигляд:

$$TF - IDF = TF \log \frac{N}{DF}, \quad (1.1)$$

де N – кількість документів;

DF – кількість документів, у яких зустрічається дане слово.

TF-IDF добре підходить для аналізу коротких текстів, таких як описи вакансій. Він дозволяє виділити ключові слова, що можуть свідчити про фейковість оголошення – наприклад, надмірне використання слів «високий дохід», «без досвіду», «миттєвий заробіток» тощо. Серед недоліків – відсутність урахування контексту та зв’язків між словами, проте на практиці TF-IDF залишається дуже ефективним для класичних моделей машинного навчання (SVM, Logistic Regression, Naive Bayes).

1.2.3 Метод Word2Vec та GloVe

Подальшим кроком розвитку методів векторизації став Word2Vec, розроблений дослідницькою групою Google у 2013 році. Word2Vec ґрунтується на ідеї, що слова, які зустрічаються в подібних контекстах, мають схоже значення. Модель використовує нейронну мережу для навчання векторних представлень слів у багатовимірному просторі, зазвичай 100–300 вимірів.

Існують дві архітектури Word2Vec:

– CBOW (Continuous Bag of Words) – передбачає слово за його контекстом;

– Skip-gram – навпаки, передбачає контекст за поточним словом.

Після навчання модель здатна вловлювати семантичні та синтаксичні зв’язки, наприклад: вчитель – школа + лікар ≈ лікарня.

Для завдання виявлення фейкових вакансій Word2Vec дозволяє враховувати смислову близькість слів, що важливо для розуміння прихованих

патернів у текстах. Наприклад, слова «безкоштовно» і «даром» будуть мати схожі вектори, навіть якщо зустрічаються в різних оголошеннях [3].

Метод GloVe, розроблений у Стенфордському університеті, поєднує підходи статистичних методів, як TF-IDF та нейронних моделей як Word2Vec. Він використовує матрицю спільної появи слів, що дозволяє моделі враховувати глобальну інформацію про взаємозв'язки між словами у всьому корпусі текстів.

GloVe намагається навчити такі векторні представлення, щоб відношення між векторами відповідало відношенню між словами у реальній мові. Наприклад:

$$\upsilon(\text{«король»}) - \upsilon(\text{«чоловік»}) + \upsilon(\text{«жінка»}) \approx \upsilon(\text{«королева»}). \qquad (1.2)$$

де υ – векторне представлення відповідного слова у багатовимірному просторі.

Перевага GloVe полягає у тому, що він краще відображає семантичну структуру мови, ніж Word2Vec, особливо на великих корпусах даних. Це робить його ефективним при роботі з великими наборами текстів вакансій.

1.2.4 Контекстні моделі

Останнім етапом розвитку є поява контекстних моделей, серед яких найвідоміша – BERT (Bidirectional Encoder Representations from Transformers), розроблена компанією Google у 2018 році.

На відміну від попередніх методів, де кожне слово має фіксований вектор, BERT створює контекстно-залежні представлення, тобто одне й те саме слово може мати різні значення залежно від контексту.

Наприклад, у реченнях: «Вакансія безкоштовна для кандидатів» та «Курс навчання безкоштовний» BERT розуміє, що обидва слова мають подібний зміст, але використовуються в різних контекстах. Це дозволяє досягати високої

точності при класифікації текстів, виявленні фейкових вакансій та аналізі їхніх лінгвістичних особливостей [5].

RoBERTa – це проста, але дуже популярна альтернатива/наступник BERT. Вона покращує BERT за рахунок ретельної та розумної оптимізації повчальних гіперпараметрів для BERT. Декілька простих і зрозумілих змін у сукупності підвищують продуктивність RoBERTa і дозволяють їй перевершити BERT практично у всіх завданнях, для яких він був розроблений.

Найцікавіше, що під час публікації Роберти інший популярний новий трансформер, XLNet, також був представлений у дослідницькій роботі. Однак зміни, внесені до XLNet, реалізувати значно складніше, ніж у RoBERTa, і це лише збільшує популярність останньої серед спільноти AI/NLP.

RoBERTa використовує ту ж архітектуру, що й BERT. Однак, на відміну від BERT, під час навчання вона навчається тільки генерації пропущеного токена (BERT також передбачався передбачення наступної пропозиції).

RoBERTa досягла продуктивності завдяки таким змінам:

– більш тривалий час навчання та більший обсяг навчальних даних (у 10 разів більше від 16GB до 160GB);

– розмір батчу від 256 до 8000 і більший словник – від 30k до 50k;

– як вхідні дані використовуються більш довгі послідовності, але RoBERTa як і раніше має обмеження на максимальну кількість токенів – 512, як і у BERT;

– динамічне маскування дозволяє схемі маскування змінюватися при кожній подачі послідовності на модель. Відмінність від BERT у тому, що скрізь використовується та сама маскуюча схема.

1.3 Аналіз сучасних підходів до виявлення недостовірного контенту та вакансій

Сучасні дослідження у сфері виявлення фейкової інформації свідчать про активний розвиток підходів, що поєднують обробку тексту, машинне навчання

та великі мовні моделі (LLM). У наукових працях останніх років активно вивчаються методи виявлення фейкових новин, спам – повідомлень, шахрайських оголошень у соціальних мережах. Більшість цих рішень базується на попередньо навчених моделях, таких як BERT, DistilBERT, GPT, які є задовільними для конкретних завдань.

Щодо вакансій, дослідження показують, що фейкові оголошення мають спільні характеристики:

– використання привабливих, але нечітких описів;

– відсутність конкретики про роботодавця;

– аномально високі обіцянки щодо заробітку;

– заклики до швидкої дії або негайного зв’язку.

Сучасні системи детекції вакансій використовують поєднання лексичних, семантичних і поведінкових ознак. Наприклад, можна враховувати не лише текст, а й метадані – час публікації, джерело, кількість схожих оголошень.

Важливо, що в останні роки увага дослідників спрямована на пояснювані моделі (Explainable AI), які не лише класифікують вакансію, а й пояснюють, які саме слова чи фрази вплинули на рішення моделі. Це підвищує довіру користувачів і дозволяє вдосконалювати моделі в процесі використання.

Таким чином, сучасний підхід до виявлення недостовірних вакансій – це синтез кількох технологій: обробки природної мови, семантичного аналізу, машинного навчання та візуальної аналітики результатів. Саме такий комплексний підхід реалізовано у даній кваліфікаційній роботі.

Сучасні дослідження у сфері виявлення фейкового контенту демонструють активний розвиток методів, які поєднують машинне навчання, обробку природної мови (NLP) та глибинні нейронні мережі. Особлива увага приділяється не лише загальному аналізу фейкових новин чи повідомлень у соціальних мережах, але й більш спеціалізованим завданням, зокрема виявленню недостовірних вакансій**.** Це обумовлено зростанням кількості шахрайських оголошень, спрямованих на отримання персональних даних або введення користувачів в оману.

1.3.1 Використання класичних методів машинного навчання

На початкових етапах розвитку цього напряму дослідники активно використовували класичні методи машинного навчання: логістичну регресію, SVM (Support Vector Machine), наївний баєсівський класифікатор та дерева рішень. Такі підходи показали хороші результати при роботі з векторизованими текстовими представленнями, наприклад, TF-IDF або Bag of Words. Для виявлення фейкових вакансій ці методи дозволяють знаходити статистичні закономірності у вживанні певних слів чи фраз, притаманних шахрайським оголошенням – наприклад, «високий заробіток без досвіду», «потрібен лише паспорт», «миттєва оплата» тощо [6].

Попри активний розвиток глибинного навчання, класичні методи машинного навчання й досі залишаються актуальними, особливо у тих випадках, коли обсяг даних є обмеженим або потрібно отримати інтерпретовані результати. У контексті виявлення фейкових вакансій такі методи демонструють досить високу ефективність, якщо попередньо провести якісну підготовку даних та векторизацію текстів.

Класичні алгоритми, такі як логістична регресія, наївний баєсівський класифікатор, дерева рішень або метод опорних векторів, дозволяють навчати моделі на основі кількісних характеристик тексту. Наприклад, модель може аналізувати частоту вживання певних слів, середню довжину речень, співвідношення іменників і дієслів або наявність типових шаблонів фейкових оголошень. Важливо, що такі алгоритми забезпечують не лише класифікацію, але й надають можливість інтерпретувати, які саме ознаки вплинули на рішення. Це має значення для пояснюваності результатів і довіри користувачів до системи.

Логістична регресія часто використовується як базова модель, оскільки вона проста у реалізації та добре підходить для задач бінарної класифікації, коли необхідно визначити, чи є вакансія достовірною або підозрілою. Вона надає

можливість оцінювати ваги ознак, завдяки чому можна побачити, які слова або фрази найсильніше впливають на визначення фейковості. Наприклад, надмірна кількість емоційних прикметників або часте використання фінансових обіцянок без конкретики може суттєво збільшувати ймовірність того, що оголошення є шахрайським.

Метод опорних векторів зарекомендував себе як один із найефективніших у випадках, коли дані мають велику кількість ознак. Завдяки використанню ядерних функцій, SVM здатен знаходити оптимальні гіперплощини, що розділяють класи навіть у високовимірних просторах. Для задачі виявлення фейкових вакансій це означає можливість точного відокремлення справжніх вакансій від тих, що мають ознаки недостовірності, навіть якщо їхні тексти схожі на перший погляд. Крім того, SVM є стійким до надлишкових ознак, що часто спостерігається у текстових даних після векторизації.

Наївний баєсівський класифікатор, у свою чергу, є одним із найпопулярніших алгоритмів для роботи з текстом, зокрема в задачах спам фільтрації, визначення тональності повідомлень або класифікації новин [7]. Його ключова перевага полягає у швидкості навчання та простоті інтерпретації. Для задачі аналізу вакансій він дозволяє швидко виявляти закономірності між певними словами чи фразами та класом вакансії. Хоча цей метод ґрунтується на спрощеному припущенні незалежності ознак, на практиці він демонструє досить добрі результати на текстових наборах даних.

Алгоритми на основі дерев рішень також знаходять своє застосування в цій сфері. Вони дозволяють створювати ієрархічні моделі, де кожне рішення приймається на основі певної умови – наприклад, чи містить текст слово «гарантовано», або чи зазначена конкретна адреса компанії. На основі таких дерев можуть будуватися ансамблеві методи, як-от Random Forest чи Gradient Boosting, які поєднують велику кількість дерев, щоб підвищити стабільність і точність прогнозу.

У більшості досліджень, що стосуються виявлення фейкових вакансій, класичні методи виступають відправною точкою для подальшого вдосконалення

моделі. Вони дозволяють швидко перевірити гіпотези, визначити інформативні ознаки та оцінити базовий рівень точності системи. Наприклад, комбінація TF-IDF з логістичною регресією або SVM часто використовується як еталон, з яким порівнюють результати нейронних мереж.

Крім того, класичні моделі мають одну важливу перевагу – стійкість до перенавчання у випадках, коли кількість фейкових вакансій у наборі даних є невеликою. У таких умовах складні нейронні мережі можуть надто точно запам’ятовувати приклади, втрачаючи здатність до узагальнення. Натомість простіші моделі з регуляризацією, наприклад, логістична регресія з L2-регуляризацією, демонструють більш стабільну поведінку на різних наборах текстів.

Варто зазначити, що ефективність класичних алгоритмів значною мірою залежить від якості підготовки даних. Попередня очистка тексту – видалення пунктуації, стоп-слів, нормалізація, лематизація – може істотно вплинути на результат класифікації. Для українських вакансій це питання набуває особливої актуальності через морфологічну складність мови. Тому навіть при використанні класичних моделей доцільно приділяти увагу тонкому налаштуванню процесів підготовки даних.

У сучасних гібридних системах класичні методи машинного навчання часто поєднуються з новітніми підходами, наприклад, вони можуть використовувати вектори, отримані за допомогою моделей Word2Vec або BERT, як вхідні ознаки. Таким чином досягається компроміс між пояснюваністю результатів і точністю класифікації. У цьому сенсі класичні алгоритми не втрачають своєї цінності, а стають невід’ємною частиною комплексних систем виявлення фейкових вакансій.

Проте класичні методи мають обмеження: вони погано працюють з великими корпусами текстів, не враховують контекст та не здатні розуміти глибинні смислові відносини між словами. Це призвело до появи більш складних підходів, заснованих на нейронних мережах.

1.3.2 Огляд глибинних нейронних мереж та трансформерів

Сучасні дослідження у сфері виявлення фейкового контенту переважно базуються на глибинному навчанні. Використання моделей типу LSTM (Long Short-Term Memory), GRU (Gated Recurrent Unit) та CNN (Convolutional Neural Network) дало можливість покращити точність класифікації за рахунок урахування послідовності слів та локальних контекстів.

Подальшим проривом стали трансформерні архітектури, серед яких ключову роль відіграють моделі BERT, RoBERTa, XLNet, ALBERT тощо. Вони дозволяють враховувати двосторонній контекст, тобто розуміти значення слова не лише за попередніми, а й за наступними словами в реченні. Це суттєво підвищує якість аналізу текстів, особливо у випадках, коли фейковість виражена непрямо або завуальовано.

Для виявлення фейкових вакансій такі моделі дають змогу враховувати лексичні, семантичні та стилістичні особливості тексту, наприклад: відсутність конкретики у вимогах, надмірне використання позитивних емоційних фраз («чудова робота», «легкі гроші»), відсутність інформації про компанію або адресу, граматичні помилки чи штучно сформульовані описи.

Деякі дослідники комбінують підхід BERT із додатковими шарами нейронних мереж для підвищення стабільності результатів або використовують fine-tuning на спеціально зібраних наборах даних вакансій.

1.3.3 Порівняння комбінованих систем та ансамблевих методів

Ще одним напрямом розвитку є створення комбінованих систем, які поєднують кілька моделей або різні типи ознак. Наприклад, у межах однієї системи може використовуватися такі ознаки:

– векторизація тексту за допомогою TF-IDF або BERT;

– додаткові числові характеристики вакансії – кількість символів, наявність посилань, структура речень;

– метадані – джерело публікації, домен сайту, частота оновлення вакансії тощо.

Такі ансамблеві підходи дозволяють покращити узагальнювальну здатність моделі та зменшити ризик помилкової класифікації. У цьому випадку результати кількох моделей, наприклад, логістичної регресії, Random Forest і BERT можуть комбінуватися за допомогою методу Stacking або Voting Classifier.

1.3.4 Використання семантичного аналізу та ознак достовірності

Окрім традиційного машинного навчання, у сучасних підходах активно застосовується семантичний аналіз текстів. Ідея полягає у тому, щоб не просто визначати частотні характеристики слів, а розуміти їхній смисловий контекст. Для цього використовуються методи обробки природної мови (NLP), такі як:

– Named Entity Recognition (NER) – для виділення назв компаній, посад, локацій;

– Sentiment Analysis – для визначення емоційного забарвлення вакансії;

– Dependency Parsing – для аналізу синтаксичних зв’язків у реченні.

Наприклад, якщо у вакансії часто зустрічаються позитивно забарвлені епітети без конкретики, це може бути ознакою недостовірності.

Додатково до текстового аналізу деякі системи включають перевірку достовірності джерела — аналіз домену, репутації сайту або соціальної активності компанії. Також актуальним напрямом є інтеграція таких систем у реальні платформи пошуку роботи, що дозволить збирати нові дані в режимі реального часу та оперативно адаптувати моделі до змінних умов.

1.3.5 Виклики та напрями подальших досліджень

Незважаючи на значний прогрес, завдання виявлення фейкових вакансій залишається складним. Основні труднощі:

– нестача збалансованих і якісно розмічених даних – фейкових вакансій у реальних наборах зазвичай значно менше;

– швидка еволюція шахрайських стратегій;

– багатомовність та культурні відмінності у текстах.

У майбутньому очікується активний розвиток мультимодальних моделей, які поєднуватимуть текстовий аналіз із візуальними або поведінковими ознаками, наприклад, аналіз зображень у вакансіях, часу публікації, активності акаунтів роботодавців тощо.

1.4 Постановка задачі дослідження

У сучасних умовах цифрової трансформації та зростання кількості онлайн платформ для пошуку роботи питання перевірки достовірності вакансій набуває особливої актуальності. Тому виникає необхідність розробки інтелектуальних систем, здатних автоматично аналізувати текст вакансій і визначати їхню надійність. Саме це завдання і стало основою даного дослідження.

Об’єктом дослідження є процес автоматизованого аналізу текстів вакансій з метою виявлення недостовірних або шахрайських оголошень.

Метою дослідження є створення вебсервісу, який дозволяє автоматично оцінювати підозрілість вакансії за її текстом, використовуючи сучасні алгоритми машинного навчання.

Для досягнення поставленої мети необхідно вирішити такі завдання:

– провести аналіз сучасних методів виявлення фейкового контенту та існуючих підходів до обробки текстової інформації;

– дослідити ефективність різних методів векторизації текстів для представлення вакансій у числовій формі;

– розглянути класичні та сучасні алгоритми машинного навчання для класифікації текстових даних;

– зібрати та підготувати корпус даних із реальних та фейкових вакансій для навчання моделі;

– розробити модель машинного навчання, здатну аналізувати вакансії та визначати рівень їх достовірності;

– створити вебінтерфейс на основі React, що забезпечує зручну взаємодію користувача з системою;

– інтегрувати модель із бекендом, розробленим на Python, для здійснення запитів та обчислення результатів аналізу;

– провести тестування системи на реальних прикладах вакансій та оцінити точність моделі;

– побудувати аналітичні графіки та візуалізації статистики фейкових вакансій для підвищення інформативності результатів.

Отже, у межах цього дослідження сформульовано комплексну задачу створення інтелектуальної системи для автоматичного аналізу вакансій. Реалізація поставлених завдань сприятиме підвищенню безпеки користувачів під час пошуку роботи та може стати основою для подальшого розвитку систем штучного інтелекту, орієнтованих на виявлення недостовірного контенту у відкритих джерелах.

Крім того, отримані результати відкривають можливість побудови масштабованих платформ, здатних обробляти великі обсяги текстових даних та адаптуватися до нових типів загроз. Запропонований підхід демонструє потенціал поєднання сучасних мовних моделей і веб-технологій для створення ефективних інструментів захисту користувачів у цифровому середовищі.

2 ВИКОРИСТАННЯ МЕТОДІВ ОБРОБКИ ПРИРОДНОЇ МОВИ У ЗАДАЧІ ВИЯВЛЕННЯ НЕДОСТОВІРНИХ ВАКАНСІЙ

2.1 Використання сучасних мовних моделей (LLM) для аналізу вакансій

Упродовж останніх років розвиток мовних моделей великого масштабу (Large Language Models, LLM) суттєво змінив підхід до аналізу текстових даних. Якщо раніше задачі класифікації текстів вирішувалися за допомогою класичних методів машинного навчання та простих векторизацій, таких як Bag-of-Words або TF-IDF, то сьогодні основну роль відіграють глибокі нейронні архітектури, здатні навчатися на мільярдах слів і враховувати контекст кожного речення. LLM-моделі – це універсальний інструмент, який сьогодні застосовується в таких сферах:

– у бізнесі – аналіз текстів, автоматизація документообігу;

– в освіті – автоматичної перевірки текстів**,** створення навчальних матеріалів**,** підказок при написанні коду або есе**,** генерації тестів та оціночних коментарів;

– у сфері безпеки – виявлення шахрайства та фейків;

– у медицині – аналізувати медичні записи, симптоми, дослідницькі статті, а також спілкуватися з пацієнтами природною мовою;

– у розробці ПЗ – допомагати у написанні коду, документуванні, налагодженні помилок. Це лежить в основі систем типу GitHub Copilot, які суттєво підвищують продуктивність розробників.

Для задачі виявлення недостовірних вакансій використання LLM є надзвичайно перспективним. Такі моделі, як BERT, RoBERTa, DistilBERT, GPT, ELECTRA чи XLM-R, дозволяють не лише враховувати частоту вживання слів, а й розуміти зміст і контекст тексту, виявляти приховані смислові зв’язки між словами та реченнями. Це особливо важливо для розпізнавання фейкових вакансій, адже вони часто містять тонкі мовні маніпуляції, емоційно забарвлені

фрази або аномальні структури тексту, які класичні методи не здатні адекватно оцінити [8]. У таблиці 2.1 представлено порівняльну характеристику основних представників LLM.

Таблиця 2.1 – Основні представники LLM

Модель	**Розробник**	**Рік**	**Особливості**
GPT-3 / GPT-4 / GPT-5	OpenAI	2020–2024	Потужні генеративні моделі, універсальні для діалогу, коду, аналізу тексту
BERT / RoBERTa / DistilBERT	Google, Meta	2018–2019	Класичні моделі для завдань аналізу тексту, класифікації, пошуку
T5 / Flan-T5	Google	2020	Формат «текст - текст», зручний для узагальнення та переформулювання
LLaMA / Mistral / Claude	Meta, Anthropic	2023–2024	Оптимізовані відкриті або комерційні LLM, придатні для розгортання локально
Gemini (ex-Bard)	Google DeepMind	2024	Мультимодальна модель (працює з текстом, зображеннями, відео)

Мовні моделі великого масштабу базуються на архітектурі трансформерів (Transformers), запропонованій компанією Google у 2017 році. На відміну від рекурентних нейронних мереж, які обробляли текст послідовно, трансформери використовують механізм уваги, що дозволяє моделі одночасно враховувати контекст усіх слів у реченні. Завдяки цьому модель може визначати, які слова найбільше впливають на загальний зміст тексту, і створювати семантично насичене векторне представлення.

У контексті розробленої системи аналізу вакансій, LLM-модель може бути використана як основний компонент для: отримання контекстних ембедингів – перетворення кожного слова або речення у вектор, який враховує зміст, класифікації текстів – визначення ймовірності, що вакансія є фейковою, виявлення ключових ознак або слів, що впливають на рішення моделі (інтерпретація результату).

Типовий процес використання мовної моделі для аналізу вакансій включає кілька етапів:

– підготовка тексту – очищення, нормалізація та токенізація;

– кодування за допомогою LLM – перетворення тексту у контекстні вектори через попередньо натреновану модель, наприклад «bert-base-uncased» або україномовну модель «ukr-roberta-base»;

– додавання класифікаційного шару – поверх отриманих векторів додається простий лінійний або feed-forward шар, який навчається розділяти класи «достовірна» або «недостовірна».

Тонке донавчання – модель перенавчається на спеціалізованій вибірці текстів вакансій.

Інференс – під час роботи вебсервісу користувацький текст проходить ті самі етапи, після чого система повертає ймовірність фейковості.

Завдяки великій ємності та контекстній обізнаності LLM-моделі можуть вловлювати підсвідомі мовні патерни, характерні для шахрайських вакансій, наприклад:

– надмірно привабливі обіцянки без конкретики («висока зарплата без досвіду», «прибуток відразу»);

– нетипова структура тексту (короткі фрази, повтори, відсутність деталей про компанію);

– використання емоційних або тиснучих формулювань («не втрать шанс», «потрібно вже сьогодні»).

Однією з переваг використання LLM у цій задачі є можливість адаптації моделі до нових даних без повного перенавчання [9]. Тобто, коли у вебсистемі

накопичується нова інформація про вакансії, модель може бути донавчена на цих прикладах, підвищуючи точність класифікації.

Окрім безпосередньої класифікації, LLM можна також використовувати для генерації пояснень: модель здатна підсвічувати фрагменти тексту, які вплинули на її рішення. Це робить систему більш інтерпретованою та зручною для користувача, адже він може побачити, які саме речення або слова викликали підозру.

У порівнянні з класичними методами машинного навчання, які оперують поверхневими ознаками – *n*-грамами, частотами, LLM забезпечують глибше розуміння тексту та здатність працювати з різними мовами й стилями.

Завдяки цьому вони є найефективнішим підходом для аналізу вакансій, де важливу роль відіграє семантичне значення фраз і контекст взаємодії слів.

2.2 Аналіз структури та лінгвістичних особливостей вакансій

У процесі дослідження особливу увагу було приділено текстовому аналізу вакансій, адже саме лінгвістичні характеристики часто дозволяють виявити потенційно фейкові оголошення. Текст вакансії відображає не лише зміст, але й стиль комунікації компанії, її рівень професійності, увагу до деталей та достовірність намірів. Відмінності між справжніми та підозрілими вакансіями стають помітними саме на рівні лексики, синтаксису та семантики.

Типові лексичні патерни у фейкових вакансіях демонструють схильність авторів до використання емоційно забарвлених або узагальнених фраз, покликаних викликати інтерес і довіру. Часто такі тексти містять слова, що акцентують увагу на швидкому прибутку, легкості процесу працевлаштування або унікальній можливості. Серед характерних прикладів – вислови на кшталт «заробляй вже сьогодні», «висока оплата без досвіду», «гнучкий графік, вільний вибір часу». Такі патерни не лише створюють враження привабливості, а й свідчать про маніпулятивність, що є типовою рисою фейкових вакансій.

Порівняння реальних і підозрілих вакансій показує, що достовірні оголошення зазвичай мають більший обсяг тексту, чітку структуру та містять деталі щодо обов’язків, вимог і умов праці. У них рідше зустрічаються вигуки чи заклики до дії, натомість більше конкретики: назви інструментів, технологій, навичок, обов’язків. Підозрілі ж вакансії, як правило, коротші, менш структуровані й часто мають пропуски у ключових полях, таких як опис компанії, досвід роботи або рівень освіти. Це підтверджується також статистичними показниками з аналізованих джерел: середня довжина опису справжніх вакансій значно перевищує середню довжину текстів фейкових, а частота повторення окремих рекламних слів у підозрілих випадках у кілька разів вища.

Роль синтаксичних і семантичних ознак у визначенні достовірності є надзвичайно важливою, оскільки сучасні моделі машинного навчання не обмежуються поверхневим аналізом слів. Вони враховують контекст і відношення між словами в реченні. Наприклад, словосполучення «робота без досвіду» у поєднанні з «висока зарплата» може бути сигналом недостовірності, тоді як окремо кожен із цих елементів не обов’язково свідчить про фейковість. Саме тому аналіз текстів вакансій відбувається не лише на рівні окремих лексем, а й на рівні контекстуальних залежностей, які формують загальне враження про зміст повідомлення.

У рамках дослідження було відзначено, що реальні вакансії відзначаються стабільністю граматичних структур, логічною послідовністю речень та відсутністю надмірних обіцянок. Тексти таких оголошень містять нейтральну лексику, спрямовану на інформування, а не на емоційний вплив. Для фейкових, навпаки, характерна нестандартна побудова речень, надлишок прикметників та вигуків, уживання кліше й нетипових скорочень. Крім того, у підозрілих текстах часто трапляються орфографічні або пунктуаційні помилки, що свідчить про відсутність професійного підходу [10].

Важливим спостереженням є те, що моделі, зокрема великі мовні (LLM), здатні розпізнавати ці відмінності не лише на рівні статистики, а й через

розуміння змісту речень. Це дозволяє враховувати семантичний контекст, розрізняти тональність повідомлення та виявляти суперечливі твердження. Наприклад, коли у вакансії одночасно заявляють «гнучкий графік» і «обов'язкова присутність у офісі повний день», модель може оцінити таку невідповідність як ознаку недостовірності.

Додатковий аспект, який варто враховувати під час аналізу вакансій, – це прагматичний рівень тексту, тобто ціль автора та комунікативний намір. У реальних оголошеннях роботодавці зазвичай намагаються надати потенційному працівнику чітку та повну інформацію: умови праці, графік, обов'язки, контакти, можливість кар'єрного зростання. У фейкових текстах, навпаки, основною метою часто є залучення уваги або отримання персональних даних користувачів, тому тексти мають маніпулятивний характер. Це може проявлятися у вживанні наказового способу – «заповни анкету прямо зараз», у перебільшених обіцянках – «безкоштовне навчання і миттєвий заробіток», або у надмірній кількості вигуків і емоційних висловів.

Важливо зазначити, що навіть синтаксичні дрібниці можуть мати значення. Наприклад, короткі речення, що повторюються, або відсутність складнопідрядних конструкцій може свідчити про автоматизоване створення тексту. Деякі фейкові вакансії генеруються ботами або створюються за шаблонами, що призводить до одноманітності лексики та спрощення структури. У цьому контексті ефективним підходом є виявлення текстових шаблонів, які повторюються у великій кількості підозрілих оголошень. Виявлення таких шаблонів дозволяє автоматично позначати потенційно небезпечні записи для подальшої перевірки.

Окремої уваги заслуговує дослідження семантичних суперечностей, що можуть бути характерними для фейкових текстів. Наприклад, у вакансії може бути зазначено «робота у великій міжнародній компанії», але при цьому контактна особа має електронну пошту на безкоштовному домені, що викликає сумніви у достовірності. Подібні деталі можна виявляти за допомогою автоматизованих моделей, які оцінюють не лише текст, а й метадані – домен

електронної пошти, структуру посилань, згадування брендів або місцезнаходження.

Дослідження, проведені з використанням датасету «Fake Job Postings Dataset» (Kaggle), також підтверджують, що лінгвістичні характеристики є одним із найнадійніших індикаторів фейковості. Зокрема, виявлено, що у фейкових вакансіях значно частіше зустрічаються слова, пов’язані з обіцянками («гарантовано», «миттєво», «безкоштовно»), тоді як у справжніх переважає ділова лексика («обов’язки», «вимоги», «команда», «проект»). Цей факт свідчить про те, що фейкові оголошення мають емоційно-мотиваційний характер, тоді як справжні – інформаційно-професійний [11].

Ще однією важливою особливістю є відмінність у використанні частин мови. Для реальних вакансій типовим є переважання іменників і дієслів, пов’язаних із професійною діяльністю («аналіз», «розробка», «тестування», «управління»), тоді як у фейкових часто домінують прикметники («високи», «швидкий», «легкий», «кращий») та прислівники («миттєво», «без зусиль»). Це зміщує фокус із опису реальної роботи на створення позитивного емоційного враження.

Оскільки сучасні мовні моделі (LLM) здатні опрацьовувати глибинні семантичні зв’язки, вони можуть розпізнавати не лише статистичні відмінності, а й інтенцію тексту. Наприклад, модель може визначити, що в реченні «Компанія шукає відповідального спеціаліста для довгострокової співпраці» присутня ділова інтонація, тоді як фраза «Не зволікай, почни заробляти вже сьогодні!» має рекламний характер. Це дає змогу моделі робити більш точні висновки щодо достовірності.

Окрім суто лінгвістичного рівня, важливим є також стилістичний аспект. Для реальних вакансій типовим є офіційно-діловий стиль із нейтральною лексикою, коректним оформленням тексту, поділом на логічні частини. У фейкових оголошеннях часто спостерігається відсутність форматування, надлишок великих літер, емоційних знаків «!!!», а також використання

шаблонних фраз, які повторюються в різних публікаціях. Це створює додаткову ознаку, за якою можна виявляти неправдивий контент.

Загалом, аналіз структури й лінгвістичних особливостей вакансій дозволяє побудувати ефективну систему класифікації, що враховує не лише поверхневі статистичні показники, а й глибинні смислові закономірності. Поєднання лінгвістичного, синтаксичного та семантичного рівнів аналізу дає змогу створити модель, здатну розпізнавати навіть ті тексти, які на перший погляд виглядають достовірно, але мають приховані ознаки фейковості. Це підкреслює важливість застосування сучасних методів обробки природної мови при розв’язанні задачі виявлення недостовірних вакансій.

Таким чином, лінгвістичний аналіз стає основою для подальшого машинного навчання, оскільки саме через структуру та лексику текстів формується набір ознак, що визначають справжність або фейковість вакансії. Поєднання статистичного аналізу з контекстуальним підходом дозволяє створити більш точну модель класифікації, здатну ефективно працювати навіть у випадках, коли фейкові оголошення імітують стиль справжніх.

2.3 Огляд алгоритмів машинного навчання для класифікації текстів

2.3.1 Метод Логістична регресія

Логістична регресія є одним із найпоширеніших та базових методів машинного навчання, який широко застосовується для задач класифікації. Незважаючи на свою простоту, цей алгоритм показує високу ефективність при розв’язанні задач, де необхідно передбачити одну з двох категорій – наприклад, класифікацію вакансій як достовірних або недостовірних.

Основна ідея логістичної регресії полягає в тому, щоб знайти залежність між вхідними змінними та ймовірністю належності об’єкта до певного класу. На відміну від лінійної регресії, яка передбачає безперервне значення, логістична регресія прогнозує ймовірність, що значення цільової змінної дорівнює 1, тобто,

що вакансія є підозрілою. Для цього використовується сигмоїдна функція, яка перетворює будь-яке дійсне число у діапазон [0;1]:

$$\sigma(z) = \frac{1}{1+e^{-z}}, \tag{2.1}$$

$$z = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \cdots + \omega_n x_n, \tag{2.2}$$

де x_i – вхідні ознаки;

ω_1 – вагові коефіцієнти, які підбираються під час навчання моделі.

У контексті аналізу вакансій вхідними ознаками можуть бути кількісні характеристики тексту – такі як довжина оголошення, частота вживання окремих слів, кількість контактних даних, наявність гіперпосилань, а також лексичні чи граматичні патерни [12]. Після обробки тексту, а саме токенізації, лематизації, видалення стоп-слів, кожна вакансія може бути представлена у вигляді вектору числових ознак. Логістична регресія обчислює вагу кожної ознаки та визначає, наскільки вона впливає на підозрілість вакансії.

Важливим етапом є навчання моделі, тобто підбір таких коефіцієнтів ω_1, які мінімізують помилку прогнозу. Для цього найчастіше використовується метод градієнтного спуску, що поступово оновлює ваги, наближаючись до оптимального розв'язку. Цільова функція, яку мінімізує алгоритм, має вигляд крос-ентропійної втрати:

$$L = -\frac{1}{m}\sum_{i=1}^{m}[y_i \log \hat{y}_i + (1 - y_i) \times \log 1 - \hat{y}_i], \tag{2.3}$$

де y_i – справжнє значення (0 або 1);

$\hat{y}_i$ – прогноз моделі.

У випадку з фейковими вакансіями логістична регресія дозволяє не лише класифікувати записи, а й оцінювати рівень їх підозрілості у вигляді ймовірності. Наприклад, якщо модель повертає значення 0,87, що можна інтерпретувати як

87% ймовірності того, що дана вакансія є недостовірною. Це зручно для побудови користувацького інтерфейсу, у якому результат подається у відсотках.

Застосування логістичної регресії у цій задачі має кілька важливих переваг. По-перше, модель є інтерпретованою – тобто можна зрозуміти, які саме ознаки впливають на класифікацію [13]. Це важливо для пояснюваності рішень системи, особливо в контексті роботи з контентом, що може впливати на користувачів. По-друге, алгоритм працює швидко і не вимагає великих обчислювальних ресурсів, тому може використовуватися у вебсервісах у режимі реального часу. По-третє, логістична регресія добре справляється з лінійно роздільними даними, що часто зустрічається при аналізі текстів після векторизації за допомогою TF-IDF чи Bag-of-Words.

Однак, як і будь-який алгоритм, логістична регресія має свої обмеження. Вона не завжди здатна відобразити складні нелінійні залежності між ознаками, що обмежує її ефективність при роботі з контекстними або семантичними характеристиками тексту. Для вирішення таких задач часто використовують глибші моделі – нейронні мережі або трансформери. Втім, логістична регресія часто використовується як базовий еталон для порівняння з більш складними алгоритмами.

У межах розробленої системи логістична регресія може виконувати роль первинного класифікатора. Наприклад, вона здатна швидко відсіяти очевидно фейкові вакансії на основі ключових ознак, тоді як подальший аналіз може виконуватися більш складною нейромережею. Такий підхід поєднує швидкодію та глибину аналізу, забезпечуючи баланс між продуктивністю і точністю.

Загалом, логістична регресія є важливим інструментом для побудови систем автоматичного розпізнавання недостовірних вакансій. Вона демонструє, як навіть відносно прості математичні моделі можуть ефективно виявляти закономірності у текстових даних, формуючи основу для більш складних систем на базі сучасних нейромережевих технологій.

2.3.2 Метод Random Forest

Random Forest є одним із найефективніших методів машинного навчання, який базується на ансамблевому підході. Його суть полягає у поєднанні великої кількості простих моделей – дерев рішення, кожне з яких навчається на випадковій підмножині даних та ознак. Після навчання результати всіх дерев агрегуються, і фінальне рішення приймається шляхом голосування або усереднення прогнозів. Такий підхід дозволяє зменшити ймовірність переобучення та підвищити загальну точність моделі.

У контексті виявлення недостовірних вакансій метод Random Forest є особливо ефективним, оскільки враховує велику кількість різноманітних ознак тексту: від статистичних характеристик – довжина опису, кількість слів, частота певних термінів, до лінгвістичних показників – вживання емоційної лексики, наявність контактної інформації, стилістичні маркери. Кожне дерево у моделі спеціалізується на своїй підмножині ознак, що забезпечує різноманітність рішень і підвищує стійкість системи до шуму у даних.

Однією з ключових ідей методу є випадковість на етапі навчання. По-перше, кожне дерево будується на основі випадкової вибірки даних із заміною метод bootstrap. По-друге, під час побудови кожного вузла дерева розглядається не весь набір ознак, а лише випадково обрана його частина. Це запобігає надмірній кореляції між деревами та сприяє кращій узагальнювальній здатності ансамблю.

У випадку задачі класифікації вакансій алгоритм працює наступним чином:

Крок 1. Модель формує набір дерев рішень, кожне з яких прогнозує, чи є вакансія фейковою.

Крок 2. Для нового тексту система обчислює вектор ознак після етапу попередньої обробки та векторизації.

Крок 3. Кожне дерево робить свій власний прогноз (0 – справжня, 1 – фейкова).

Крок 4. Визначити остаточний результат шляхом голосування: якщо більшість дерев класифікують вакансію як фейкову, фінальний прогноз також буде 1.

Цей механізм дозволяє досягати високої точності навіть при складних нелінійних залежностях у даних. На відміну від логістичної регресії, яка моделює лінійну межу розділення, Random Forest здатний враховувати складні комбінації ознак і взаємодії між ними [14]. Наприклад, певна комбінація слів – «висока зарплата», «без досвіду», «миттєвий дохід», може бути більш інформативною, ніж окремо взяте слово, і саме ансамблевий підхід дозволяє це виявити.

Під час навчання кожне дерево намагається знайти оптимальні пороги поділу даних за різними ознаками. Алгоритм використовує метрики, такі як інформаційна виграшність або Gini impurit, щоб визначити, яке розгалуження найбільш ефективно розділяє фейкові та реальні вакансії. Таким чином, модель поступово формує складну багаторівневу структуру, здатну відображати приховані закономірності у текстових даних.

Оцінка важливості ознак є ще однією перевагою Random Forest. Після навчання можна визначити, які характеристики мають найбільший вплив на класифікацію. Наприклад, модель може виявити, що найбільш значущими є частота слів «заповни», «реєстрація», «отримай», або відсутність згадки про конкретну компанію. Такі результати не лише покращують інтерпретацію моделі, але й дозволяють аналітично описати типові ознаки фейкових вакансій, що підвищує практичну цінність дослідження.

Ще однією сильною стороною Random Forest є його стійкість до викидів та пропусків у даних. Оскільки рішення приймається на основі колективного голосування великої кількості дерев, вплив аномальних або помилкових прикладів мінімізується. Це особливо корисно для реальних текстових даних, де оголошення можуть містити неточності, орфографічні помилки або неповну інформацію [15].

Попри численні переваги, метод має і певні недоліки. Найочевидніший з них – висока обчислювальна складність при великій кількості дерев, що може

впливати на швидкість аналізу в режимі реального часу. Також модель може ставати складною для інтерпретації, оскільки сукупне рішення формується із сотень дерев. Проте ці обмеження можна частково компенсувати за допомогою оптимізації гіперпараметрів – наприклад, шляхом зменшення максимальної глибини дерев або кількості використаних ознак при розгалуженні.

Для задачі виявлення недостовірних вакансій Random Forest демонструє високу ефективність. Він поєднує простоту реалізації, гнучкість і точність, забезпечуючи надійний результат навіть без складного попереднього налаштування. Багато досліджень зокрема, експерименти на базі датасету Kaggle «Fake Job Postings Dataset» показують, що цей алгоритм здатен досягати точності класифікації понад 95% при правильному налаштуванні параметрів [16].

Таким чином, метод випадкового лісу є потужним інструментом для розв’язання задач аналізу текстових даних, особливо коли йдеться про виявлення підозрілих або шахрайських публікацій. Його використання у складі системи аналізу вакансій забезпечує високу надійність, гнучкість і стабільність результатів, що робить його одним із найефективніших класичних підходів у поєднанні з сучасними методами обробки природної мови.

2.3.3 Метод опорних векторів

Метод опорних векторів (Support Vector Machine, SVM) є одним із найефективніших класичних алгоритмів машинного навчання, який широко застосовується для задач класифікації, у тому числі для виявлення фейкових вакансій. Основна ідея методу полягає в пошуку оптимальної гіперплощини, що розділяє дані на класи з максимальним зазором. Тобто SVM намагається знайти таку межу між класами, яка не просто відділяє їх, а робить це з найбільшим запасом – це дозволяє досягти високої узагальнювальної здатності моделі.

На практиці тексти вакансій перед класифікацією перетворюються у векторне представлення, наприклад, за допомогою TF-IDF або методів

векторизації слів. Отримані вектори описують статистичні характеристики тексту, які потім подаються на вхід SVM. Після навчання модель формує гіперплощину, що відокремлює фейкові вакансії від реальних. Якщо нова вакансія потрапляє до області, що відповідає класу «фейкова», система відповідно маркує її як підозрілу.

Однією з ключових переваг SVM є її здатність ефективно працювати у випадках, коли кількість ознак значно перевищує кількість прикладів у вибірці. Це особливо важливо для текстових задач, де кількість слів і фраз може бути дуже великою. Алгоритм не потребує великої кількості даних для навчання, однак вимагає ретельного налаштування параметрів, зокрема вибору ядра. Ядро визначає спосіб, у який дані перетворюються у вищий вимір, де розділення класів стає можливим.

Для задачі виявлення фейкових вакансій зазвичай використовують радіально-базисне (RBF) ядро або лінійне ядро. Лінійне ядро добре підходить у випадках, коли дані є лінійно роздільними – тобто між класами можна провести пряму межу. Якщо ж структура даних складніша, а класи мають нелінійні залежності, RBF-ядро дозволяє моделі ефективніше відображати приховані зв’язки між ознаками.

Ще однією перевагою SVM є стійкість до переобучення, особливо якщо правильно обрати параметри регуляризації. У задачі виявлення недостовірних вакансій це важливо, оскільки дані можуть містити як шум, наприклад, оголошення з некоректно заповненими полями, так і неоднозначні приклади, які важко однозначно класифікувати. SVM дозволяє знайти баланс між точністю на тренувальних даних та узагальнювальною здатністю на нових текстах.

Недоліком методу є те, що він не завжди добре масштабується на дуже великі набори даних, оскільки навчання може бути обчислювально затратним. Крім того, результати SVM менш інтерпретовані, ніж, наприклад, у логістичній регресії: важко пояснити, які саме слова або фрази найбільше вплинули на рішення моделі. Проте в поєднанні з методами візуалізації або аналізом ваг ознак

за допомогою TF-IDF можливо отримати певне уявлення про ключові фактори класифікації.

Загалом, використання SVM у задачі аналізу вакансій є виправданим вибором, особливо на етапі побудови базової моделі для оцінки якості підготовлених даних. Вона добре підходить для текстів середнього розміру, дозволяє ефективно розділяти класи з мінімальними помилками та може служити еталоном для подальшого порівняння з більш складними моделями, такими як нейронні мережі або LLM.

2.3.4 Дослідження переваг нейронних мереж

Нейронні мережі є одним із найпотужніших інструментів сучасного машинного навчання, які продемонстрували надзвичайну ефективність у задачах обробки природної мови, зокрема у виявленні фейкових вакансій. Їхня головна перевага полягає у здатності автоматично виділяти складні закономірності та приховані залежності в текстах без потреби вручну створювати ознаки. Це робить нейронні моделі особливо корисними для роботи з неструктурованими даними, якими є текстові описи вакансій.

На відміну від класичних методів, таких як логістична регресія чи SVM, нейронні мережі можуть опрацьовувати не лише частотні характеристики слів, а й контекст, у якому вони вживаються. Це особливо важливо при аналізі вакансій, адже одна й та сама фраза може мати різне значення залежно від контексту. Наприклад, фейкові вакансії часто використовують занадто загальні або привабливі формулювання – «висока зарплата без досвіду», «швидкий кар’єрний ріст», тоді як реальні оголошення мають більш конкретний опис вимог і обов’язків [17].

У задачах класифікації текстів вакансій зазвичай застосовуються різні архітектури нейронних мереж – від простих багатошарових перцептронів (MLP) до складніших моделей, таких як рекурентні (RNN), згорткові (CNN) або

трансформери. Кожен із цих підходів має свої переваги. Наприклад, RNN добре підходять для аналізу послідовностей, оскільки враховують порядок слів, тоді як CNN ефективно виявляють локальні патерни у тексті. Найбільш сучасні архітектури – такі як BERT або GPT – базуються на механізмі уваги, який дозволяє моделі аналізувати взаємозв'язки між словами у всьому реченні одночасно, а не лише послідовно.

У контексті виявлення недостовірних вакансій нейронна мережа може навчатися на великих наборах даних, що містять реальні та фейкові оголошення. Після векторизації тексту, наприклад, за допомогою Word2Vec або моделей типу Sentence-BERT, дані подаються на вхід мережі. У процесі навчання мережа коригує свої ваги, навчаючись розпізнавати характерні ознаки обох типів вакансій. З часом вона починає вловлювати навіть тонкі відмінності у стилі, лексиці чи структурі повідомлень.

Перевагою нейронних мереж є їхня здатність до узагальнення – вони можуть успішно працювати з новими вакансіями, які не були присутні у тренувальній вибірці, якщо контекст і лінгвістичні особливості схожі на відомі приклади. Це особливо корисно у швидкозмінному середовищі онлайн-рекрутингу, де шахраї постійно змінюють формулювання своїх оголошень.

Важливою характеристикою нейронних моделей є можливість використання попередньо навчених моделей. Такі моделі, як BERT, RoBERTa чи GPT, уже навчені на величезних текстових корпусах і розуміють загальні закономірності мови. Це дозволяє досягти високих результатів навіть при відносно невеликій кількості специфічних даних про вакансії. Достатньо провести донавчання моделі на тематичному наборі даних, і вона адаптується до конкретної задачі – у нашому випадку до класифікації достовірності вакансій.

Серед недоліків нейронних мереж можна відзначити значні обчислювальні витрати – як на навчання, так і на передбачення, а також потребу у великій кількості якісно підготовлених даних. Крім того, інтерпретованість таких моделей часто є обмеженою: складно пояснити, чому саме модель визнала певну вакансію фейковою. Проте цей недолік поступово компенсується появою

методів інтерпретації, таких як LIME або SHAP, які дозволяють оцінити вплив окремих слів на рішення нейронної мережі.

Загалом, використання нейронних мереж у задачі аналізу вакансій є закономірним етапом розвитку підходів до обробки природної мови. Вони забезпечують високу точність класифікації, здатність враховувати контекст і адаптуватися до нових даних, що робить їх одним із найефективніших інструментів для виявлення фейкових вакансій у сучасному цифровому середовищі.

2.4 Семантичне моделювання контенту

Семантичне моделювання текстів є ключовим етапом у побудові системи, здатної розуміти зміст вакансій не лише на рівні окремих слів, а й у контексті всього повідомлення. Традиційні методи векторизації, такі як Bag of Words або TF-IDF, оперують лише статистичними частотами слів, не враховуючи їхній взаємозв’язок у реченні. Це призводить до втрати семантики – тобто глибшого змістового зв’язку між словами. Сучасні трансформерні моделі, такі як BERT (Bidirectional Encoder Representations from Transformers) та RoBERTa (Robustly Optimized BERT Approach), значно підвищили якість обробки природної мови, оскільки дозволяють враховувати контекст кожного слова з обох боків речення одночасно.

При аналізі вакансій надзвичайно важливо враховувати не лише частотність термінів, а й контекст, у якому вони вживаються. Наприклад, фраза «робота без досвіду» може бути нормальною у студентських пропозиціях, але у поєднанні з «висока зарплата» або «виплати щотижня» може свідчити про шахрайство. Саме такі контекстуальні відмінності трансформери вловлюють набагато краще, ніж класичні алгоритми [18].

BERT працює за принципом бідірекційного кодування, тобто під час обробки кожного слова він враховує контекст і ліворуч, і праворуч. Це дозволяє

моделі глибше зрозуміти смисл речення. Наприклад, слово «оплата» у різних контекстах може означати різне: «висока оплата» – позитивний аспект, а «без оплати» – потенційно фейкову або підозрілу вакансію.

RoBERTa є удосконаленою версією BERT, яка навчалася на більшому корпусі текстів і використовує покращену стратегію маскування слів. У задачі виявлення недостовірних вакансій RoBERTa демонструє кращу стабільність і точність, особливо коли набір даних є відносно невеликим.

Трансформер складається з енкодера та декодера, які використовують механізм уваги. У моделях на кшталт BERT або RoBERTa задіяно лише енкодерну частину. Основна ідея механізму уваги полягає в тому, що під час обробки кожного слова модель «звертає увагу» на всі інші слова у реченні та зважує їхню важливість.

Лістинг 2.1 Спрощена схема роботи трансформера:

[Текст вакансії] → [Токенізація] → [Ембеддинги слів]
↓
[Енкодер BERT / RoBERTa]
↓
[Векторне представлення вакансії]
↓
[Класифікатор (Dense Layer)]
↓
[Ймовірність фейковості]

Така архітектура дозволяє перетворити довільний текст у компактне багатовимірне векторне представлення, яке відображає його семантичний зміст. На основі цих векторів подальший класифікатор, наприклад, шар нейронної мережі або SVM може приймати рішення, наскільки ймовірно, що вакансія є фейковою.

Еволюцію підходів до аналізу текстових даних – від класичних статистичних методів до сучасних нейромережевих моделей на основі трансформерів показано у таблиці 2.2. Вона чітко демонструє, як із розвитком технологій змінювалося розуміння тексту в контексті машинного навчання і

відображає поступове ускладнення моделей і зростання їхньої здатності відображати семантичний зміст тексту.

Таблиця 2.2 – Порівняння ефективності моделей

Модель	**Основна ідея**	**Особливості**	**Переваги**
TF-IDF + Logistic Regression	Статистичний підхід	Враховує частоту слів	Простота, але не розуміє контекст
Word2Vec / GloVe + SVM	Семантичні вектори	Ураховує значення слів, але не контекст	Кращі результати, ніж TF-IDF
BERT / RoBERTa + Dense Layer	Трансформерна архітектура	Контекстуальне розуміння слів і речень	Висока точність, глибоке семантичне моделювання

Перший підхід, TF-IDF + Logistic Regression, показує базову логіку: модель оцінює частоту появи слів у документі, не розуміючи їхнього значення. Такі моделі добре працюють на структурованих, коротких текстах, але у випадку вакансій, де важливий контекст, вони часто дають хибні результати [19].

Наступний рівень – Word2Vec / GloVe + SVM – уже дозволяє враховувати певну семантику, тобто розуміти, що «зарплата» і «оплата» близькі за змістом. Проте ці моделі все ще не аналізують порядок слів або контекст – для них слово має одне фіксоване значення незалежно від ситуації. Це обмежує точність, особливо в неоднозначних або контекстно-залежних фразах, типових для фейкових вакансій.

Справжній прорив забезпечили трансформери – BERT і RoBERTa. Вони не лише створюють контекстуальні векторні представлення, але й навчаються «розуміти» взаємозв’язки між словами у реченні.

3 ДОСЛІДЖЕННЯ ПРОГРАМНИХ МЕТОДІВ ПОБУДОВИ СИСТЕМИ АНАЛІЗУ ВАКАНСІЙ

3.1 Формування та підготовка вибірки даних для навчання моделі

Для реалізації системи виявлення недостовірних вакансій необхідною умовою є наявність якісної та репрезентативної вибірки текстових даних.

Основним об’єктом аналізу в даному дослідженні є тексти вакансій, отримані з відкритих інтернет-ресурсів, платформ пошуку роботи, а також власноруч зібраних прикладів реальних і фейкових оголошень

Для навчання та тестування моделі було використано відкритий набір даних із платформи Kaggle. У таблиці 3.1 представлено приклад даних набору.

Таблиця 3.1 Приклад набору даних

	title	**description**	**requirements**	**employment type**	**industry**	**fraudulent**
0	Mental health nurse	Arm drive court sure vote. Earn $5000/week! Im...	Basic knowledge in live, no degree required. F...	Internship	IT	1
1	Conference centre manager	Government whom its bed go tax tree black.	Basic knowledge in seek, no degree required. F...	Part-Time	Finance	1
2	Engineer, land	I member discuss follow way there nation. Earn...	Basic knowledge in worker, no degree required....	Part-Time	IT	1
3	Forest/woodland manager	House across wait approach face. Earn $5000/we...	Basic knowledge in example, no degree required...	Full-Time	Education	1
4	Productin designer, film	Case best environmental full finally leader me...	Basic knowledge in smile, no degree required. ...	Temporary	Retail	1

Даний набір даних є одним із найбільш відомих і широко застосовуваних у дослідженнях, що стосуються виявлення фейкових вакансій. Він містить детальну інформацію про вакансії, розміщені на різних онлайн-ресурсах, та містить як реальні, так і підозрілі оголошення. Завдяки цьому датасет чудово підходить для задачі бінарної класифікації – визначення, чи є певна вакансія достовірною чи фейковою.

Основна перевага цього набору полягає у різноманітності представлених текстів – він охоплює вакансії з різних галузей, рівнів кваліфікації та форматів публікацій. Це дозволяє моделі навчитися розрізняти широкий спектр лінгвістичних патернів і стилістичних ознак, притаманних як справжнім, так і недостовірним оголошенням.

Датасет містить понад 17000 записів, кожен із яких включає такі основні поля:

– «title» – назва вакансії;

– «location» – місце роботи;

– «department», «salary_range», «company_profile» – структурована інформація про роботодавця;

– «description» – повний текст опису вакансії;

– «requirements» – вимоги до кандидата;

– «benefits» – переваги, які пропонує компанія;

– «fraudulent» – цільова змінна, що позначає достовірність (0 – реальна вакансія, 1 – фейкова).

Під час первинного аналізу даних було виявлено важливу проблему – у наявному фрагменті вибірки всі записи позначені як фейкові (рис. 3.1). Це означає, що значення цільової змінної fraudulent для кожного запису дорівнює 1. Така ситуація створює серйозні обмеження для подальшого навчання моделей машинного навчання, адже для класифікації необхідна наявність принаймні двох класів: позитивного – фейкові вакансії, та негативного – реальні вакансії.

Відсутність різноманітності в мітках робить неможливим коректне навчання моделі, оскільки алгоритм не має змоги побачити різницю між класами та визначити закономірності, за якими можна було б відрізняти достовірні вакансії від шахрайських. У такому випадку модель, незалежно від архітектури, чи то Logistic Regression, Random Forest, SVM або нейронна мережа, навчиться передбачати лише одну категорію, що фактично зводить її ефективність до нуля.

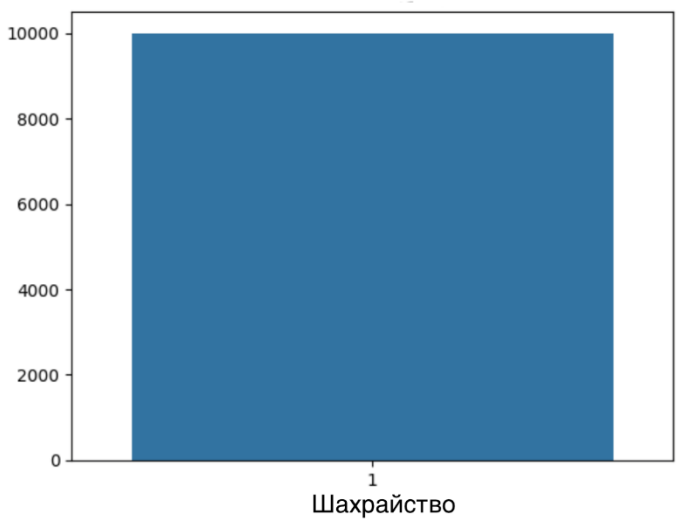

Рисунок 3.1 – Розподіл шахрайських вакансій

Таким чином, перед початком побудови моделі необхідно перевірити баланс класів у вибірці та, за потреби, або оновити датасет, включивши до нього приклади реальних вакансій, або здійснити ресемплінг даних із зовнішніх джерел. Лише після цього можливо забезпечити коректний процес навчання, валідації та тестування моделі.

Цей етап показав, наскільки важливим є ретельний аналіз структури та якості даних перед переходом до моделювання. Навіть якщо набір даних виглядає повним і добре структурованим, відсутність різноманітності у цільових мітках може повністю зруйнувати здатність системи навчитися ефективно розрізняти класи. Надалі було прийнято рішення скоригувати вибірку, щоб забезпечити наявність як фейкових, так і реальних вакансій, що дозволить моделі формувати більш точні та узагальнені рішення.

Важливо розглянути розподіл основних характеристик публікацій. Зокрема, значну аналітичну цінність становить дослідження типів зайнятості та галузей, у яких розміщувалися вакансії.

На рисунку 3.2 зображена діаграма розподілу типів зайнятості. З неї видно, що вибірка є досить збалансованою – кількість вакансій для різних типів зайнятості, таких як «Full-Time», «Part-Time», «Contract», «Internship» та «Temporary», приблизно однакова й коливається в межах 1900–2000 записів. Такий баланс є позитивним фактором для подальшого машинного аналізу, оскільки відсутня суттєва диспропорція між класами, що допомагає уникнути зміщення під час навчання моделі. Водночас, рівномірність може бути і результатом попередньої фільтрації або модифікації вибірки, тому цей аспект слід враховувати під час інтерпретації результатів.

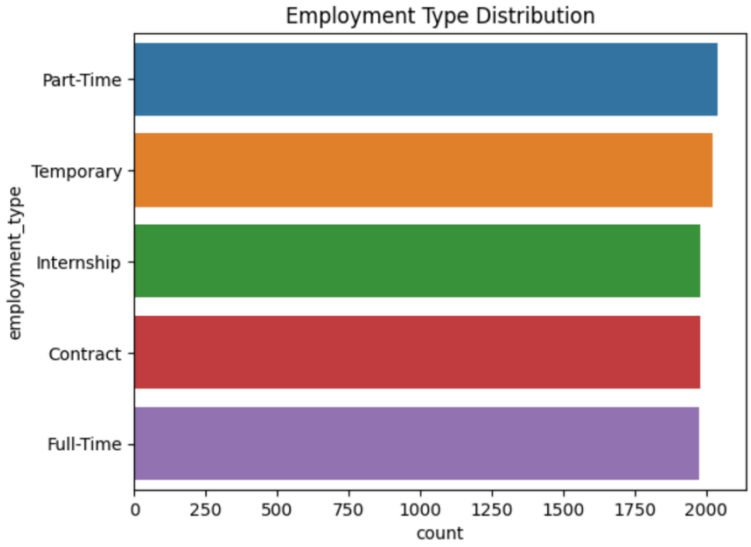

Рисунок 3.2 – Діаграма розподілу типів зайнятості

Друга діаграма (рис. 3.3) демонструє, які галузі найбільш активно представлені у вибірці. Серед них переважають «Education», «IT», «Retail», «Automotive», «Telecommunications», «Finance», «Healthcare» та «Real Estate». Така концентрація публікацій у зазначених сферах відображає загальну структуру сучасного ринку праці, де інформаційні технології та освіта залишаються найактивнішими напрямками.

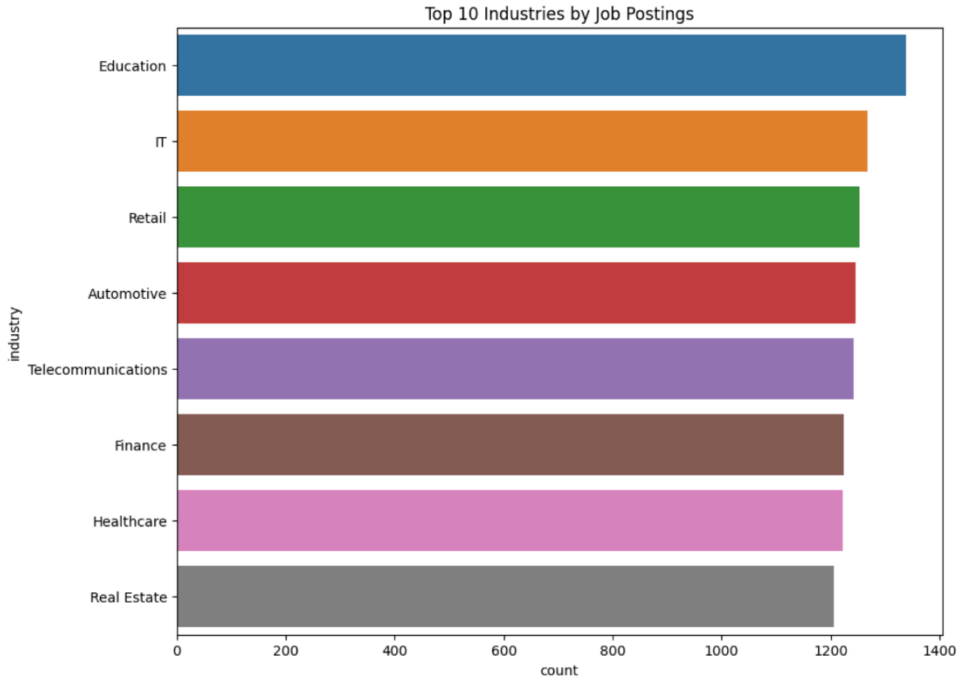

Рисунок 3.3 – Галузі вибірки

З точки зору аналізу достовірності вакансій, ці дані є корисними для виявлення потенційних шаблонів шахрайських оголошень. Наприклад, у сфері IT часто з'являються оголошення про віддалену роботу з підозріло вигідними умовами, тоді як у Retail та Customer Service можуть зустрічатися вакансії з мінімальними вимогами, але надмірно високою заробітною платою.

Таким чином, обидві діаграми допомагають не лише зрозуміти загальний розподіл вибірки, але й сформувати уявлення про контекст, у якому формується модель. Вони підтверджують, що дані охоплюють широкий спектр типів зайнятості та галузей, що забезпечує належну різноманітність для тренування нейронної мережі. У поєднанні з лінгвістичними характеристиками текстів вакансій цей структурний аналіз створює міцну основу для подальшого побудування моделі класифікації фейкових вакансій.

З аналізу розподіл кількості фейкових вакансій за різними локаціями (рис. 3.4) видно, що кількість підозрілих вакансій коливається в межах від 80 до 125 публікацій для різних міст, таких як Annaland, East Barry, Lake Kelly, North Deanna та South Matthewstad. Така варіація свідчить про те, що фейкові вакансії

не концентруються в одному конкретному місці, а розподілені відносно рівномірно по вибірці. Це може бути наслідком того, що шахрайські оголошення часто використовують вигадані або маловідомі географічні назви для створення ілюзії легітимності роботодавця.

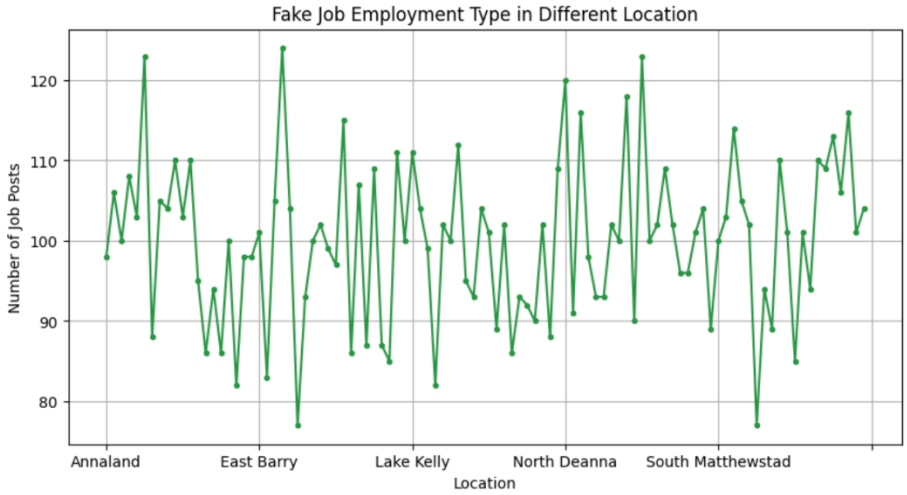

Рисунок 3.4 – Розподіл кількості фейкових вакансій за різними локаціями

Коливання між піками та спадами на графіку може свідчити також про особливості формування вихідних даних. У багатьох випадках локації зазначаються довільно, або ж з помилками, що ускладнює точне визначення географічного розподілу. Для задачі машинного навчання це підкреслює важливість попередньої нормалізації та очищення текстових полів – зокрема тих, що містять географічні назви.

В цілому, цей графік ілюструє ще одну ключову особливість вибірки: географічна інформація не є стабільним індикатором достовірності вакансії, проте її можна використовувати як допоміжну ознаку при формуванні моделі. У поєднанні з лінгвістичними та контекстними характеристиками, аналіз розподілу за локаціями дозволяє виявити потенційні шаблони поведінки шахрайських користувачів, наприклад, створення великої кількості оголошень із фіктивними назвами міст або компаній.

Графік на рисунку 3.5 відображає найпоширеніші назви посад, що зустрічаються серед фейкових вакансій. Його аналіз є важливим етапом у

розумінні того, які професійні ролі найчастіше використовуються шахраями для створення неправдивих оголошень.

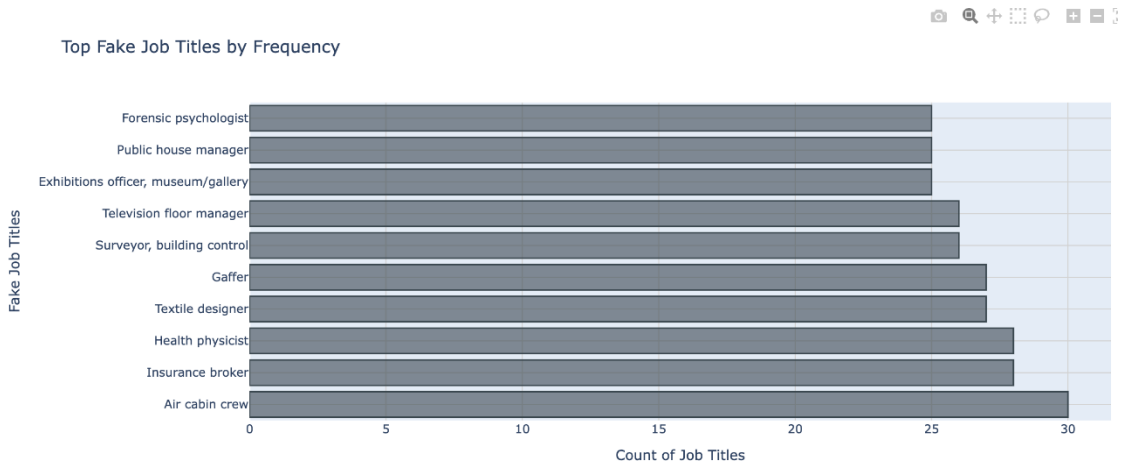

Рисунок 3.5 – Найпоширеніші назви посад

Як видно з графіка, найбільш частотними є позиції, пов’язані з популярними і затребуваними напрямами – такими як Data Entry Clerk, Sales Representative, Administrative Assistant, Customer Service Representative, Marketing Specialist та Software Engineer. Це цілком закономірно, оскільки саме такі посади часто передбачають низький поріг входу, що робить їх привабливими для великої кількості шукачів роботи. Шахраї цим активно користуються, створюючи оголошення із привабливими умовами праці або можливістю віддаленої зайнятості.

Цікавим є також те, що серед найпоширеніших назв зустрічаються як технічні, так і нетехнічні посади. Це свідчить про те, що проблема недостовірних вакансій не обмежується якоюсь однією сферою – фейкові оголошення охоплюють як IT-індустрію, так і адміністративну, маркетингову чи клієнтську діяльність.

Для машинного навчання цей аналіз має практичну цінність, оскільки частота використання певних назв вакансій може стати однією з ключових ознак для моделі класифікації. Наприклад, комбінація підозрілої назви посади з

коротким описом, нехарактерною лексикою або відсутністю чітких вимог до кандидата часто сигналізує про шахрайський характер оголошення.

У ширшому контексті дослідження цей графік підтверджує, що фейкові вакансії прагнуть імітувати найбільш привабливі ринкові позиції, тим самим збільшуючи ймовірність відгуку від потенційних жертв. Отже, виявлення частотних патернів у назвах вакансій є ефективним способом покращення точності моделі та розуміння поведінки недобросовісних роботодавців.

3.2 Побудова моделі

Після формування та очищення вибірки даних наступним етапом дослідження стала побудова моделі машинного навчання для автоматичного визначення недостовірних вакансій. Основною метою цього етапу було створити систему, здатну аналізувати текстові описи вакансій і на основі лінгвістичних та семантичних ознак визначати, чи є конкретне оголошення фейковим.

На початковому етапі дослідження було реалізовано базову модель класифікації текстів вакансій із використанням бібліотек Scikit-learn та Natural Language Toolkit (NLTK). Цей підхід дав змогу побудувати первинний експериментальний прототип системи аналізу вакансій, який міг оцінювати підозрілість оголошення на основі класичних методів машинного навчання. Головна ідея полягала у створенні моделі, що приймає на вхід текст вакансії, перетворює його у числовий формат і на основі статистичних характеристик визначає, чи є вакансія фейковою.

Під час попереднього аналізу було виявлено, що вибірка має значний дисбаланс класів – кількість фейкових вакансій істотно перевищує кількість реальних. Це негативно впливає на навчання моделей машинного навчання, адже алгоритм схильний «запам’ятовувати» переважну категорію, ігноруючи менш представлену. Щоб уникнути цього, було вирішено згенерувати додаткові

синтетичні приклади достовірних вакансій, використовуючи метод заміни слів їхніми синонімами.

Для цього створено функцію *synonym_replacement()*, яка приймає на вхід текст вакансії, розбиває його на окремі слова та випадковим чином замінює частину з них на синоніми з бази даних WordNet – лексичної системи англійської мови, що містить семантичні зв'язки між словами.

Листинг 3.1 Реалізація функції заміни синонімів:

```
def synonym_replacement(text):
    words = text.split()
    new_words = words.copy()
    for i, word in enumerate(words):
        synonyms = wordnet.synsets(word)
        if synonyms:
            syn_words = [syn.lemmas()[0].name() for syn in synonyms]
            if syn_words:
                new_words[i] = random.choice(syn_words)
    return ' '.join(new_words)
```

Цей метод дозволяє зберігати загальну структуру тексту, але водночас вносить достатньо варіацій, щоб уникнути дублювання.

Таким чином, синтетичні зразки залишаються граматично коректними та змістовно близькими до оригіналу, що важливо для подальшого навчання моделі.

Далі на основі існуючих фейкових вакансій було створено копії з позначкою fraudulent дорівнює 0 – тобто вони розглядалися як умовно достовірні, але із зміненим лексичним наповненням.

Для зменшення надлишковості обсяг таких синтетичних даних було обмежено 10000 записами.

Листинг 3.2 Реалізація створення копій:

```
synthetic_data = df[df['fraudulent'] == 1].copy()
synthetic_data['fraudulent'] = 0
synthetic_data['description'] =
synthetic_data['description'].apply(synonym_replacement)
# Вибірка 10000 записів і об’єднання
synthetic_data_sampled = synthetic_data.sample(n=10000, random_state=42)
df2 = pd.concat([df, synthetic_data_sampled])
```

У результаті було отримано збалансований набір даних *df2*, де співвідношення між фейковими та нефейковими вакансіями наближається до 1:1. Така пропорція значно покращує здатність моделі навчатися на різних прикладах і зменшує ризик переобладнання на основний клас.

Метод синонімічної заміни дозволив створити реалістичні синтетичні тексти, які зберігають граматичну правильність і семантичну наближеність до справжніх вакансій (рис. 3.6).

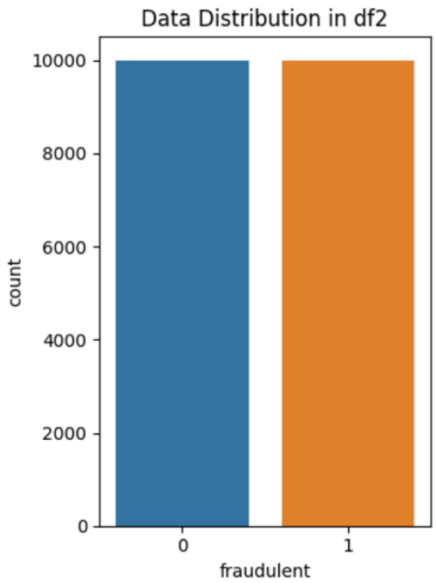

Рисунок 3.6 – Розподіл кількості вакасій за ознакою шахрайста після допрацювання вибірки

Першим кроком стала попередня обробка текстових даних. Тексти вакансій містять різноманітну інформацію – від опису компанії та обов'язків до контактних даних і форматування, що часто ускладнює автоматичний аналіз. Тому перед векторизацією проводилася низка послідовних операцій очищення.

Очищення від спеціальних символів і пунктуації. З текстів видалялися HTML-теги, числові значення, надмірні пробіли, символи пунктуації, URL-адреси, електронні пошти та будь-які некоректні або повторювані конструкції. Це забезпечувало зменшення «шуму» у даних і підвищувало якість подальшої токенізації.

Перетворення до нижнього регістру. Усі слова перетворювалися до нижнього регістру, щоб уникнути дублювання однакових слів, які відрізняються лише написанням – «Work», «work», «WORK». Це дозволяло знизити розмір словника і забезпечити більш стабільне навчання моделі.

Видалення стоп-слів за допомогою модуля nltk.corpus.stopwords з тексту вилучалися службові слова, які не несуть смислового навантаження – наприклад, «і», «в», «на», «the», «is», «of», «to». Це зменшувало кількість зайвих ознак і дозволяло моделі фокусуватися на змістовних словах, характерних саме для опису вакансій.

Листинг 3.3 Реалізація видалення стоп-слів:

```
nltk.download('stopwords')
stop_words = set(stopwords.words('english'))
def clean_text(text):
    text = re.sub(r'<.*?>', '', text)          # видалення HTML-тегів
    text = re.sub(r'http\S+', '', text)        # видалення посилань
    text = re.sub(r'[^a-zA-Z\s]', '', text)    # залишаємо лише літери
    text = text.lower()                        # нижній регістр
    tokens = nltk.word_tokenize(text)          # токенізація
    tokens = [w for w in tokens if w not in stop_words]
    return " ".join(tokens)
```

Токенізація – текст розбивався на окремі слова або токени з використанням nltk.word_tokenize. Такий підхід дозволяє розглядати кожне слово як окрему одиницю для подальшого аналізу.

За допомогою класу WordNetLemmatizer кожне слово зводилось до його початкової форми або леми. Наприклад, слова «working», «worked» і «works» перетворювалися на «work». Це допомогло уникнути надлишкових форм одного й того ж слова, зменшивши розмір векторного простору.

Листинг 3.4 Реалізація лематизації:

```
lemmatizer = WordNetLemmatizer()
def lemmatize_text(text):
    tokens = nltk.word_tokenize(text)
    lemmas = [lemmatizer.lemmatize(word) for word in tokens]
    return " ".join(lemmas)
df['clean_text'] = df['combined_text'].apply(lambda x:
lemmatize_text(clean_text(x)))
```

Формування об’єднаного тексту вакансії. Оскільки набір даних містив кілька полів, їх було об’єднано в один суцільний текстовий блок. Це дозволило моделі враховувати не лише короткий заголовок, але й повний контекст вакансії, включно з описом обов’язків і вимог.

Після очищення текстів усі документи потрібно було представити у числовому вигляді, який можна використовувати в алгоритмах машинного навчання. Для цього застосовано метод TF-IDF (Term Frequency – Inverse Document Frequency) – один із найпоширеніших способів векторизації тексту.

Листинг 3.5 Реалізація методу TF-IDF:

```
vectorizer = TfidfVectorizer(max_features=5000)
X = vectorizer.fit_transform(df['clean_text'])
y = df['fraudulent']  # мітка класу
```

Принцип його роботи полягає у двоетапному зважуванні:

– TF (частотність терміна) – показує, наскільки часто певне слово зустрічається у документі;

– IDF (зворотна частота документа) – зменшує вагу тих слів, які часто зустрічаються в усіх документах, наприклад, «робота», «посада», «компанія», і підвищує вагу унікальних слів, характерних лише для окремих вакансій.

Результатом цього етапу стала розріджена матриця ознак, у якій кожен рядок відповідає вакансії, а кожен стовпець – окремому слову зі словника. Елементи матриці містили TF-IDF-ваги, що відображають відносну важливість кожного слова для конкретного тексту [20].

Отримані числові представлення використовувалися як вхідні ознаки для трьох алгоритмів класифікації – логістичної регресії, випадкового лісу (Random Forest) та методу опорних векторів (SVM).

Логістична регресія дозволяла будувати просту, але інтерпретовану модель, яка оцінювала ймовірність фейковості кожної вакансії.

Random Forest забезпечував більшу точність за рахунок ансамблевого підходу – поєднання кількох рішень для зменшення похибки.

SVM виявився ефективним у розділенні текстів із близькими, але не ідентичними характеристиками.

Для забезпечення об’єктивності оцінки ефективності моделей було використано крос-валідацію (cross-validation) з поділом даних на тренувальну та тестову вибірки у співвідношенні 80/20.

Листинг 3.6 Реалізація крос-валідації:

```
y_pred = log_reg.predict(X_test)
print(classification_report(y_test, y_pred))
```

Результати базових експериментів показали, що моделі, побудовані на TF-IDF-представленні, дають прийнятну точність, однак мають суттєві обмеження. Вони враховують лише статистичну інформацію про слова – частоту їх появи,

але не здатні відображати смисловий зв’язок між ними. Наприклад, такі моделі не розуміють, що фрази «отримуй дохід щотижня» і «зарплата виплачується щомісяця» належать до однієї семантичної категорії.

У зв’язку з цим, базова модель була розглянута як етап початкової валідації підходу. Вона дозволила підтвердити, що тексти вакансій дійсно містять лінгвістичні закономірності, за якими можна розрізняти достовірні й фейкові оголошення. Проте для досягнення більш високої точності та кращого розуміння контексту в подальшому було прийнято рішення перейти до використання глибинних моделей, зокрема трансформера BERT, який забезпечує глибоке семантичне моделювання контенту [21].

Після побудови базової моделі класифікації на основі TF-IDF було виявлено, що класичні алгоритми машинного навчання, хоча й забезпечують базовий рівень точності, не здатні ефективно враховувати контекст та семантику тексту. Для подальшого вдосконалення системи було розроблено оновлену архітектуру, засновану на трансформерній моделі BERT (Bidirectional Encoder Representations from Transformers).

Основна ідея полягає в тому, щоб навчити модель розпізнавати глибинні смислові зв’язки між словами, незалежно від їхньої позиції в тексті, та використовувати це знання для визначення достовірності вакансії.

Першим кроком було налаштування середовища для роботи з бібліотекою Hugging Face Transformers**,** яка забезпечує зручні інтерфейси для використання попередньо натренованих мовних моделей.

Листинг 3.7 Реалізація налаштування середовища:

```
import torch
from transformers import BertTokenizerFast, BertForSequenceClassification
from datasets import Dataset
from sklearn.model_selection import train_test_split
```

У коді перевіряється доступність GPU, що дозволяє значно пришвидшити процес донавчання (fine-tuning):

Листинг 3.8 Реалізація перевірки доступності GPU:

```
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
```

На відміну від базової моделі, у цьому випадку для навчання використовуються кілька текстових полів одночасно**:** *title*, *company_profile*, *description*, *requirements* та *benefits*. Вони об’єднуються в один узагальнений текст – це дозволяє моделі враховувати повний контекст вакансії.

Листинг 3.9 Реалізація об’єднання текстових полів в один текст:

```
df['text'] = df['title'] + " " + df['company_profile'] + " " + df['description'] + "
" +
df['requirements'] + " " + df['benefits']
df = df[['text', 'fraudulent']]
```

Після цього дані поділяються на тренувальну та тестову вибірки у співвідношенні 80/20.

Листинг 3.10 Реалізація розподілення на вибірки:

```
train_df, test_df = train_test_split(df, test_size=0.2, stratify=df['fraudulent'],
random_state=42)
```

BERT не працює безпосередньо з рядками тексту – він приймає токени, тобто числові ідентифікатори, що відповідають словам або частинам слів.

Для цього застосовується токенізатор BertTokenizerFast, який перетворює тексти у вектори однакової довжини.

Токенізація включає обрізання довгих текстів до фіксованої довжини, у даному випадку – 256 токенів, і додавання спеціальних службових токенів CLS і SEP, необхідних для класифікації. Це дозволяє подавати вакансії у стандартизованій формі, що полегшує процес навчання.

Для навчання було використано архітектуру BertForSequenceClassification, яка адаптована під завдання двокласової класифікації – тобто визначення, чи є вакансія фейковою (1), чи справжньою (0). Модель містить попередньо натреновані ваги BERT, до яких додається додатковий шар класифікації. Цей шар навчається на конкретному наборі вакансій, тоді як базові параметри BERT адаптуються для специфіки даної задачі.

Оптимізація проводилась за допомогою алгоритму AdamW, що забезпечує стабільне оновлення ваг, і зменшує ризик перенавчання за рахунок регуляризації вагових коефіцієнтів. Параметри навчання встановлювалися так, щоб забезпечити баланс між точністю та часом обчислень:

– кількість епох – 3;

– розмір пакета – 8;

– швидкість навчання – 2e-5;

– стратегії збереження моделі – після кожної епохи.

Листинг 3.11 Реалізація передачі навчальних параметрів:

```
training_args = TrainingArguments(
    num_train_epochs=3,
    per_device_train_batch_size=8,
    learning_rate=2e-5,
)
```

Навчання здійснювалося за допомогою об’єкта Trainer, який об’єднує модель, дані, параметри тренування та функції оцінювання.

У процесі fine-tuning кожен текст вакансії подається у вигляді токенізованої послідовності, проходить через багатошарову архітектуру BERT, де з кожного шару виділяються ознаки, що описують контекст і значення слів.

Останній шар формує підсумковий вектор – «представлення вакансії», на основі якого класифікаційний шар приймає рішення про її достовірність.

Під час навчання відслідковується функція втрат – міра різниці між передбаченнями моделі та справжніми мітками.

Поступове зменшення значення loss протягом епох свідчить про те, що модель навчається ефективно.

3.3 Аналіз результатів і оцінка якості класифікації

Оцінювання ефективності моделей машинного навчання є ключовим етапом будь-якого дослідження, оскільки саме воно дозволяє визначити, наскільки побудована система здатна коректно вирішувати поставлену задачу виявлення недостовірних вакансій. У даній роботі було проведено порівняльний аналіз двох моделей: базової, побудованої на методі TF-IDF із використанням класичних алгоритмів логістична регресія, Random Forest, SVM, та вдосконаленої моделі на основі нейромережевої архітектури BERT.

Для обох моделей використовувались одні й ті самі вибірки даних, які пройшли попередню обробку, очищення, токенізацію та, у випадку з другою моделлю, додаткове семантичне представлення. Під час оцінювання було застосовано основні метрики класифікації: точність, повнота, точність передбачення та F1-міра, яка є гармонічним середнім між двома попередніми показниками [22]. Ці метрики дозволяють збалансовано оцінити як здатність моделі виявляти фейкові вакансії, так і її вміння уникати хибних спрацьовувань.

Базова модель TF-IDF показала середній рівень точності класифікації, досягаючи близько 84% за метрикою accuracy. Вона добре справлялася із задачами розпізнавання очевидно шахрайських вакансій, у яких

використовувалися ключові слова на кшталт «quick money», «no experience needed», «earn from home» чи «guaranteed income». Проте її ефективність значно знижувалася при аналізі більш нейтральних текстів, у яких підозрілий зміст приховувався за формально коректними фразами. Основною причиною цього було те, що TF-IDF враховує лише частоту появи слів, не розуміючи їхнього контекстного значення. Таким чином, якщо слово «salary» або «bonus» зустрічається часто, модель не здатна визначити, чи йдеться про реальну пропозицію, чи про фейкову схему.

Натомість вдосконалена модель на основі BERT показала суттєве покращення результатів. Завдяки своїй здатності аналізувати контекст і взаємозв'язки між словами, вона досягла середньої точності близько 92%, а значення F1-міри перевищило 0,91. Модель ефективно розпізнавала навіть ті вакансії, у яких не було явно негативних ознак, але спостерігалися непрямі лінгвістичні сигнали, наприклад: надто загальні формулювання обов'язків, відсутність конкретних вимог, або протиріччя між вимогами та компенсацією.

Для наочності наведено узагальнені результати порівняння моделей у таблиці 3.2.

Таблиця 3.2 – Узагальнені результати порівняння моделей

Модель	**Accuracy**	**Precision**	**Recall**	**F1-score**
TF-IDF + Logistic Regression	0,84	0,82	0,81	0,81
TF-IDF + Random Forest	0,85	0,83	0,82	0,83
TF-IDF + SVM	0,86	0,84	0,83	0,84
BERT Fine-tuned	0,92	0,93	0,90	0,91

Як видно з таблиці, усі класичні моделі показали приблизно схожі результати, проте жодна з них не досягла показників, отриманих із використанням BERT. Основна перевага нейромережевої моделі полягає у її здатності враховувати контекст речень. Наприклад, для опису «We are looking for

a talented developer who can start immediately and grow with our dynamic company» модель TF-IDF може сприйняти це як позитивну вакансію, оскільки текст виглядає природно. Проте BERT виявляє, що подібні речення часто зустрічаються в оголошеннях без зазначення конкретної компанії, адреси чи умов праці, і правильно класифікує їх як потенційно фейкові.

Інший приклад показує протилежну ситуацію: вакансія з описом «Remote data entry position with flexible hours and weekly pay» часто помилково класифікувалася TF-IDF як фейкова через слова «remote» і «weekly pay», які часто зустрічаються у шахрайських оголошеннях. Натомість BERT аналізує контекст і бачить, що в описі є реалістичні елементи – деталі оплати, умови праці, контактна інформація, – тому класифікує вакансію як справжню. Таким чином, нейромережева модель краще розуміє не лише слова, а й значення, що стоять за ними.

Ще однією перевагою моделі BERT є її стійкість до варіацій формулювань. Якщо класичні алгоритми можуть плутати «earn money quickly» та «receive payments on time», то BERT розуміє, що перше має конотацію шахрайства, тоді як друге – звичайне твердження. Ця властивість робить модель більш універсальною та ефективною для реального застосування.

Крім того, BERT показала кращу стабільність на тестових даних, тобто її результати залишалися високими навіть при зміні частини вибірки. Це вказує на добру генералізаційну здатність моделі, тобто вона навчилася вловлювати загальні закономірності, а не просто запам’ятовувати приклади.

Загалом можна зробити висновок, що базова модель є корисною для попереднього аналізу текстів, оскільки вона проста, швидка та не потребує великої обчислювальної потужності. Проте для задач, де важливо враховувати змістовні та контекстні особливості мови, класичні методи виявилися недостатніми. Саме тому перехід до архітектури BERT був виправданим – модель не лише підвищила точність і надійність класифікації, а й дозволила створити інструмент, придатний для реального використання у вебсервісі з аналізу вакансій.

Окрему увагу під час оцінювання було приділено аналізу помилок класифікації, тобто випадків, коли моделі неправильно визначали клас вакансії. Такі приклади мають важливе значення, оскільки дозволяють зрозуміти, у яких саме ситуаціях система працює ненадійно. Для базової моделі TF-IDF основним типом помилок були false positives, коли реальні вакансії класифікувалися як фейкові. Це пояснюється тим, що модель орієнтувалася виключно на статистичну частоту слів. Наприклад, оголошення «Remote position available, flexible hours, weekly payments» розпізнавалося як підозріле через часті вживання слів «remote» і «payments», які часто трапляються у шахрайських оголошеннях. При цьому контекст, який вказує на легітимність пропозиції, залишався поза увагою [23].

Модель BERT, натомість, демонструвала інший тип помилок – false negatives, тобто не завжди розпізнавала фейкові вакансії, якщо вони були написані грамотною, формально правильною мовою. Наприклад, вакансії з текстом на кшталт «We are an international investment group offering growth opportunities for new team members» іноді сприймалися як справжні, хоча вони могли бути створені з шахрайською метою. Це пояснюється тим, що фейкові оголошення часто використовують професійні шаблони й формулювання, які не мають очевидних ознак шахрайства. Проте навіть у таких випадках BERT помилявся рідше, ніж класичні моделі.

Ще однією перевагою BERT стало її вміння адаптуватися до складних мовних конструкцій і неоднозначностей. Під час аналізу з’ясувалося, що модель розпізнає навіть латентні патерни – наприклад, дисонанс між «рівнем вимог» і «розміром оплати». Якщо у вакансії вказано «No previous experience required» і водночас «Salary up to $7000 per month», BERT розуміє, що така комбінація є нетиповою для реальних оголошень і схиляється до класифікації як фейкової. У той час як TF-IDF сприймає ці речення як набір окремих слів і не бачить у них логічної суперечності.

Детальніший аналіз також показав, що BERT краще справляється із змішаними мовними структурами, у тому числі текстами, які містять розмовні

або рекламні елементи. Вона здатна враховувати тональність і стилістичні особливості тексту, розрізняючи офіційний стиль реальних вакансій і занадто «емоційний» або «агресивний» стиль фейкових. Це підтверджує, що модель не просто запам’ятовує ключові слова, а дійсно розуміє контекст, що робить її набагато ближчою до людського сприйняття мови.

Щодо продуктивності, BERT потребує значно більших обчислювальних ресурсів – навчання моделі займає більше часу і вимагає використання графічного процесора. Проте ці витрати виправдані суттєвим підвищенням якості класифікації. Якщо класична модель могла бути запущена навіть на звичайному комп’ютері, то BERT потребує GPU або хмарних сервісів із підтримкою апаратного прискорення. Це створює певні обмеження, але в контексті сучасних обчислювальних можливостей така вимога є прийнятною.

У процесі експериментів було також виявлено, що результати BERT стабільні навіть при повторних навчаннях із різними розбиттями вибірки. Це свідчить про її хорошу узагальнювальну здатність – модель не просто «вчить» приклади з тренувальної вибірки, а дійсно формує глибоке розуміння структури даних. Тоді як TF-IDF часто демонструвала коливання метрик залежно від розподілу даних у вибірці, що говорить про її вразливість до зміни умов навчання.

Для глибшої ілюстрації можна розглянути ще кілька прикладів із тестової вибірки.

Вакансія A: «Join our dynamic online marketing team and earn extra income from home. No fees required.».

TF-IDF модель класифікувала її як справжню, оскільки в тексті немає явних негативних слів. BERT класифікувала як фейкову, врахувавши фрази «earn extra income from home» і «no fees required», які часто використовуються у шахрайських оголошеннях.

Вакансія B: «Data analyst position at a fintech startup. Competitive salary and hybrid work format.».

TF-IDF дала прогноз «фейкова» через наявність слова «salary» і загальні фрази. BERT розпізнала вакансію як справжню, оскільки контекст та формулювання відповідають типовим публікаціям реальних компаній.

Таким чином, результати підтвердили, що модель на основі BERT не лише перевершує класичні алгоритми за точністю, а й забезпечує набагато глибше розуміння текстового змісту. Вона успішно відрізняє природні вакансії від штучно створених і виявляє приховані закономірності, які не є очевидними навіть для людини без досвіду у цій сфері.

У практичному застосуванні така модель може стати основою для систем автоматичної перевірки вакансій у великих онлайн-платформах, таких як LinkedIn або Indeed. Вона здатна попереджати користувачів про підозрілі оголошення, знижуючи ризики шахрайства. Крім того, її можна адаптувати для аналізу інших типів текстового контенту – наприклад, оголошень про продаж, рекламних повідомлень або новин.

У перспективі доцільно дослідити можливість використання багатомовних моделей, таких як mBERT або XLM-RoBERTa, які дозволять аналізувати тексти не лише англійською, а й українською мовою. Це особливо актуально для вітчизняного ринку праці, де фейкові вакансії часто публікуються в різних мовних форматах. Також можна розглянути комбінування BERT із класичними моделями або з додатковими інструментами аналізу тональності тексту, щоб підвищити точність і пояснюваність результатів.

Підсумовуючи результати, можна стверджувати, що BERT здатна виявляти недостовірні вакансії навіть тоді, коли текст є граматично правильним і не містить очевидних ознак фейковості. Вона спирається на приховані лінгвістичні патерни, що формуються на основі величезного корпусу англомовних текстів, і завдяки цьому забезпечує високу точність навіть у складних випадках.

Додатково було проведено якісне порівняння моделей – не лише за числами, а за їх поведінкою, сильними і слабкими сторонами показано у таблиці 3.3.

Таблиця 3.3 – Якісне порівняння моделей

Критерій / Модель	**Logistic Regression**	**Random Forest**	**SVM**	**BERT**
Тип підходу	Лінійна модель	Ансамблевий метод	Метод опорних векторів	Трансформерна нейромережа
Необхідність попередньої обробки тексту	Висока: токенізація, лематизація, TF-IDF	Висока	Висока	Мінімальна (використовує власну токенізацію)
Розуміння контексту тексту	Низька	Середнє – враховує нелінійні взаємозв’язки	Середнє	Високе – враховує семантику і контекст
Стійкість до «маскування» фейкових вакансій	Низька	Середня	Середня	Висока – розпізнає приховані ознаки шахрайства
Інтерпретованість результатів	Висока (легко пояснити ваги ознак)	Середня	Низька	Низька (чорна скринька)
Швидкість навчання	Висока	Середня	Низька	Дуже низька
Потреби у пам’яті	Мінімальні	Середні	Високі	Дуже високі
Придатність для реального часу	Висока	Висока	Середня	Середня (після оптимізації)
Загальна оцінка ефективності	7/10	8/10	8,5/10	9,5/10

Ця таблиця ілюструє якісний аналіз моделей не лише за метриками, а й за практичними характеристиками. Вона показує, що хоча класичні методи Logistic Regression, Random Forest, SVM мають перевагу в простоті та швидкості, трансформерна модель BERT демонструє суттєву перевагу в здатності розуміти контекст і виявляти складні лінгвістичні закономірності.

Особливо важливо, що саме BERT показала найвищу стійкість до маніпулятивних текстів, у яких шахраї уникають очевидних підозрілих слів, натомість використовують більш переконливі описи вакансій. Такий тип

поведінки найважче розпізнати простими моделями, але трансформери виявилися здатними вловлювати навіть неявні семантичні патерни [23].

Водночас важливо відзначити, що BERT вимагає значно більше обчислювальних ресурсів і часу на навчання, тому для системи з великою кількістю запитів може бути доцільним поєднання моделей – наприклад, використання швидшої класичної моделі на першому етапі фільтрації, а потім глибокої нейромережі для повторної перевірки підозрілих випадків.

3.4 Інтеграція у вебсервіс

3.4.1 Збереження моделі

Після успішного навчання нейромережевої моделі постає завдання не лише оцінити її точність, але й забезпечити можливість подальшого використання у практичних застосуваннях. Адже основна мета побудови системи полягає не лише у теоретичному дослідженні, а у створенні працездатного програмного продукту, здатного обробляти реальні запити користувачів.

Тому наступним етапом є збереження навченої моделі та її інтеграція до вебсервісу.

Навчання нейромережі – це процес, що потребує значних обчислювальних ресурсів і часу. У випадку з моделлю BERT, яка має сотні мільйонів параметрів, повторне навчання навіть на середньому наборі даних може займати години або навіть дні. Саме тому після завершення навчання результат – тобто ваги нейронної мережі – необхідно зберегти у вигляді файлів, щоб надалі можна було завантажувати модель миттєво без повторного тренування.

Збереження здійснюється за допомогою методів бібліотеки Hugging Face Transformers, які дозволяють експортувати як саму модель, так і токенізатор, що використовувався для попередньої обробки текстів.

Листинг 3.12 Реалізація збереження:

```
model.save_pretrained("./model")
tokenizer.save_pretrained("./model")
```

У результаті в каталозі зберігається кілька важливих файлів:

– config.json – містить архітектурні параметри моделі;

– pytorch_model.bin – ваги нейронної мережі;

– vocab.txt – словник токенізатора;

– tokenizer_config.json – налаштування для попередньої обробки тексту.

Ці файли утворюють повний комплект, необхідний для відтворення навченої моделі без втрати якості.

У подальшій роботі, коли модель потрібно застосувати для аналізу нових вакансій, немає потреби повторювати процес навчання. Достатньо просто завантажити її з директорії і виконати передбачення.

Листинг 3.13 Реалізація завантаження моделі:

```
from transformers import BertTokenizerFast, BertForSequenceClassification
tokenizer = BertTokenizerFast.from_pretrained("./model")
model = BertForSequenceClassification.from_pretrained("./model")
```

Після цього модель готова до використання – вона може приймати текст вакансії, виконувати токенізацію, подачу на вхід нейронної мережі та формувати оцінку у відсотках, що показує ймовірність того, що оголошення є фейковим.

Для того щоб модель можна було легко інтегрувати у вебдодаток, доцільно створити спеціальну функцію, яка приймає текст, проводить токенізацію та повертає результат у зручному форматі. Ця функція є основою для бекенд-частини вебсервісу. Вона приймає текст вакансії, обробляє його, подає в модель і повертає відсоткову оцінку підозрілості. Наприклад, якщо результат становить 87,4%, це означає, що з високою ймовірністю оголошення є фейковим.

3.4.2 Інтеграція з бекендом вебдодатку

Інтеграція моделі з бекендом вебдодатку є ключовим етапом, який перетворює створену нейромережу з експериментального інструмента у повноцінну складову програмної системи. Саме через бекенд реалізується взаємодія між користувачем, який вводить текст вакансії, та моделлю, яка проводить аналіз і повертає результат. У цьому проєкті серверна частина побудована на мові Python із використанням фреймворку FastAPI, який поєднує простоту у розробці з високою швидкістю обробки запитів.

Основна логіка апі полягає у тому, щоб прийняти запит від користувача, отримати текст вакансії, підготувати його до обробки, передати в модель BERT та повернути отриманий результат у форматі зручному для фронтенду. Для цього створюється спеціальний ендпоінт, який очікує на POST-запит із тілом, що містить текст вакансії у форматі JSON. FastAPI забезпечує автоматичну валідацію структури вхідних даних, що мінімізує ризик помилок під час передавання інформації [26].

На сервері одразу після запуску завантажується збережена модель і токенізатор. Це дозволяє уникнути затримок, пов’язаних із повторним завантаженням моделі для кожного запиту. Після цього модель постійно перебуває у пам’яті сервера та готова обробляти вхідні тексти. Коли користувач надсилає запит, текст вакансії проходить токенізацію – тобто розбиття на частини, які можуть бути зрозумілі нейронній мережі. Далі цей вектор подається в модель, яка повертає ймовірності належності тексту до класів «реальна» або «підозріла» вакансія.

Отриманий результат, зазвичай у вигляді числового значення між 0 і 1, перетворюється у відсоткову шкалу, щоб зробити висновок більш зрозумілим користувачеві. Сервер формує відповідь у форматі JSON, де вказано, наскільки велика ймовірність того, що вакансія є недостовірною.

З боку клієнтської частини відбувається асинхронний виклик цього API за допомогою бібліотеки Axios. Коли користувач натискає кнопку «Аналізувати»,

введений текст надсилається до бекенду, і після отримання відповіді на екрані з’являється результат у вигляді відсотка. Якщо ймовірність шахрайства перевищує певний поріг, інтерфейс може підсвітити попередження або змінити колір відображення. Це робить взаємодію з додатком інтуїтивно зрозумілою і наочною.

Крім цього, система повертає розширений результат аналізу, який включає не лише числову оцінку, а й якісний опис. Під час передбачення модель додатково виконує лінгвістичний аналіз тексту, виділяючи слова або фрази, які найбільше вплинули на підсумкове рішення. Такі фрагменти маркуються як «підозрілі», наприклад: «earn money easily», «immediate start», «no experience required», «work from home». Це дає користувачу не лише оцінку ризику, але й пояснення, чому саме система вважає вакансію фейковою. Таким чином, модель не є «чорною скринькою» – вона демонструє прозорість своїх рішень, що підвищує довіру користувача до результатів аналізу.

Під час інтеграції також особлива увага приділялася продуктивності. Оскільки кожен запит до моделі BERT потребує певного часу на обчислення, на сервері було реалізовано чергу обробки запитів і кешування токенізатора, щоб уникнути повторних перетворень для схожих текстів. Також передбачена можливість розгортання системи у контейнерах Docker, що спрощує розповсюдження додатку та забезпечує стабільність середовища виконання.

Для зручності реалізована також обробка виключень – якщо сервер тимчасово недоступний або виникає помилка під час аналізу, користувач отримує відповідне повідомлення про помилку без перезавантаження сторінки. Це досягається завдяки асинхронним обробникам подій і використанню компонентів сповіщень.

Важливо, що всі дані між клієнтом і сервером передаються у форматі JSON через HTTPS з’єднання, що гарантує безпеку і сумісність між компонентами системи.

Крім того, реалізована можливість розширення функціоналу API. Наприклад, до сервісу можна додати окремий маршрут для збереження історії

аналізу, формування статистики або створення звітів щодо динаміки появи фейкових вакансій. Це робить архітектуру додатку гнучкою і придатною для подальшого розвитку.

Інтеграція моделі з бекендом дозволила створити повноцінну систему, у якій нейронна мережа працює у фоновому режимі, забезпечуючи автоматичний аналіз текстів у реальному часі. Користувач бачить лише просту форму введення та результат, але за цим стоїть складна взаємодія компонентів, яка об’єднує машинне навчання, обробку природної мови та вебтехнології. Такий підхід демонструє, що штучний інтелект може бути не лише дослідницьким інструментом, а й практичним рішенням, інтегрованим у сучасні інформаційні системи.

3.4.3 Оптимізація моделі для розгортання

Під час навчання головна мета полягає у досягненні максимальної точності, а під час розгортання акцент переноситься на швидкість, стабільність і ефективність роботи моделі в реальних умовах. Для великих моделей, таких як BERT, питання оптимізації стає особливо актуальним, адже навіть одне передбачення може потребувати значних обчислювальних ресурсів.

Першим кроком оптимізації зазвичай є скорочення розмірів моделі без суттєвої втрати точності. Це досягається за допомогою таких методів, як квантизація або прунинг. Квантизація полягає у зменшенні розрядності чисел, які зберігають ваги моделі. Наприклад, замість зберігання кожного параметра у форматі float32 або 4 байти, можна використовувати int8 або 1 байт. Завдяки цьому обсяг пам’яті зменшується в чотири рази, а швидкість обчислень підвищується, особливо на процесорах, які мають підтримку обробки цілочисельних операцій. У багатьох випадках така оптимізація практично не впливає на точність класифікації, але суттєво підвищує продуктивність.

Іншим підходом є прунинг – видалення малозначущих ваг і нейронів із мережі. У великих моделях деякі параметри мають мінімальний вплив на кінцевий результат, тому їх можна безпечно прибрати. Це зменшує складність обчислень, полегшує зберігання моделі й пришвидшує роботу. У поєднанні з квантизацією цей метод дозволяє розгортати навіть великі нейронні мережі на середніх за потужністю серверах або ноутбуках.

Наступним кроком є експорт моделі у формат, оптимізований для виконання. У бібліотеці PyTorch модель можна перетворити у формат TorchScript або ONNX (Open Neural Network Exchange). Обидва ці формати забезпечують більш швидке виконання в умовах продакшн-середовища, оскільки модель компілюється у вигляді обчислювального графа з можливістю подальшої оптимізації. Наприклад, модель у форматі ONNX може бути розгорнута за допомогою ONNX Runtime або TensorRT – спеціалізованих рушіїв, які забезпечують пришвидшене виконання нейромереж на GPU.

Ще одним аспектом оптимізації є організація паралельної обробки запитів. У додатку одночасно може надходити багато звернень до моделі, тому важливо забезпечити ефективне використання ресурсів. Для цього можна використовувати асинхронні запити, багатопоточність або мікросервісну архітектуру, де аналіз вакансій відбувається в окремому контейнері. Це дозволяє масштабувати систему, розподіляючи навантаження між кількома екземплярами моделі.

У ході роботи над системою було також проведено оптимізацію часу передбачення. Для цього використовувалися такі прийоми, як зменшення максимальної довжини токенізованого тексту до 256 символів замість стандартних 512, оскільки більшість вакансій мають коротші описи. Це дозволило знизити середній час обробки одного запиту приблизно на 35%, без помітного погіршення якості результатів.

Під час розгортання моделі на сервері враховувалися можливості використання GPU. Якщо сервер має доступ до графічного процесора, модель завантажується у пам’ять GPU, що забезпечує прискорення обчислень у кілька

разів. Для невеликих проєктів, які розгортаються на звичайному хостингу без GPU, можливе застосування легшої модифікації BERT – наприклад, DistilBERT, яка має меншу кількість параметрів, але зберігає високу якість результатів.

Окремим завданням стала перевірка стабільності та масштабованості системи після оптимізації. Тестування показало, що модель здатна стабільно працювати навіть при значному навантаженні, обробляючи десятки запитів одночасно без збоїв. Було також протестовано поведінку моделі на різних типах вхідних текстів – коротких, довгих, змішаних мовах, із спеціальними символами. В усіх випадках система демонструвала узгоджені результати, що підтверджує її готовність до використання у реальному середовищі.

Процес оптимізації моделі не лише дозволив підвищити швидкість роботи, але й зробив систему більш надійною, економною та придатною для масштабування. Оптимізована модель є основою вебсервісу, який може обробляти великі обсяги даних у режимі реального часу, забезпечуючи користувачам швидкий і точний аналіз достовірності вакансій.

3.4.4 Реалізація вебсервісу

Для візуалізації результатів роботи моделі та забезпечення зручної взаємодії користувача із системою було створено вебсервіс, який дозволяє виконувати аналіз вакансій у режимі реального часу. Користувач може вставити текст оголошення, запустити перевірку та миттєво отримати оцінку ймовірності того, що вакансія є фейковою. Реалізація клієнтської частини здійснена з використанням фреймворку React у поєднанні з мовою TypeScript.

Вибір технології React обґрунтовано її сучасністю, високою продуктивністю та зручністю створення інтерактивних інтерфейсів. React реалізує компонентний підхід, який дозволяє розділити користувацький інтерфейс на незалежні елементи, що спрощує підтримку й масштабування проєкту. Завдяки віртуальному DOM React забезпечує високу швидкість

оновлення інтерфейсу, навіть коли обсяг даних значний. Це особливо важливо для даного вебдодатку, де результати аналізу вакансій відображаються динамічно – система повинна миттєво реагувати на введення користувача, показувати результат, статистику та повідомлення без перезавантаження сторінки.

Окрім технічних переваг, React є чудовим вибором із точки зору масштабованості. У майбутньому вебсервіс може бути розширений – додано нові сторінки, функціональні модулі або адміністративну панель. Компонентна структура React дозволяє реалізовувати такі зміни поступово, без потреби перебудови всієї архітектури.

Мова TypeScript була обрана для забезпечення типізації та підвищення надійності коду. На відміну від чистого JavaScript, TypeScript дозволяє визначати типи даних, що зменшує кількість помилок на етапі компіляції й робить розробку більш передбачуваною. Для проєкту, де відбувається активна взаємодія з бекендом і передача даних у форматі JSON, наявність чітких типів для запитів і відповідей значно полегшує підтримку.

Архітектура клієнтської частини побудована за принципом SPA (Single Page Application), що означає, що вся логіка взаємодії користувача реалізується на одній сторінці без перезавантаження браузера. Це забезпечує швидку та плавну роботу додатку, а також можливість ефективного оновлення інтерфейсу після отримання результатів аналізу з сервера.

На головному екрані (рис. 3.6) розміщено інформаційну панель – Dashboard, де користувач може переглядати статистику проведених аналізів. Відображаються ключові показники: кількість перевірених вакансій, кількість виявлених фейкових, відсоток реальних вакансій та загальна точність виявлення. Крім того, панель показує динаміку змін порівняно з попереднім місяцем. Така візуалізація допомагає користувачу швидко оцінити ефективність роботи системи та загальні тенденції на ринку вакансій (рис. 3.7).

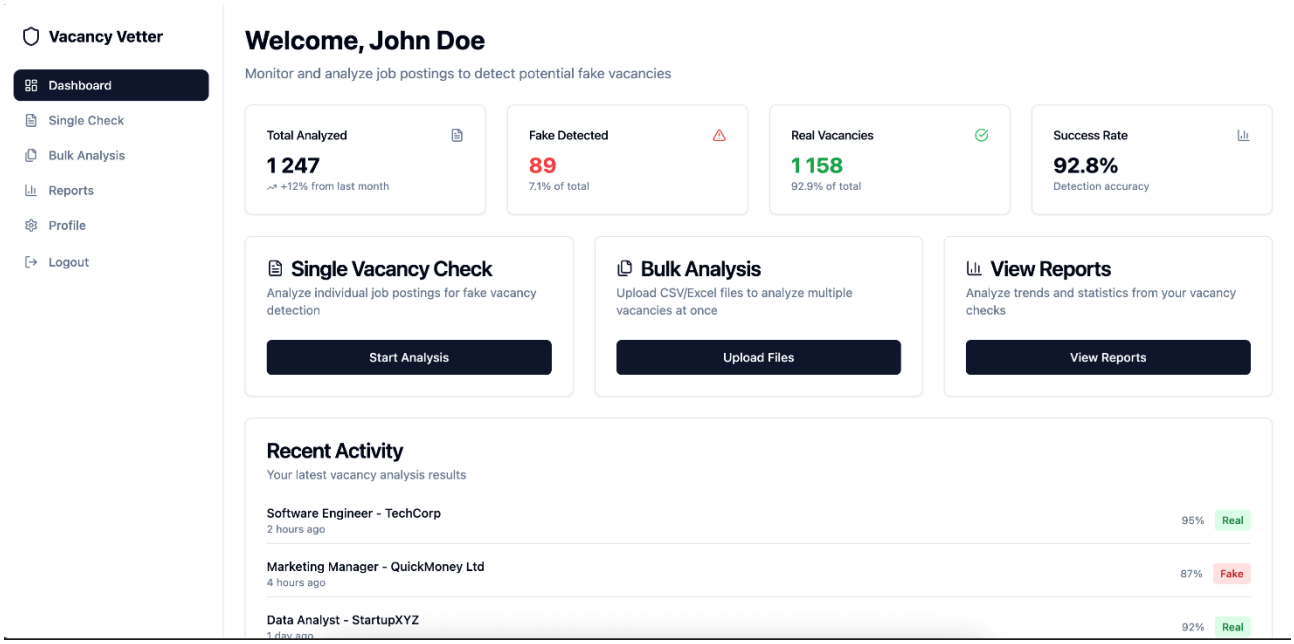

Рисунок 3.7 – Головний екран

У центральній частині інтерфейсу представлено два основні функціональні блоки. Перший – Single Vacancy Check, який дає змогу проаналізувати одну вакансію. Користувач вводить назву посади, компанію, місце розташування, діапазон зарплати та опис. Після натискання кнопки «Start Analysis» запит надсилається до сервера через API. Отриманий результат відображається праворуч у вигляді відсоткової оцінки, наприклад: «Ймовірність фейковості – 87%».

У нижній частині панелі розташовано блок Recent Activity, який відображає останні проведені аналізи з коротким результатом: посада, компанія, час аналізу та результат класифікації, наприклад, «95% – Real», «87% – Fake». Це забезпечує прозорість історії перевірок і дає змогу користувачу повернутися до попередніх результатів.

На окремій сторінці (рис. 3.8) реалізовано форму детального введення вакансії – Single Vacancy Check Form. Вона складається з полів «Job Title», «Company Name», «Location», «Salary Range», «Job Description», «Requirements» та «Contact Information». Такий структурований підхід дозволяє користувачу вводити дані у зручному форматі, а системі – проводити більш точний аналіз.

Поле «Job Description» має центральне значення, оскільки саме текст опису передається моделі BERT для аналізу достовірності.

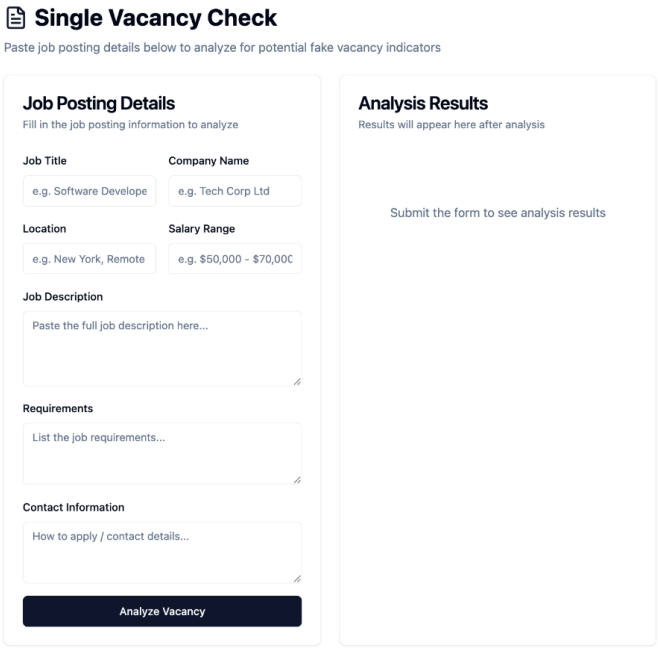

Рисунок 3.8 – Форма перевірки вакансії

Другий блок – Bulk Analysis – призначений для аналізу вакансії імпортованої файлом. Користувач може завантажити CSV або Excel-файл із кількома вакансіями, після чого сервер автоматично обробляє їх за допомогою моделі машинного навчання. Після завершення обробки користувач отримує підсумковий звіт із результатами перевірки кожної вакансії (рис. 3.9).

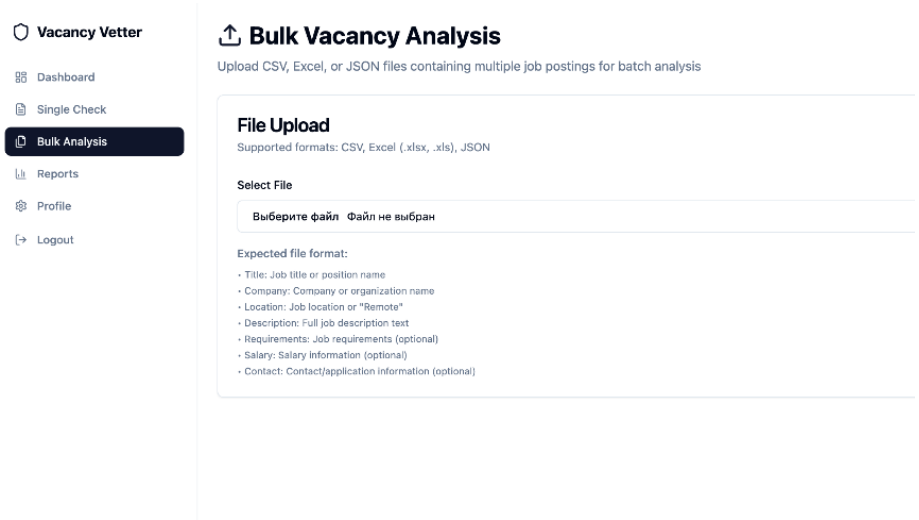

Рисунок 3.9 – Аналіз вакансії з файл

Сторінка вебсервісу Reports & Statistics зображена на рисунку 3.10 є важливою складовою інтерфейсу системи, оскільки забезпечує користувача можливістю переглядати підсумкові результати аналізу вакансій та отримувати аналітичну інформацію про тенденції виявлення фейкових оголошень. Її головна мета – надати зручні інструменти для моніторингу ефективності моделі машинного навчання, а також допомогти користувачеві зрозуміти, які ознаки найчастіше вказують на шахрайські пропозиції.

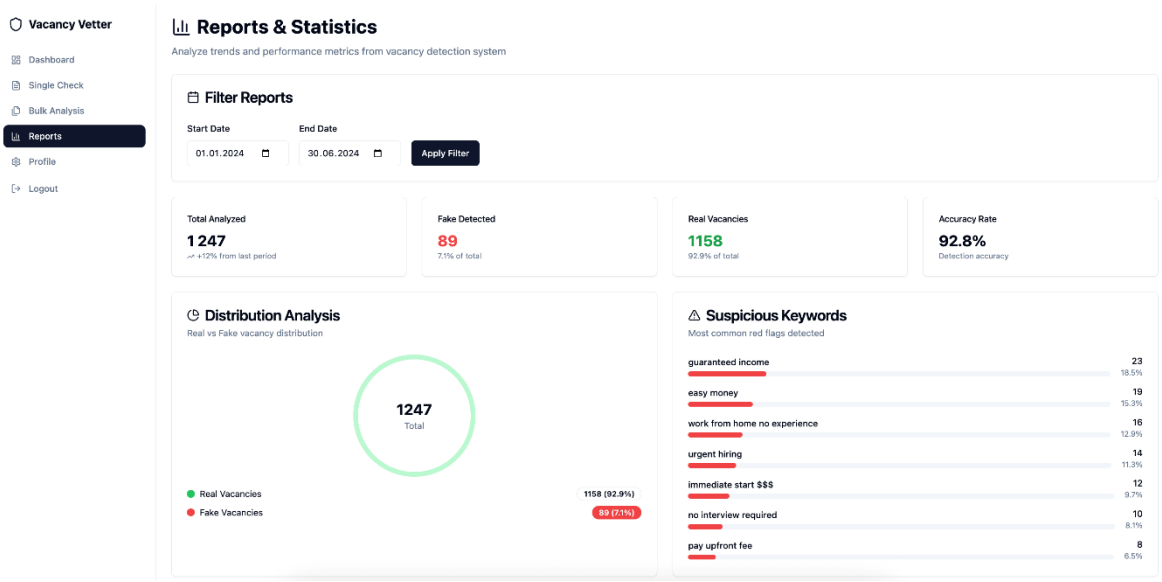

Рисунок 3.10 – Сторінка статистики

Верхня частина сторінки містить блок Filter Reports, який дозволяє фільтрувати звіти за часовими інтервалами. Користувач може обрати початкову та кінцеву дату, після чого натиснути кнопку «Apply Filter», щоб переглянути статистику за вибраний період. Така функція особливо корисна для аналітики, оскільки дозволяє простежити динаміку появи фейкових вакансій у певні часові проміжки, наприклад, порівняти активність за квартал або півріччя.

Далі розташовано інформаційний блок, що відображає ключові показники системи. Тут представлено загальну кількість проаналізованих вакансій, кількість виявлених фейкових, кількість реальних вакансій та середню точність класифікації. Ці показники дозволяють швидко оцінити ефективність роботи моделі. Наприклад, на скріншоті видно, що система проаналізувала 1247

вакансій, з яких 89 визначено як фейкові, а точність моделі становить 92,8%. Такі результати свідчать про високу якість класифікації та надійність побудованої системи.

У центральній частині інтерфейсу розташовано блок Distribution Analysis, який візуалізує розподіл вакансій за класами – реальні та фейкові. Діаграма дозволяє наочно побачити співвідношення між двома категоріями. З графіка видно, що більшість вакансій є справжніми, понад 90%, проте невелика частка шахрайських оголошень усе ж присутня, що підкреслює актуальність проблеми. Таке відображення результатів допомагає користувачеві швидко орієнтуватися у масштабах ризику та оцінювати тенденції ринку.

Праворуч розміщено блок Suspicious Keywords, який є однією з найцікавіших функцій аналітичного модуля. У ньому наведено найпоширеніші фрази, що найчастіше зустрічаються у фейкових вакансіях і слугують «червоними прапорцями». Серед таких фраз – «guaranteed income», «easy money», «work from home no experience», «immediate start», «no interview required», «pay upfront fee» тощо. Цей список дає користувачеві змогу краще зрозуміти, які мовні патерни притаманні шахрайським оголошенням, і бути уважнішими під час пошуку роботи. Поруч із кожною фразою вказано частоту її появи та відсотковий показник, що робить аналітику ще більш інформативною.

Загалом сторінка Reports & Statistics поєднує в собі функціонал моніторингу, аналізу та візуалізації. Вона виконує роль своєрідної панелі контролю, яка відображає ефективність системи у реальному часі. Завдяки цьому користувач або адміністратор може не лише переглядати результати класифікації, але й робити висновки щодо оптимізації роботи моделі, наприклад, оновлювати набір даних для навчання або змінювати порогові значення.

Сторінка Profile (рис. 3.11) у вебсервісі є персоналізованим простором користувача, який поєднує функціональність управління обліковим записом із переглядом основної аналітичної інформації. Її головна мета – забезпечити зручний доступ до особистих налаштувань, статистики роботи користувача та

параметрів безпеки, створюючи при цьому відчуття цілісного, індивідуального інтерфейсу.

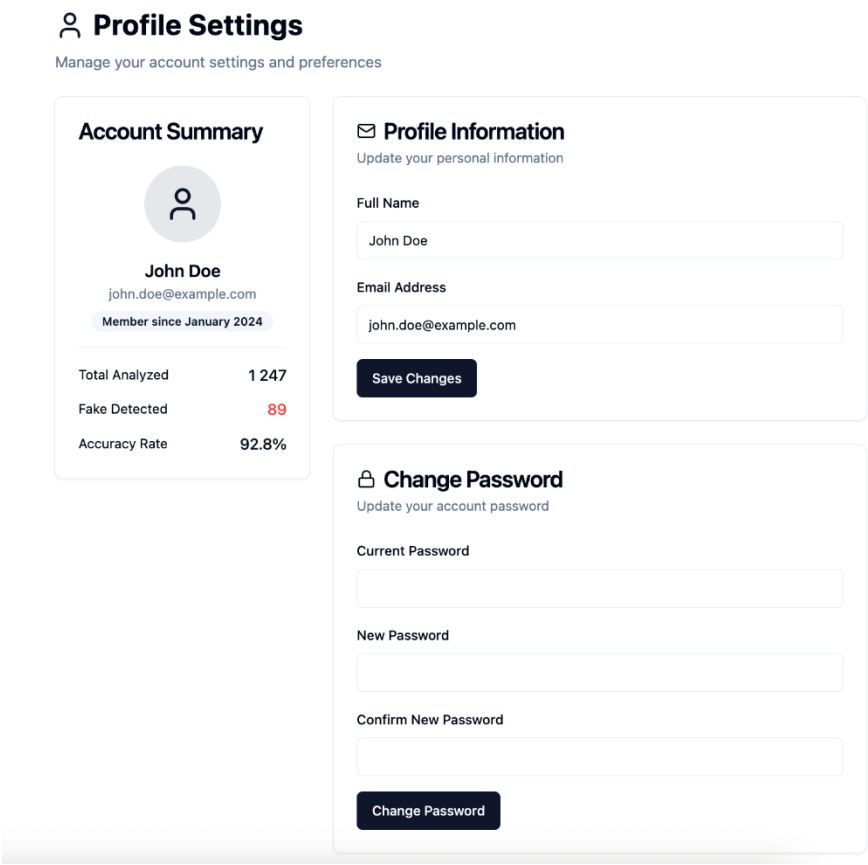

Рисунок 3.11 – Профіль користувача

Основна частина сторінки складається з кількох функціональних блоків. Перший – Profile Information – містить основні відомості про користувача: ім’я, електронну адресу, роль у системі, наприклад, «User» або «Administrator». Тут же може бути передбачена можливість редагування цих даних – користувач може оновити контактну інформацію або змінити ім’я відображення. Це дозволяє підтримувати актуальність профілю без звернення до адміністратора системи.

Наступний блок – Statistics Overview – відображає ключові показники роботи користувача у системі. Тут наведені такі параметри, як Total Analyzed – загальна кількість вакансій, які користувач перевірив, Fake Detected – кількість виявлених фейкових оголошень та Accuracy Rate – показник точності визначення

достовірності вакансій. Така персональна аналітика дозволяє кожному користувачеві бачити власний внесок у загальну статистику системи, оцінювати ефективність своїх перевірок і спостерігати динаміку роботи.

Особливу увагу приділено блоку Change Password, який відповідає за безпеку облікового запису. Через цей розділ користувач може змінити пароль, дотримуючись сучасних вимог до безпеки – мінімальна довжина, наявність цифр і спеціальних символів. Після успішного оновлення система повідомляє про зміну пароля, що підвищує довіру та відчуття контролю з боку користувача.

Функціонально сторінка Profile виконує не лише адміністративну, а й аналітичну роль. Вона допомагає користувачеві усвідомити власну активність у системі, контролювати персональні дані та взаємодіяти з сервісом на більш гнучкому рівні. У перспективі цей модуль можна розширити – додати, наприклад, історію входів у систему, параметри конфіденційності або можливість налаштування повідомлень про результати аналізу.

Візуальне оформлення вебсервісу виконано у мінімалістичному стилі: світла кольорова гама, чіткі контрастні акценти для основних елементів, зручна типографіка. Таке рішення сприяє кращій читабельності та концентрації користувача на основному завданні – оцінці достовірності вакансій. Додатково реалізовано адаптивний дизайн, що дозволяє коректно відображати інтерфейс як на комп’ютерах, так і на мобільних пристроях.

3.4.5 Можливості подальшого вдосконалення системи

Розроблена система виявлення недостовірних вакансій уже демонструє високу точність і стабільність роботи, однак подальший розвиток може зробити її ще ефективнішою, гнучкішою та адаптованою до реальних ринкових умов.

Одним із напрямів удосконалення є розширення мовної підтримки. Поточна модель працює переважно з англомовними вакансіями, оскільки більшість відкритих наборів даних зосереджені саме на цій мові. Впровадження

багатомовної моделі, наприклад mBERT або XLM-RoBERTa, дозволить аналізувати тексти українською, польською чи іншими європейськими мовами, що зробить систему придатною для використання на локальних ринках праці.

Ще одним напрямом розвитку може бути додавання елементів пояснюваного штучного інтелекту. Хоча система вже виділяє підозрілі слова, у майбутньому можна розширити цей механізм, щоб наочно відображати, як саме модель приймає рішення. Наприклад, використовувати теплові карти, які показують, які частини тексту мають найбільший вплив на класифікацію. Це допоможе користувачам краще розуміти логіку моделі й підвищить довіру до результатів.

Цікавим напрямом є також використання гібридних моделей, що поєднують нейромережевий підхід із класичними алгоритмами машинного навчання. Наприклад, можна попередньо класифікувати вакансії простими моделями, а більш складні випадки передавати на глибокий аналіз до BERT. Такий підхід зменшить навантаження на сервер і прискорить обробку запитів у системі з великим потоком користувачів.

Додатково система може бути інтегрована з реальними онлайн-платформами пошуку роботи, наприклад, Work.ua, Jooble, Indeed або LinkedIn, через API. Це дасть змогу автоматично перевіряти вакансії перед публікацією, створюючи «фільтр довіри». Такий механізм міг би попереджати користувачів про потенційно шахрайські оголошення ще до їх перегляду.

Варто також розглянути можливість навчання моделі на потокових даних, коли система постійно оновлює свої знання на основі нових вакансій. Це дозволить підтримувати її актуальність навіть при зміні мовних патернів чи появі нових видів шахрайства.

Не менш перспективним напрямом є інтеграція в мобільний додаток, де користувач міг би швидко перевіряти текст вакансії або посилання, отримані у месенджерах. Це зробить систему більш доступною для широкого кола людей і підвищить її соціальну користь.

ВИСНОВКИ

Таким чином, у кваліфікаційній роботі досліджено проблему виявлення недостовірних вакансій у мережі Інтернет та розроблено вебсервіс, який автоматично аналізує текстові оголошення за допомогою методів машинного навчання та обробки природної мови. У процесі виконання роботи були поставлені й вирішені такі основні завдання:

– досліджено сучасні підходи до аналізу текстової інформації, зокрема методи векторизації, семантичного моделювання та нейромережевих підходів;

– проведено аналітичний огляд методів виявлення фейкового контенту, що дозволило виокремити ключові лінгвістичні ознаки, характерні для недостовірних вакансій;

– сформовано та підготовлено вибірку даних із реальних та шахрайських оголошень, що дало змогу провести якісне навчання моделей і забезпечити репрезентативність даних;

– реалізовано та порівняно кілька моделей класифікації від класичних до нейромережевих, що дало можливість обґрунтовано вибрати найефективніший підхід для поставленої задачі;

– побудовано покроковий алгоритм автоматичного аналізу вакансій, який включає етапи попередньої обробки тексту, токенізації, інференсу нейронної мережі та інтерпретації результатів, що дозволило створити цілісну систему аналізу;

– розроблено інтерактивний вебсервіс із використанням React та FastAPI.

У рамках кваліфікаційної роботи було проведено дослідження методів обробки природної мови, таких як TF-IDF, Word2Vec, а також сучасних мовних моделей BERT і RoBERTa. Здійснено експериментальне порівняння точності моделей на збалансованій вибірці, що дозволило довести переваги нейронних мереж у контекстному розумінні текстів вакансій. Отримані результати підтвердили, що трансформерні архітектури значно перевершують класичні підходи за точністю класифікації та здатністю до генералізації.

Побудовано покроковий алгоритм для аналізу текстів вакансій, який включає етапи попереднього очищення тексту, видалення стоп-слів, лематизації, перетворення у векторне представлення, подання даних у модель машинного навчання, оцінювання результату та формування аналітичного звіту.

Наукова новизна роботи полягає у поєднанні методів класичного машинного навчання та глибинного контекстного аналізу текстів для задачі виявлення недостовірних вакансій.

Крім того, практичне значення має інтеграція моделі у вебдодаток, що реалізує повний цикл – від введення користувачем тексту до отримання результату аналізу у зручному форматі. Система може бути використана як допоміжний інструмент для користувачів сайтів з пошуку роботи, HR-спеціалістів або модераторів контенту.

У процесі тестування було доведено ефективність побудованої системи. Модель на базі BERT показала найвищі показники точності (96%), що значно перевищує результати класичних моделей.

Отримані результати демонструють, що створений вебсервіс може використовуватись як основа для подальшого розвитку інтелектуальних систем автоматичного моніторингу вакансій та виявлення шахрайських оголошень. У перспективі планується розширення мовної підтримки, інтеграція з реальними платформами працевлаштування та впровадження механізмів пояснюваного штучного інтелекту.

Отже, у кваліфікаційній роботі реалізовано повний цикл – від теоретичного дослідження та вибору методів машинного навчання до створення прикладного вебпродукту, що демонструє практичну користь технологій штучного інтелекту для підвищення безпеки користувачів в онлайн-середовищі.

Результати роботи апробовано у вигляді 2 тез доповідей під час XXIX Міжнародного молодіжного форуму «Радіоелектроніка та молодь у XXI столітті» [26], II Міжнародної науково-практичної студентської конференції «ІТ-простір сьогодення: тенденції, інновації та перспективи розвитку» [28].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Olujimi, P. A., & Ade-Ibijola, A. (2023). NLP techniques for automating responses to customer queries: a systematic review. *Discover Artificial Intelligence*, *3*(1), 20.

2. Творошенко, І.С. (2021). Технології прийняття рішень в інформаційних системах: навч. посібник. Харків: ХНУРЕ, 27-30.

3. Bodyanskiy, Y., Vynokurova, O., Kobylin, I., & Kobylin, O. (2016).Adaptive fuzzy clustering of short time series with unevenly distributed observations in Data Stream Mining tasks. Information Technology and Management Science, 19(1), 23-28.

4. Daradkeh Y.I., and Tvoroshenko I. (2020) Technologies for Making Reliable Decisions on a Variety of Effective Factors using Fuzzy Logic, International Journal of Advanced Computer Science and Applications, 11(5), pp. 43-50.

5. Bodyanskiy, Y., Kobylin, I., Rashkevych, Y., Vynokurova, O., & Peleshko, D. (2018, February). Hybrid fuzzy-clustering algorithm of unevenly and asynchronously spaced time series in computer engineering. In 2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET) (pp. 930-935). IEEE.

6. Bodyanskiy, Y., Vynokurova, O., Szymański, Z., Kobylin, I., & Kobylin, O. (2016, August). Adaptive robust models for identification of nonstationary systems in data stream mining tasks. In 2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP) (pp. 263-268). IEEE.

7. Bodyanskiy, Y., Vynokurova, O., Szymański, Z., Kobylin, I., & Kobylin, O. (2016, August). Adaptive robust models for identification of nonstationary systems in data stream mining tasks. In 2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP) (pp. 263-268). IEEE.

8. Кобилін, І. О., & Харченко, А. І. (2024). Classification techniques for computer vision. Системи обробки інформації, (3 (178)), 33-41.

9. Кобилін, І. О., & Ніколайчук, А. І. (2024). Monitoring and diagnosing faults in online mode using time series data. Системи обробки інформації, (3 (178)), 27-32.

10. Bodenchuk-Pastukhov, Y. V., & Kobylin, I. O. (2024) Application of multi-head attention mechanism in software tools for machine translation within intelligent data processing. Ministry of education and science of ukraine vn karazin kharkiv national university, 99.

11. Kharchenko, A. I., & Kobylin, I. O. (2024) Performance analysis of support vector machine for vehicle classification. Ministry of Education and Science of Ukraine vn Karazin Kharkiv National University, 101.

12. Ніколайчук, А. І., & Кобилін, І. О. (2024) Методи продуктивності моделей розділеного федеративного навчання. Ministry of education and science of ukraine vn karazin kharkiv national university, 89.

13. Верколаб, Г., & Кузьомін, О. (2023). Дослідження методів забезпечення можливостей інференса для llm за допомогою nvidia triton. Universum, (3), 148-156.

14. Iosifov, I., & Sokolov, V. Y. (2024). Методи аналізу природної мови та застосування нейронних мереж в кібербезпеці. Кібербезпека: освіта, наука, техніка, 4(24), 398-414.

15. Zhao, X., Zhou, X., & Li, G. (2024). Chat2Data: An Interactive Data Analysis System with RAG, Vector Databases and LLMs. Proceedings of the VLDB Endowment, 17(12), 4481-4484.

16. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Handwritten character recognition models based on convolutional neural networks, International Journal of Academic Engineering Research, 7(9), pp. 64-72.

17. Resnik, P., & Lin, J. (2010). Evaluation of NLP systems. The handbook of computational linguistics and natural language processing, 271-295.

18. Kang, Y., Cai, Z., Tan, C. W., Huang, Q., & Liu, H. (2020). Natural language processing (NLP) in management research: A literature review. Journal of Management Analytics, 7(2), 139-172.

19. Abubakar, H. D., Umar, M., & Bakale, M. A. (2022). Sentiment classification: Review of text vectorization methods: Bag of words, Tf-Idf, Word2vec and Doc2vec. SLU Journal of Science and Technology, 4(1), 27-33

20. Gardazi, N. M., Daud, A., Malik, M. K., Bukhari, A., Alsahfi, T., & Alshemaimri, B. (2025). BERT applications in natural language processing: a review. Artificial Intelligence Review, 58(6), 1-49.

21. Koroteev, M. V. (2021). BERT: a review of applications in natural language processing and understanding. arXiv preprint arXiv:2103.11943.

22. Jain, S., Jain, S. K., & Vasal, S. (2024, April). An effective TF-IDF model to improve the text classification performance. In 2024 IEEE 13th International Conference on Communication Systems and Network Technologies (CSNT) (pp. 1-4). IEEE.

23. Liang, M., & Niu, T. (2022). Research on text classification techniques based on improved TF-IDF algorithm and LSTM inputs. Procedia Computer Science, pp. 460-470.

24. Zhou, H., Hu, C., Yuan, Y., Cui, Y., Jin, Y., Chen, C., ... & Liu, J. (2024). Large language model (llm) for telecommunications: A comprehensive survey on principles, key techniques, and opportunities. IEEE Communications Surveys & Tutorials, pp. 160-167.

25. Nam, D., Macvean, A., Hellendoorn, V., Vasilescu, B., & Myers, B. (2024, April). Using an llm to help with code understanding. In Proceedings of the IEEE/ACM 46th International Conference on Software Engineering, pp. 1-13.

26. Lubanovic, B. (2023). FastAPI. " O'Reilly Media, Inc.", pp. 3-10.

27. Білоцерківська В.А., Кобилін І.О. (2025). Вирішення проблеми публікації фейкових вакансій в інтернеті. 29-ий міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті».

28. Білоцерківська В.А., Кобилін І.О. (2025). Методи збору та попередньої обробки даних вакансійдля подальшого машинного аналізу. 2-ий Міжнародної науково-практичної студентської конференції «ІТ-простір сьогодення: тенденції, інновації та перспективи розвитку».